Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Algorithmische Methoden für kombinatorische chemische Bibliotheken

Kummulative Dissertation

zur Erlangung des akademischen Grades
Dr. rer. nat
an der Fakultät
für Mathematik, Informatik und Naturwissenschaften der
Universität Hamburg

eingereicht am Fachbereich Informatik von

Louis Bellmann

geboren in Hamburg

Hamburg, den 9. Juni 2022

Erstgutachter: Prof. Dr. Matthias Rarey
Zweitgutachter: Prof. Dr. Andrew Torda
Drittgutachterin: Prof. Dr. Sereina Riniker
Tag der Disputation: 4. November 2022

Kurzfassung

Chemische Bibliotheken in virtueller Form bilden den Grundstein des computergestützten Wirkstoffentwurfs. Sie werden beispielsweise nach Molekülen durchsucht, die interessante Eigenschaften im Rahmen einer bestimmten Anwendung aufweisen und als Leitstruktur für ein neues Medikament innerhalb eines Forschungsprojekts verwendet werden könnten. Hierbei spielt sowohl die Qualität als auch die Menge der in einer Bibliothek enthaltenen Moleküle eine entscheidende Rolle. Klassischerweise wird die Molekülmenge einer Bibliothek enumeriert, das heißt jedes Moleküle wird einzeln aufgelistet. Dadurch skaliert der benötigte Speicherplatz und die beanspruchte Rechenzeit für die Durchsuchung der Bibliothek linear mit der Anzahl der enthaltenen Moleküle. Kombinatorische Bibliotheken in virtueller Form verwenden chemische Bausteine und Reaktionen, um einen kombinatorischen Raum von Produkten aufzuspannen. Auf diese Weise ist es möglich eine große Menge von Produkten implizit durch eine begrenzte Menge von Bausteinen und Reaktionen zu beschreiben, wodurch kombinatorische Bibliotheken in der Lage sind mit weniger Ressourcen eine weitaus größere Anzahl von Molekülen abzubilden und durchsuchbar zu machen als klassische enumerierte Bibliotheken.

Algorithmische Verfahren zur Durchsuchung und Analyse von enumerierten chemischen Bibliotheken basieren auf der individuellen Prozessierung aller enthaltenen Moleküle. Damit können diese Ansätze nicht auf kombinatorische Bibliotheken angewendet werden, wenn diese eine für die Enumeration zu große Menge von Molekülen beschreiben. In dieser Dissertation werden deshalb neuartige algorithmische Methoden entwickelt, die den kombinatorischen Charakter dieser Bibliotheken nutzen, um auf Mengen von Milliarden Molekülen und darüber hinaus effizient zu operieren. Zunächst wird eine kompakte Darstellungsform kombinatorischer Bibliotheken beschrieben, die alle nötigen Informationen aus den chemischen Bausteinen und Reaktionen einfängt. Zusätzlich wird eine neue Methode zur Berechnung molekularer Fingerabdrücke vorgestellt, die kompakte und feingranulare, von chemischen Substrukturen getriebene Molekulardeskriptoren erzeugt. Zusammen bilden diese Werkzeuge das Grundgerüst für drei neuartige algorithmische Methoden. Diese ermöglichen erstmals die topologische Ähnlichkeitssuche, Schnittmengenberechnung und Bildung von Eigenschaftsverteilungen in kombinatorischen Bibliotheken. Damit werden für den computergestützten Wirkstoffentwurf essenzielle Funktionalitäten für chemische Bibliotheken verfügbar gemacht, die Milliarden oder sogar Billionen von Molekülen enthalten.

Abstract

Virtual chemical libraries are a corner stone of computer-aided drug design. They are, for example screened to identify compounds with properties that are interesting for a certain application scenario and can function as a lead structure in a drug development project. In this context the quality and amount of compounds contained in a chemical library is of up most importance. Traditionally a chemical library is presented in enumerated form, meaning each of its compounds is listed individually. Due to this representation the amount of memory needed to store and time needed to search the library scales linear with the number of contained compounds. Combinatorial virtual libraries use building blocks and chemical reactions to span a combinatorial space of products. This way they are able to implicitly describe a potentially vast amount of compounds while only employing a limited number of building blocks and reactions. In the process, combinatorial libraries enable the storage and screening of far larger quantities of compounds compared to traditional enumerated libraries.

Algorithmic approaches for the screening and analysis of enumerated chemical libraries are based on the individual processing of compounds. Consequently, these techniques cannot be applied to combinatorial libraries that are too large for enumeration. In this dissertation novel algorithmic methods are developed that utilize the combinatorial character and operate efficiently on libraries exceeding billions of molecules in size. First of all, a compact form of representation for combinatorial libraries is described that captures all relevant information from the employed building blocks and reactions. Additionally, a new molecular fingerprint technique is reported that constitutes a compact, fine-grained and substructure-driven molecular descriptor. Collectively, these tools form the basis for three novel algorithmic methods enabling topological similarity search, overlap calculation and property distribution computation of combinatorial libraries for the first time. With these methods at hand, medicinal chemists are enabled to use essential functionalities of computer-aided drug design on chemical libraries containing billions or even trillions of compounds.

Inhaltsverzeichnis

1. Einleitung

Ein Molekül ist im weitesten Sinne eine abgeschlossene chemische Einheit, die aus einem oder mehreren Atomen besteht. [1] In der Medizinalchemie werden in mehrstufigen Verfahren zu einer konkreten Anwendung passende Moleküle als sogenannte Leitstrukturen identifiziert. Diese Leitstrukturen dienen am Ende eines Optimierungsprozesses als Wirkstoff-Kandidaten für ein neues Arzneimittel. [2]

Zu Beginn der Suche nach einem neuen Wirkstoff könnte theoretisch jedes synthetisch zugängliche Molekül und jeder Naturstoff eine Leitstruktur bilden. Diese Molekülmenge wird chemischer Raum oder chemisches Universum genannt und seine Größe wird auf über 10^{60} Moleküle geschätzt. [3] Auch wenn diese Schätzung nicht unbedingt exakt sein muss, so ist der chemische Raum doch bei Weitem zu groß und unerforscht, um eine für ihn repräsentative Menge an Molekülen auf ihre Eignung als Leitstruktur zu testen.

Stattdessen werden bei der Suche nach Kandidaten für Leitstrukturen chemische Bibliotheken verwendet, die Moleküle einer bestimmten Gruppe, Naturstoffe, kommerziell verfügbare Moleküle oder bereits identifizierte Wirkstoff-Kandidaten aus anderen Forschungsprojekten [4] enthalten. In chemischen Bibliotheken mit teilweise Millionen enthaltener Moleküle wird im Rahmen des Hochdurchsatz-Screenings (*high-throughput screening*) [5]–[7] nach geeigneten Leitstrukturen gesucht. Hierbei werden alle Moleküle einer chemischen Bibliothek automatisiert mithilfe von Robotern einzeln prozessiert [8] und mit speziellen Testsystemen (*assays*) [9] auf ihre Verwendbarkeit als Leitstruktur hin analysiert. Moleküle, die in diesem Rahmen eine bestimmte Bioaktivität aufweisen, werden Treffer (*hit*) genannt. Sie stellen potenzielle Kandidaten für Leitstrukturen dar.

Bei dieser Suche können nur Moleküle als Treffer identifiziert werden, die auch in den verwendeten chemischen Bibliotheken enthalten sind. Aus diesem Grund werden chemische Bibliotheken mithilfe von Vorwissen für eine spezielle Anwendung entwickelt, sogenannte fokussierte Bibliotheken (*focused libraries*). [10] Ein anderer Ansatz ist es chemische Bibliotheken mit hoher Diversität zu verwenden, die einen guten Querschnitt

des chemischen Raumes darstellen. Hierbei kann es sinnvoll sein, sehr viele Moleküle in eine chemische Bibliothek mit aufzunehmen. [11]

1.1. Kombinatorische chemische Bibliotheken

Um die Synthese einer großen Menge von Molekülen zu vereinfachen bzw. erst zu ermöglichen, werden Methoden aus der kombinatorischen Chemie verwendet. [12]–[14] Hierbei werden die gleichen Syntheseschritte auf eine Menge chemischer Bausteine angewandt, um eine große Anzahl von Molekülen effizient zu generieren. Die Menge der so erzeugten Moleküle wird *kombinatorische chemische Bibliothek* genannt. Hierbei wird ein kombinatorischer Raum aufgespannt, ähnlich dem kartesischen Produkt mehrerer Mengen. Die Anzahl der so generierten Moleküle kann die Anzahl der verwendeten chemischen Bausteine um mehrere Größenordnungen übersteigen.

Da die verwendeten Syntheseverfahren bei diesem Vorgehen immer gleich sind, können sie auf alle chemischen Bausteine bzw. Bausteinkombinationen auf die gleiche Weise angewendet werden. [15] Dies ergibt die Möglichkeit der parallelen Synthese mehrerer oder aller Moleküle [16]. In einem *split-and-pool*-Verfahren [17] wird eine Menge von chemischen Bausteinen oder Zwischenprodukten in gleiche Teile partitioniert und jeder Teil wird in einem Syntheseschritt mit einem anderen chemischen Baustein kombiniert. Danach werden alle Teile zusammengefasst und gegebenenfalls wird der Vorgang wiederholt.

Zunächst wurden in einer Festphasen-Synthese (*solid-phase synthesis*) [18] lineare Peptide synthetisiert, die einen kombinatorischen Raum aus Aminosäuren aufspannen. [12], [16], [19] Später gelang es auch kombinatorische chemische Bibliotheken aus makrozyklischen Peptiden [20] und strukturell diversen, naturstoff-ähnlichen Molekülen [21] zu synthetisieren.

Durch die parallele Synthese im Rahmen des split-and-pool-Verfahrens entstehen Lösungen, in denen Moleküle mit vielen unterschiedlichen Strukturen enthalten sind. Wenn im Rahmen der Forschung an einem neuen Wirkstoff ein Molekül aus der Lösung mit interessanten Eigenschaften identifiziert wird, dann ist nicht unbedingt klar, welche der vielen möglichen Strukturen es aufweist. Um dieses Problem zu umgehen, können die Moleküle einer kombinatorischen chemischen Bibliothek mit einem Marker kodiert werden. Es wurden verschiedene Methoden der Kodierung entwickelt [20], [22], aber das heute gebräuchlichste Verfahren verwendet DNA-Fragmente. [23] Kombinatorische chemische

Bibliotheken, die die Kodierung über DNA-Fragmente verwenden, werden DNA-kodierte Bibliotheken (*DNA-encoded libraries* (DEL)) genannt. Es wurde verschiedene Synthese-Verfahren für DNA-kodierte Bibliotheken entwickelt [24]–[26], aber ihnen ist gemein, dass jeder Marker eines Moleküls aus einer Kombination aus DNA-Fragmenten besteht. Jedes DNA-Fragment kodiert einen chemischen Baustein, der in der Synthese des Moleküls verwendet wurde. Durch die Art und Reihenfolge der im Marker enthaltenen DNA-Fragmente kann die chemische Struktur des kodierten Moleküls eindeutig bestimmt werden. Durch diese Kodierung ist es nun möglich, alle Moleküle einer DNA-kodierten Bibliothek gleichzeitig in einer einzigen Lösung im Rahmen eines Tests zu prozessieren. Durch einen Aufbereitungsschritt verbleiben nur noch die Treffer-Moleküle. Diese können anschließend über ihren DNA-Marker, beispielsweise mithilfe einer Polymerase-Kettenreaktion (*polymerase chain reaction*(PCR)), [27] identifiziert werden. Dieses Verfahren ist effizienter umsetzbar, als das klassischen Hochdurchsatz-Screening, da nicht jedes Molekül der chemischen Bibliothek einzeln analysiert werden muss.[28], [29]

In dieser Dissertation werden computergestützte Methoden beschrieben, die kombinatorische chemische Bibliotheken in virtueller Form prozessieren und Medizinalchemiker bei der Suche nach Treffer-Molekülen und Leitstrukturen unterstützen. Ähnlich wie bei der Synthese von physischen kombinatorischen chemischen Bibliotheken, wird der zugrundeliegende kombinatorische Charakter dieser Bibliotheken genutzt, um Effizienz zu erreichen.

1.2. Virtuelle chemische Bibliotheken

In den 1950er Jahren wurde begonnen chemische Daten in digitaler Form zu repräsentieren, zu sammeln und diese Daten computergestützt zu durchsuchen. [30] Hierbei können zu einem Molekül zB. Information über dessen Struktur, chemische Eigenschaften, experimentelle Daten, weiterführende Literatur und vorliegende Patente hinterlegt werden. Um ein Molekül über seine Struktur automatisiert identifizieren zu können, ist eine entspreche Repräsentation nötig, die von Computern prozessierbar ist. Zu diesem Zweck wurden viele Datenformate entwickelt, die die Struktur [31]–[33] und sogar räumliche Konformation [34], [35] der Moleküle einfangen.

Virtuelle chemische Bibliotheken beschreiben eine Menge von Molekülen, die durch ihr Repräsentation in Datenformaten automatisiert prozessierbar sind. In der Bio- und

Chemieinformatik wurden zahlreiche Verfahren der Analyse virtueller chemischer Bibliotheken entwickelt. Es ist etwa möglich, computergestützt nach Molekülen mit bestimmten Eigenschaften zu suchen, [30], [36]–[38] den Bindungsmodus von Molekülen der Bibliothek an einem Protein zu simulieren, [39] Moleküle zu gruppieren [40] oder physikochemische Eigenschaftsverteilungen zu berechnen. [41], [42] Die Suche in einer virtuellen chemischen Bibliothek ist meist mit deutlich weniger Aufwand an Zeit und Ressourcen verbunden, als ein Testverfahren einer physischen chemischen Bibliothek. Heute existieren viele virtuelle chemische Bibliotheken [35], [43]–[46], die eine große Rolle bei der Suche nach neuen Wirkstoffen spielen. [47]

1.3. Motivation

Klassischerweise wird jedes Molekül in einer virtuellen chemischen Bibliothek durch einen eigenen Eintrag beschrieben. In dieser Dissertation werden wir in diesem Fall von einer *enumerierten* virtuellen chemischer Bibliothek, oder verkürzt von einer enumerierten Bibliothek, sprechen. Heute existieren enumerierte virtuelle chemische Bibliotheken, die mehrere Milliarden Molekülen enthalten. [48] Zwar sind dies weitaus mehr Moleküle als jede physische chemische Bibliothek enthält, allerdings ist die Speicherung und die Suche nur mit hohem Ressourcenaufwand in einem serverbasierten Ansatz möglich. [48], [49] Der Ressourcenbedarf enumerierter virtueller chemischer Bibliotheken wächst proportional mit der Anzahl enthaltener Moleküle. Dies stellt einen Engpass bei der Entwicklung neuer, stetig wachsender Bibliotheken dar.

Um dieses Problem zu umgehen, werden im Kapitel 1.1 beschriebene kombinatorische Ideen angewendet. Das Ziel ist es, eine große Menge von Molekülen durch eine deutlich kleinere Anzahl an Elementen zu beschreiben. Die Menge und Eigenschaften der Moleküle, die in einer kombinatorischen physischen chemischen Bibliothek enthalten sind, ergeben sich eindeutig aus den verwendeten chemischen Bausteinen und der Art, wie sie in der Synthese durch chemische Reaktionen verknüpft werden. Es ist also möglich mit einer begrenzten Anzahl chemischer Bausteine und Reaktionen eine weitaus größere Menge an Molekülen einer kombinatorischen chemischen Bibliothek zu beschreiben. Genau dieses Konzept verwenden wir zum Aufbau einer *kombinatorischen* virtuellen chemischen Bibliothek, die wir auch verkürzt kombinatorische Bibliothek nennen. Hierbei werden chemische Bausteine, durch Datenformate repräsentiert, und chemische Reaktionen, durch Verknüpfungsregeln beschrieben, abgespeichert. Die finalen Moleküle, die bei der

Synthese entstehen würden, werden nicht explizit aufgelistet. Dies unterscheidet die kombinatorische von der enumerierten virtuellen chemischen Bibliothek.

In den 1990er Jahren wurde begonnen kombinatorische Ansätze im Kontext der Entwicklung virtueller chemischer Bibliotheken zu verwenden. [50]–[52] Hierbei wurden Repräsentationen chemischer Bausteine und Verknüpfungsregeln verwendet, um effizient große Mengen potenziell noch unbekannter Moleküle in virtueller Form zu generieren. So entstanden virtuelle chemische Bibliotheken, die zwar eine zugrundeliegende kombinatorische Struktur aufwiesen, allerdings dennoch enumerierte Bibliotheken waren.

Der Grund für die Enumeration der Moleküle liegt in der computergestützten Prozessierbarkeit dieser virtuellen chemischen Bibliotheken. Alle zu diesem Zeitpunkt existierenden Verfahren zur Analyse von und Suche in virtuellen chemischen Bibliotheken basierten auf der individuellen Prozessierung jedes Moleküls einer Bibliothek. Dies ist nicht möglich in einer kombinatorischen virtuellen chemischen Bibliothek, da die Moleküle nicht explizit aufgelistet werden. Durch diese Eigenschaft existierender Verfahren, wächst ihr Bedarf an Rechnerressourcen proportional mit der Anzahl Moleküle einer enumerierten Bibliothek.

Eine computergestützte Analyse- oder Suchmethode in kombinatorischen Bibliotheken muss in der Lage sein die Menge der implizit beschriebenen Moleküle zu betrachten. Dabei darf sie allerdings nur auf der Menge der repräsentierten chemischen Bausteine und Verknüpfungsregeln operieren. Um dies zu erreichen, muss sie den kombinatorischen Charakter der Bibliothek zu ihrem Vorteil nutzen. Dies wird das grundlegende Paradigma aller in dieser Dissertation vorgestellten Methodiken sein. Das erste Suchverfahren in kombinatorischen virtuellen chemischen Bibliotheken wurde 2001 entwickelt. [53] Nun war es der Methode Ftrees-FS erstmals möglich eine Menge an Molekülen virtuell zu durchsuchen und dabei nur die repräsentierten chemischen Bausteine und Verknüpfungsregeln zu verwenden. Die genaue Funktionsweise der Methode wird in Kapitel 4.2 beschrieben.

In dieser Dissertation beschreiben wir neue computergestützte Methoden, die in kombinatorischen Bibliotheken operieren können. Die Implementation der beschriebenen Methoden basiert auf der NAOMI-Plattform [54] und erweitert diese. Die entwickelten Methoden können, in Software-Modulen implementiert, von Medizinalchemikern im Wirkstoffentwurf verwendet werden. Ihr Aufbau und ihre Bedienung wird in dieser Dissertation beschrieben.

In Kapitel 2 werden wir auf existierende kombinatorische virtuelle chemische Bibliotheken eingehen, ihre Darstellungsformen beschreiben und schließlich eine eigene Repräsentation vorstellen. Die Vorteile dieser neuen Struktur werden in den nachfolgenden Kapiteln verwendet. In Kapitel 3 wird ein neuer Molekulardeskriptor vorgestellt, der für die Anwendung in kombinatorischen Bibliotheken vorteilhafte Eigenschaften besitzt. Wir beschreiben die zugrundeliegende Methode und vergleichen seine Leistung mit existierenden Ansätzen. In Kapitel 4 stellen wir ein neuartiges Verfahren zur Suche in kombinatorischen Bibliotheken vor. Es verwendet den in Kapitel 3 beschriebenen Molekulardeskriptor und ermöglicht erstmals eine Suche nach Molekülen, die topologisch ähnlich zu einer Anfrage sind. In Kapitel 5 und Kapitel 6 werden Methoden vorgestellt, mit denen es erstmals möglich ist, kombinatorische Bibliotheken in ihrer Gesamtheit zu vergleichen und nach gewissen Eigenschaften zu analysieren. Die beschriebenen Methoden werden auf prominente kombinatorische Bibliotheken angewandt.

2. Kombinatorische Bibliotheken und ihre Darstellung

In Kapitel 1.3 haben wir die Möglichkeit kombinatorischer virtueller chemischer Bibliotheken beschrieben, eine weitaus größere Menge an Molekülen darzustellen als klassische enumerierte virtuelle chemische Bibliotheken. Hierbei werden chemischen Bausteine und Synthesewege verwendet, um einen kombinatorischen Raum aufzuspannen. Diese Informationen müssen von einer Darstellungsform einer kombinatorischen Bibliothek eingefangen werden. In diesem Kapitel gehen wir auf existierende Formen der Darstellung kombinatorischer Bibliotheken und etablierte Werkzeuge zu ihrer Erzeugung ein. Weiterhin geben wir einen Überblick über virtuelle chemische Bibliotheken, die mithilfe dieser Darstellungsformen bereits erzeugt wurden. Schlussendlich werden die innerhalb dieser Dissertation entwickelten topologischen Fragmenträume vorgestellt und von existierenden Verfahren abgegrenzt.

2.1. Werkzeuge im Rahmen der Erstellung virtueller chemischer Bibliotheken

Um eine physische kombinatorische chemische Bibliothek zu entwickeln, wird eine Auswahl chemischer Bausteine und auf diese angewendete Syntheseprotokolle benötigt. In ähnlicher Weise werden zur Erstellung einer virtuellen chemischen Bibliothek mit kombinatorischem Charakter ein oder mehrere Datensätze chemischer Bausteine, sowie Beschreibungen chemischer Reaktionen verwendet. Diese Informationen werden dann computergestützt prozessiert, um die virtuelle chemische Bibliothek zu generieren.

2.1.1. Virtuelle chemische Bausteinbibliotheken

Unternehmen aus der Chemiebranche, die chemische Bausteine in physischer Form anbieten, stellen auch oft virtuelle chemische Baustein-Bibliotheken für die computergestützte Suche zur Verfügung. [55]–[58] Diese enumerierten virtuellen chemischen Bibliotheken können als Datensätze für die Auswahl passender chemischer Bausteine im Rahmen der Erzeugung einer kombinatorischen virtuellen chemischen Bibliothek dienen und sind meist in den prominentesten Datenformaten [32], [34] für Moleküle verfügbar. Der Vorteil bei der Verwendung dieser Baustein-Bibliotheken ist die Möglichkeit der Akquise physischer Bausteine nach einer computergestützten Suche oder Identifikation eines interessanten Moleküls. Des Weiteren gibt es Plattformen, die diese Baustein-Bibliotheken zusammenfassen. [44], [59] So erweitern sie die Auswahl an beschriebenen chemischen Bausteinen und ermöglichen auch beispielsweise den Vergleich von Lieferzeiten und Preisen.

2.1.2. Beschreibung chemischer Reaktionen

Es wurden verschiedene Datenformaten entwickelt, um chemische Reaktionen computergestützt prozessierbar zu machen und relevanten Informationen einzufangen. Manche Formate, sowie SMIRKS [32] und RXN, [34] beschreiben die explizite Veränderung von Atomen und chemischen Bindungen am Reaktionszentrum. Das Format reactionSMARTS [37] erweitert das SMIRKS Format um die Verwendung formaler Sprache und rekursiver Ausdrücke, wodurch mehr Details der beschriebenen chemischen Reaktionen beachtet werden können. reactionSMARTS wurden verwendet, um eine Sammlung robuster chemischer Reaktionen zu beschreiben, die für den Aufbau virtueller chemischer Bibliotheken verwendet werden können. [60] Das Datenformat-Paar CHMTRN/PATRAN [61] ist in der Lage Eigenschaften von Edukten und Produkten zu beschreiben, die sich nicht unbedingt auf das Reaktionszentrum beschränken. Mithilfe dieses Format-Paars wurde ein Datensatz von 2300 beschriebenen chemischen Transformationen entwickelt. [62]

2.1.3. Vorhersage chemischer Zugänglichkeit und retrosynthetische Verfahren

Bisher haben wir die Möglichkeit beschrieben, eine virtuelle chemische Bibliothek mithilfe eines schon existierenden Datensatzes chemischer Bausteine zu generieren. Die Einträge dieser Datensätze repräsentieren dabei physische chemische Bausteine, die z.B. von Chemie-Unternehmen geliefert werden können. Dies kann nützlich sein, da es so einfacher

möglich ist, ein Produkt der virtuellen chemischen Bibliothek tatsächlich zu synthetisieren und es im Wirkstoffentwurf zu verwenden. Diese Art der Erzeugung entspricht dem Vorgehen bei der Entwicklung einer physischen kombinatorischen Bibliothek. Allerdings sind somit die chemischen Eigenschaften der Produkte nicht direkt klar, da lediglich über die Auswahl an chemischen Bausteinen und Reaktionen entschieden wird. Zusätzlich können nur bereits bekannten chemische Bausteine verwendet werden.

Stattdessen können auch Verfahren aus der Retrosynthese angewendet werden, um Datensätze chemischer Bausteine und Verknüpfungsregeln für eine virtuelle chemische Bibliothek zu generieren. In der Retrosynthese geht es um die Aufgabe, für ein gegebenes Molekül einen möglichen Syntheseweg und dazugehörige Edukte zu identifizieren. Für die Erzeugung einer virtuellen chemischen Bibliothek wird zunächst eine Menge an Molekülen mit geeigneten chemischen Eigenschaften ausgewählt. Für diese Moleküle werden, durch computergestützte retrosynthetische Analysen, ein oder mehrere Synthesewege und chemische Bausteine identifiziert. Diese Informationen werden dann verwendet, um eine virtuelle chemischen Bibliothek kombinatorischen Charakters zu erzeugen. Durch dieses Verfahren ist gesichert, dass die ursprüngliche Menge an Molekülen mit interessanten chemischen Eigenschaften in der virtuellen chemischen Bibliothek enthalten ist. Zusätzlich wird die Bibliothek durch diese Methode potenziell durch die Aufspannung des kombinatorischen Raums mit weiteren Molekülen aufgefüllt. Für diese zusätzlichen Moleküle ist die Synthetisierbarkeit allerdings nicht unbedingt gesichert, da für sie keine retrosynthetische Analyse durchgeführt wurde. Dies stellt einen Nachteil gegenüber der direkten Erzeugung einer virtuellen chemischen Bibliothek mit bekannten chemischen Bausteinen und Reaktionen dar.

In den 1980er Jahren wurden erste computergestützte Verfahren zur Retrosynthese entwickelt. [63] Hierbei wird ein Datensatz von chemischen Transformationen [62] verwendet und ihre Anwendbarkeit auf ein gegebenes Eingabemolekül überprüft. Andere wissensbasierte Verfahren wurden entwickelt, die ebenfalls auf einem Datensatz chemischer Reaktionen beruhen. [64] Dieser regelbasierte Ansatz wurde in der RECAP [65] Methode und später in BRICS [66] verwendet, um chemische Bausteine und Verknüpfungsregel automatisiert mithilfe einer Eingabemenge von Molekülen zu erzeugen.

In den letzten Jahren wurden zusätzlich Konzepte des maschinellen Lernens für die computergestützte Retrosynthese verwendet. [67]–[69] Das Ziel ist es, durch Wissensextraktion aus einer Reaktionsdatenbank und Verallgemeinerung des Gelernten, Synthesewege für Moleküle zu bestimmen, die außerhalb der Wissensdomäne der verwendeten Datenbank

liegen. Auf diese Art bestimmte Reaktionen und chemischen Bausteine wurden bisher nicht zur Erzeugung einer virtuellen chemischen Bibliothek verwendet.

2.2. Überblick über bestehende virtuelle chemische Bibliotheken mit kombinatorischen Eigenschaften

Im Folgenden beschreiben wir existierende virtuelle chemische Bibliotheken mit kombinatorischem Charakter. Wir erläutern ihre Darstellungsform und die in Kapitel 2.1 beschriebenen Werkzeuge, die zu ihrer Erzeugung verwendet wurden.

SAVI [70] ist eine virtuelle chemische Bibliothek, die mithilfe eines Datensatzes von 152.532 chemischen Bausteinen des Unternehmens Enamine Ltd. [55] und 53 chemischen Reaktionen aus dem Projekt LHASA [62] erzeugt wurde. Alle chemischen Reaktionen verwenden zwei chemische Bausteine. Mit dem Programm CACTVS [71] und dem Datenformat-Paar CHMTRN/PATRAN [61] wurden die Reaktionen und chemischen Bausteine prozessiert, um 1,75 Milliarden Moleküle zu erzeugen. Alle Produkte in SAVI sind einzelnd aufgelistet. Somit ist SAVI eine enumerierte Bibliothek mit zugrundeliegendem kombinatorischen Charakter.

SCUBIDOO [72] ist eine enumerierte virtuelle chemische Bibliothek, die 21 Millionen virtuelle Moleküle enthält. Zur Erzeugung wurden chemische Bausteine der Firma ChemBridge [58] verwendet, sowie 58 chemische Reaktionen, die als reactionSMARTS [37] kodiert sind. Die Moleküle wurden mit dem Python Modul RDKit [73] enumeriert und nach den PAINS [74] Kriterien für Assay-Interferenzen gefiltert. Durch diesen Filterungs-Prozess weißt SCUBIDOO keinen reinen kombinatorischen Charakter mehr auf.

Mit der Pythonbibliothek SynthI [75] ist es Nutzern möglich eine maßgeschneiderte virtuelle chemische Bibliothek zu erzeugen und zu durchsuchen. Dies ist sowohl über die Eingabe eines Datensatzes chemischer Bausteine möglich, als auch mithilfe einer Menge von Molekülen, aus denen in einem retrosynthetischen Verfahren chemische Bausteine und Verknüpfungsregeln entstehen. Für beide Ansätze werden reactionSMARTS [37] zur Repräsentation von chemischen Reaktionen verwendet. Die beschriebenen Reaktionen verwenden zwei Komponenten und es werden keine Ringe gebildet. Im Gegensatz zu SCUBIDOO und SAVI können die repräsentierten chemischen Bausteine, hier Synthons genannt, auch individuell ausgegeben werden. Diese Synthons enthalten markierte Wasserstoffe, die die Reaktionsstelle markieren und zur späteren Verknüpfung der Synthons zu

Molekülen verwendet werden können. Mithilfe SynthI ist es möglich eine kombinatorische Bibliothek zu erzeugen, die nicht enumeriert ist.

Das Verfahren TOPAS [76] verwendet in einem retrosynthetischen Ansatz die RECAP-Regeln [65], um Repräsentationen chemischer Bausteine und Verknüpfungsregeln zu generieren. Die Methode wurde auf die Molekül datenbank World Drug Index [77] angewandt, um eine kombinatorische virtuelle chemische Bibliothek mit 24.563 Repräsentationen chemischer Bausteine zu generieren. Die erzeugten kombinatorischen Bibliotheken können mit einem evolutionären Algorithmus und einer Pharmakophor-basierten Beschreibung der chemischen Bausteine [78] durchsucht werden.

Kürzlich wurden der quelloffenen Javabibliothek OpenChemLib [79] Funktionalitäten zur Erzeugung und Darstellung kombinatorischer Bibliotheken hinzugefügt. [80] Die Definitionen und Darstellungsform wurden an existierende Verfahren, [53] teilweise in dieser Dissertation vorgestellte Verfahren, [D2] angelehnt. Mithilfe eines Datensatzes 10.000 chemischer Bausteine des Unternehmens Enamine Ltd. [55] und einer Liste von 56 chemischen Reaktionen [60] wurde eine kombinatorische Bibliothek erzeugt, die $5,1 \times 10^{12}$ Moleküle implizit beschreibt.

Eine prominente Darstellungsform kombinatorischer Bibliotheken wird als Fragmentraum bezeichnet und wurde 2001 entwickelt. [53] In dieser Dissertation sprechen wir in diesem Zusammenhang von klassischen Fragmenträumen. Chemische Bausteine werden hier durch sogenannte Fragmente repräsentiert. Die Fragmente enthalten Atome, die als Platzhalter fungieren und Linker genannt werden. Jeder Linker hat einen speziellen Typ und über definierte Paare von kompatiblem Linktypen werden chemische Reaktionen repräsentiert und können Fragmente zu finalen Molekülen bzw. Produkten verknüpft werden. Ein Fragmentraum ist eine kombinatorische Bibliothek, da die Produkte nicht explizit enumeriert werden. Die algorithmische Methode FTrees-FS operiert auf Fragmenträumen und ist in der Lage dessen implizit beschriebene Moleküle zu durchsuchen. Die erste Bibliothek dieser Form [53] wurde mithilfe der Molekül datenbank World Drug Index [77] und der RECAP-Regeln [65] in einem retrosynthetischen Verfahren erzeugt. Hierbei enstanden 16.780 Fragmente. Durch die so enstandenen Verknüpfungsregeln beschreibt dieser Fragmentraum trotz der begrenzten Anzahl an Fragmenten eine unendliche Menge an Produktmolekülen. Später wurden weitere Fragmenträume in einem retrosynthetischen Ansatz mithilfe der BRICS Regeln erzeugt. [66] Mittlerweile existieren mehrere Fragmenträume, die mithilfe von Datensätzen chemischer Bausteine und Syntheseregeln, also ohne retrosynthetische Techniken erzeugt wurden. [81]–[87] Im Folgenden wollen wir

vier Bibliotheken genauer vorstellen, da sie in dieser Dissertation verwendet werden.

Der Enamine REAL Space [81] wurde mit einem Datensatz von über 104.000 chemischen Bausteinen des Unternehmens Enamine Ltd. und 156 chemischen Reaktionen, durch reactionSMARTS repräsentiert, erstellt. Die so entstandene kombinatorische Bibliothek beschreibt über 20 Milliarden Moleküle. Diese Anzahl ist mehr als doppelt so groß wie die Molekülmenge jeder enumerierten Bibliothek. [48] Der GalaXi Space [82] des Unternehmens WuXi AppTec beschreibt einen kombinatorischen Raum aus 2,3 Milliarden Produkten, der mithilfe von 155.000 chemischen Bausteinen und 30 Reaktionstypen aufgespannt wird. Für die Erzeugung der kombinatorischen Bibliothek CHEMriya [83] des Unternehmens Otava Ltd. wurde ein Datensatz von 30.000 chemischen Bausteinen und 44 chemischen Reaktionen verwendet. Durch das Aufspannen des kombinatorischen Raums werden 11 Millionen Produkte beschrieben. Die Moleküle dieser drei virtuellen chemischen Bibliotheken liegen zwar nicht enumeriert vor, Medizinalchemiker können aber eine Auswahl an Molekülen treffen und diese werden vom entsprechenden Unternehmen synthetisiert und geliefert. Zur Synthese werden die enkodierten chemischen Bausteine und Reaktionen verwendet, aus denen die ausgewählten Produkte der virtuellen Bibliothek gebildet wurden. Für den Enamine REAL Space wurde eine durchschnittliche Synthetisierbarkeit von über 80% ermittelt. [88], [89] Diese Möglichkeit der Ableitung von Syntheseregeln stellt, neben der weitaus größeren Menge an beschreibbaren Molekülen, einen Vorteil von kombinatorischen gegenüber enumerierten Bibliotheken dar. Der öffentlich zugängliche KnowledgeSpace [84] verwendet 117 chemische Reaktionen aus der Literatur und die chemischen Bausteine des eMolecules Datensatzes [59], um einen kombinatorischen Raum von über 100 Billionen Molekülen aufzuspannen. Damit übersteigt der KnowledgeSpace die drei vorgestellten kombinatorischen Bibliotheken der Chemieunternehmen um jeweils mindestens drei Größenordnungen. Allerdings ist die Synthetisierbarkeit seiner Moleküle nicht unbedingt gewährleistet, der Fokus der Bibliothek liegt eher in der Beschreibung einer theoretischen chemischen Wissensdomäne. Da die Reaktionen und chemischen Bausteine öffentlich zugänglich sind, kann für jedes Molekül der Bibliothek eine frei verfügbare theoretische Syntheseregel ermittelt werden.

2.3. Topologische Fragmenträume

In dieser Dissertation beschreiben wir *topologische Fragmenträume*, wie sie in [D2] eingeführt wurden. Sie bilden eine neue Darstellungsform kombinatorischer Bibliotheken.

Topologische Fragmenträume bauen in ihrer Definition auf den klassischen, in Kapitel 2.2 beschriebenen, Fragmenträumen auf.

2.3.1. Motivation und Abgrenzung von klassischen Fragmenträumen

Das Ziel eines Computerprogramms zur Durchsuchung oder Analyse einer kombinatorischen Bibliothek und das zentrale Paradigma dieser Dissertation ist es, nur auf den chemischen Bausteinen und Reaktionen zu operieren und dabei trotzdem den Suchraum der implizit beschriebenen Produkte zu erfassen. Hierfür muss das Verfahren in der Lage sein, alle relevanten Informationen über die Moleküle des Suchraums aus der Repräsentation der chemischen Bausteine und ihrer Verknüpfungsregeln zu extrahieren. Die gewählte Darstellungsform einer kombinatorischen Bibliothek muss diese Informationsgewinnung ermöglichen. Im Folgenden grenzen wir topologische Fragmenträume ausschließlich von klassischen Fragmenträumen ab. Die anderen Darstellungsformen kombinatorischer Bibliotheken in den Modulen SynthI [75] und OpenChemLib [79] sind strukturell identisch zu den Fragmenten, Linkern und kompatiblen Linkerpaaren klassischer Fragmenträume.

Der klassische Fragmentraum wurde für die computergestützte Durchsuchung mithilfe des algorithmischen Verfahrens FTrees-FS entwickelt. Hierbei werden Moleküle durch ihren sogenannten *feature tree*, [90] einen Graphen mit Baumstruktur, repräsentiert. In einem feature tree werden die Atome eines Rings des Moleküls zu einem Knoten zusammengefasst. Alle azyklischen Atome bilden ihren eigenen Knoten. Aus diesem Grund darf im klassischen Fragmentraum kein Ringschluss über mehrere Fragmente hinweg entstehen. Wäre dies der Fall, dann würde der entstandene Ring im finalen Molekül durch einen Knoten seines feature trees repräsentiert. In den einzelnen Fragmenten sind diese Atome aber noch azyklisch und werden dadurch im feature tree der Fragmente durch ihren eigenen Knoten repräsentiert. Dadurch approximiert die Kombination der feature trees der Fragmente den feature tree des finalen Produkts nicht ausreichend, der Suchraum der finalen Moleküle ist in diesem Fall nicht durch die Repräsentation der chemischen Bausteine und Reaktionen beschrieben. Für kleine Ringe und Ringsysteme ist es möglich separate Fragmente zu bilden und so auch Ringschlüsse in chemischen Reaktionen abzubilden. Allerdings entstehen so mehr neue Fragmente und Linker und für makrozyklische Ringschlüsse ist dieses Verfahren nicht mehr unbedingt möglich. Deshalb passen wir die Repräsentation chemischer Bausteine an, sodass Ringschlüsse über Fragmentgrenzen hinaus möglich sind und die chemischen Eigenschaften dieser Ringe innerhalb der Produkte aus der Repräsentation chemischer Bausteine ableitbar

ist. Dies wird für die in dieser Dissertation beschriebenen algorithmischen Ansätze von zentraler Bedeutung sein.

Wie in Kapitel 2.2 beschrieben, ist es möglich in einem klassischen Fragmentraum eine unendliche Menge an Molekülen mithilfe einer endlichen Anzahl Fragmente zu beschreiben. [53] Die Voraussetzung hierfür ist zB. ein Fragment mit zwei kompatiblen Linktypen. Dadurch ist es möglich das Fragment beliebig oft mit sich selbst zu verknüpfen und so Moleküle mit unbeschränkt wachsender Größe zu generieren. Ein Beispiel hierfür wäre die Fragmentdarstellung eines Peptids. Hierbei können durch häufige Verknüpfung an einer Amidbindung beliebig lange Polypeptid- oder Makropeptidverbindungen entstehen. Oft ist allerdings die Anzahl der zu verknüpfenden chemischen Bausteine begrenzt und durch die chemische Reaktionen festgelegt. Beispielsweise wird ein Molekül durch eine chemische Reaktion mit drei Komponenten oder durch zwei konsekutive chemische Reaktionen mit jeweils zwei Komponenten erzeugt. In beiden Fällen entsteht das Molekül aus drei chemischen Bausteinen. Diese Information ist im klassischen Fragmentraum nicht direkt kodiert, da verschiedene kompatible Paare von Linktypen in keiner direkten Beziehung zueinander stehen. Wir werden diese Information in einer Graphstruktur einfangen, die die Topologie eines Syntheseprotokolls beschreibt. Algorithmischen Verfahren wird es mithilfe dieser Graphstruktur möglich sein, Fragmente in sinnvollen Gruppen zu prozessieren und Produkte effizient zu bilden.

2.3.2. Repräsentation chemischer Bausteine

In diesem Abschnitt beschreiben wir die Definition von Fragmenten in topologischen Fragmenträumen. Sie repräsentieren die zur Erzeugung verwendeten chemischen Bausteine einer kombinatorischen Bibliothek. Fragmente beschreiben die Konfiguration aber auch die chemische Umgebung der Bausteine, die sie innerhalb der finalen Produkte nach Durchführung des Syntheseprotokolls aufweisen. Damit trägt diese Repräsentation zur Erfüllung des Paradigmas der Approximation des Produktraumes durch den Raum der Fragmente bei.

In Abbildung 2.1 ist ein Syntheseprotokoll beschrieben, in dem drei chemische Bausteine und zwei konsekutive Reaktionen verwendet werden. Durch die angewandten Reaktionen verändern sich die Eigenschaften einiger Atome innerhalb der chemischen Bausteine. Zusätzlich werden neue Bindungen zwischen ihnen geknüpft, um das Produkt zu generieren. Diese Veränderung halten wir in der Fragmentrepräsentation fest.

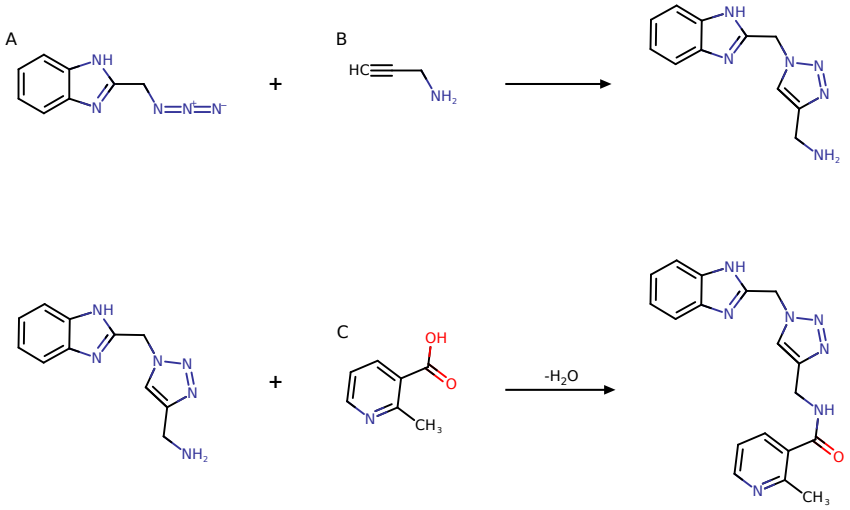


Abbildung 2.1.: Ein Syntheseprotokoll zur Erzeugung eines Produkts aus drei chemischen Bausteinen, die mit A, B und C bezeichnet sind. In einem ersten Syntheseschritt wird eine Diels-Alder Reaktion [91] auf die Bausteine A und B angewandt. In einer zweiten Reaktion wird durch eine Amidkopplung das Zwischenprodukt mit dem Baustein C verknüpft.

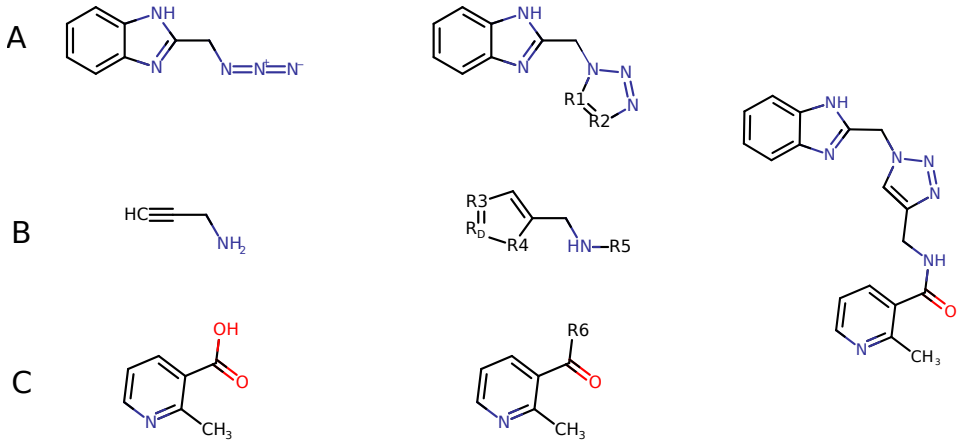


Abbildung 2.2.: Die Fragmentrepräsentation der chemischen Bausteien A, B und C aus Abbildung 2.1. Links sind jeweils die chemischen Bausteine abgebildet und mittig ihre Fragmentrepräsentation. Linkatome sind mit einem R und einer darauffolgenden Zahl gekennzeichnet. Ringplatzhalter sind mit R_D gekennzeichnet. Rechts ist das Produkt aus der Kombination der Fragmente abgebildet.

In Abbildung 2.2 zeigen wir die Fragmentrepräsentation der chemischen Bausteine aus Abbildung 2.1. Wie in klassischen Fragmenträumen wurden sogenannte Linker als

Platzhalter angefügt, um die Reaktionsstelle in den chemischen Bausteinen zu markieren. Allerdings werden zusätzlich mit Linkern neue Ringe geschlossen, wenn dies innerhalb der repräsentierten chemischen Reaktion auch der Fall ist. Das Fragment des chemischen Bausteins A enthält einen 5-Ring, der aus drei Stickstoffen des chemischen Bausteins und den Linkern $R1$ und $R2$ besteht. Die zwei Linker fungieren als Platzhalter für die zwei Kohlenstoffe des Triazols im finalen Produkt. Der 5-Ring wird als aromatisch markiert und die Ladung der drei Stickstoffe an ihre Ladung im Produkt angepasst. Für die Fragmentrepräsentation des Bausteins B wird ebenfalls ein 5-Ring gebildet. Hier wird, neben den Linkern $R3$ und $R4$ ein Ringplatzhalter R_D eingefügt. Ringplatzhalter dienen lediglich zum Auffüllen eines Rings im Fragment, sodass seine Anzahl Atome der Anzahl im korrespondieren Ring des Produkts entspricht. Falls durch die Hinzunahme von Linkern zu den Atomen des chemischen Bausteins zuviele Atome im Ring enthalten sind, werden mehrere Linker kontrahiert bis die Anzahl der Atome im Ring passend ist. Zusätzlich zu den gebildeten Ringen wird in den Fragmenten der Bausteine B und C jeweils ein Linker zur Repräsentation der Amidkupplung eingefügt. Der Linker $R5$ ersetzt ein Wasserstoffatom am Stickstoff im Baustein B. Der Linker $R6$ ersetzt die Hydroxygruppe des Bausteins C. Die ersetzen Atome werden bei der chemischen Reaktion von den Bausteinen abgespalten, deshalb sind sie in der Fragmentrepräsentation nicht mehr enthalten.

Ein Fragment beinhaltet somit Informationen des repräsentierten chemischen Bausteins, aber auch der verwendeten chemischen Reaktion. Deshalb kann ein topologischer Fragmentraum mehrere unterschiedliche Fragmente enthalten, die den gleichen chemischen Baustein in unterschiedlichen Syntheseprotokollen beschreiben. Durch diese Repräsentation enthält ein Fragment, neben Linker und Ringplatzhaltern, nur Atome, die auch in allen finalen Produkten enthalten sind, welche mithilfe des Fragments gebildet werden können. Die Valenz, Konnektivität, Ladung und Aromatizität seiner Atome entspricht den Eigenschaften der Atome innerhalb des Produkts. Daraus folgt, dass alle in einem Fragment enthaltenen chemischen Substrukturen auch im finalen Produkt enthalten sind. Diese Eigenschaft von Fragmenten in topologischen Fragmenträumen bezeichnen wir als *Substruktur-Teilmengenrelation*. Sie ist ein Grundbaustein der Erfüllung des algorithmischen Paradigmas und wird eine entscheidende Rolle für die Verfahren spielen, die wir in den weiteren Kapiteln dieser Dissertation vorstellen.

2.3.3. Topologiegraphen

Die so erzeugten Fragmente werden in eine Graphstruktur eingebettet, die wichtige Eigenschaften des Syntheseprotokolls beschreibt. Diese Graphstruktur [92] bezeichnen wir als *Topologiegraph*. Ein *topologischer Fragmentraum* besteht aus einem oder mehreren Topologiegraphen und den enthaltenen Fragmenten. Die Knoten des Topologiegraphen, auch Topologieknoten genannt, repräsentieren die Positionen an denen in der Synthese chemische Bausteine verwendet werden. Ein Knoten enthält einen oder mehrere Fragmente, die alle die gleiche Konfiguration von Linkern aufweisen. Die Fragmente eines Knotens stellen eine Gruppe chemischer Bausteine dar, die gleichberechtigt im Syntheseprotokoll verwendet werden können. Meistens handelt es sich um chemische Bausteine mit einer gemeinsamen Charakteristik, beispielsweise Carbonsäuren, die alle eine Carboxygruppe aufweisen. Die Kanten eines Topologiegraphen, auch Topologiekanten genannt, repräsentieren die Bindungen, die zwischen den chemischen Bausteinen während der Synthese geknüpft werden. Eine Kante enthält ein Paar von Knoten des Topologiegraphen. Die repräsentierte chemische Bindung wird zwischen den Gruppen chemischer Bausteine geknüpft. Der Typ der chemischer Bindung wird ebenfalls in der Kante abgespeichert. Es wird zwischen Einfach-, Doppel- und Dreifachbindungen, sowie Bindungen in aromatischen Ringen unterschieden. Zusätzlich enthält eine Topologiekante jeweils einen Linkertyp für jede der zwei Topologieknoten der Kante. Die Fragmente der Knoten enthalten alle einen Linker des jeweiligen Typs. Diese Linker markieren die Stelle, an der die von der Kante repräsentierte Bindung geschlossen wird. Bei der Bildung eines Produkts werden diese Linker aus den Fragmenten entfernt und eine Bindung des angegeben Typs zwischen ihren Schweratomnachbarn geknüpft. Dieses Linkerpaar entspricht einem Paar von kompatiblen Linkern im klassischen Fragmentraum. Um ein Molekül der repräsentierten kombinatorischen Bibliothek zu erzeugen, wird aus jedem Knoten des Topologiegraphen genau ein Fragment ausgewählt. Alle Ringplatzhalter und Linker werden entfernt und die Bindungen zwischen den Fragmenten mithilfe der Information aus den Kanten des Topologiegraphen geknüpft. Topologische Fragmenträume können mithilfe des Programms SpaceLight [D2] erzeugt werden. Hierbei werden die Fragmente und Topologiegraphen aus vom Nutzer angegeben Syntheseprotokollen und Datensätzen chemischer Bausteine abgeleitet. Der erzeugte topologische Fragmentraum wird als SQLite [93] Datenbank abgespeichert und verwendet das modulare Datenbankschema der Codebasis NAOMI. [54] Wir beschreiben die Methode detailliert in Anhang B.1.

In Abbildung 2.3 zeigen wir einen Topologiegraphen, der das Syntheseprotokoll aus Abbildung 2.1 beschreibt. Die oberen Fragmente in (c) der Knoten des Topologiegraphen

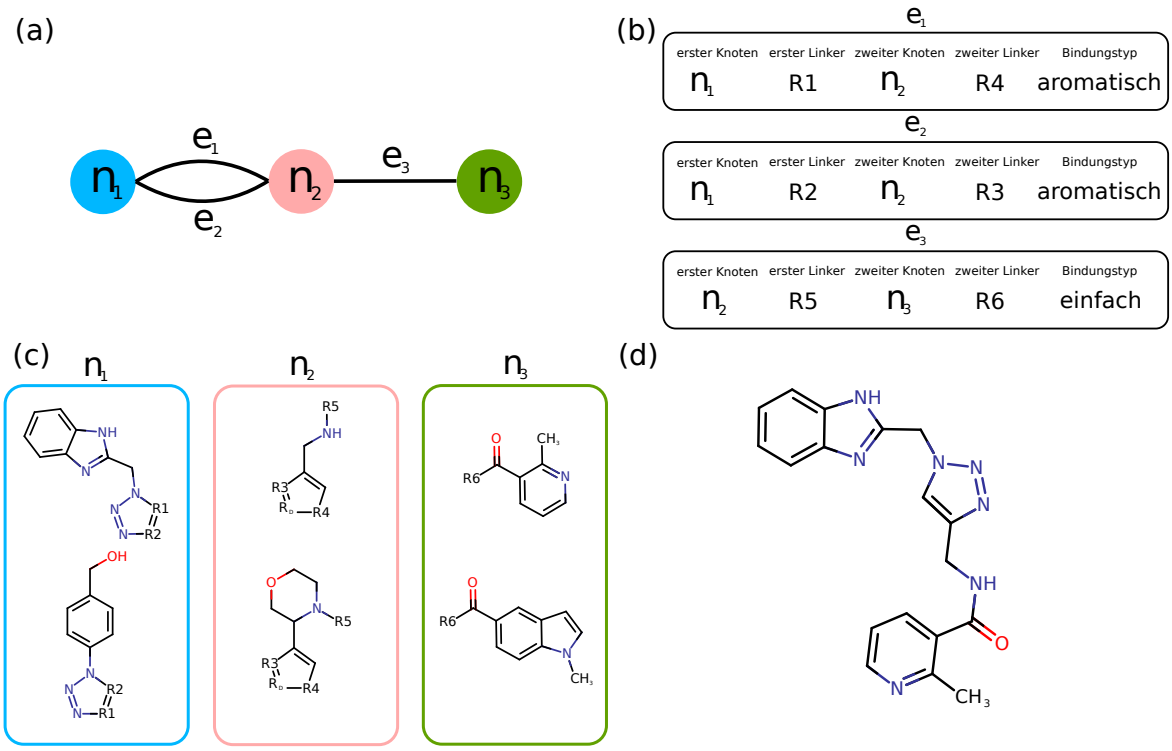


Abbildung 2.3.: (a) Ein Topologiegraph mit drei Knoten und drei Kanten. (b) Für jede Kante des Topologiegraph das Paar adjazenter Knoten, kompatibler Linker und der Bindungstyp der repräsentierten Bindung. (c) Die in den Knoten enthaltenen Fragmente. (d) Das Produkt aus Abbildung 2.1. Es entsteht aus den jeweils ersten Fragmenten der Knoten des Topologiegraphen. Aus [D2] entnommen, auf Deutsch übersetzt und angepasst.

sind die Fragmentrepräsentationen der chemischen Bausteine A, B und C aus Abbildung 2.2. Pro Knoten haben wir ein weiteres Fragment mit der gleichen Konfiguration von Linkern hinzugefügt. Durch den Topologiegraphen ist visuell, sowie computergestützt, leicht feststellbar, welche topologischen Eigenschaften die durch den Graphen beschriebenen Molekül gemein haben. Durch das hier verwendete Syntheseprotokoll entsteht beispielsweise ein aromatischer Ringschluss und eine Kupplung über eine Einfachbindung. Jedes erzeugte Produkt weist somit diese Eigenschaften auf und entsteht aus drei chemischen Bausteinen. Diese Informationen werden von den algorithmischen Verfahren genutzt, die in Kapitel 4, 5 und 6 beschrieben werden.

2.3.4. Ausblick

Die in diesem Kapitel beschriebenen topologischen Fragmenträume sind eine Darstellungsform kombinatorischer virtueller chemischer Bibliotheken. Sie bauen auf der Definition klassischer Fragmenträume auf und erweitern diese um Funktionalitäten, die für alle in dieser Disseration vorgestellten algorithmischen Verfahren benötigt werden. Hierbei wird die durch ein Syntheseprotokoll definierte Topologie der generierten Moleküle eingefangen. In der hier beschriebenen Implementation werden stereochemische Eigenschaften der chemischen Bausteine und durchgeführten Reaktionen nicht berücksichtigt. Dazu relevante Informationen können in reactionSMARTS Ausdrücken und den Datenformaten der chemischen Bausteine angegeben werden. Es ist prinzipiell möglich den Prozess der Erzeugung von Topologiegraphen und Fragmenten darauf anzupassen.

Das verwendete Datenbankschema der Codebasis NAOMI ist modular aufgebaut, da es für unterschiedlichste Anwendungen benutzt wird. Deshalb ist es leicht möglich zusätzliche Informationen über chemische Bausteine oder Reaktionen zu hinterlegen. Beispielsweise könnten Daten über Erfolgswahrscheinlichkeit oder Ausbeute einer chemischen Reaktion in der Datenbank abgespeichert werden. Denkbar sind auch Informationen über Hersteller, Preise oder Experimentaldaten chemischer Bausteine, die ebenfalls in der Datenbank dokumentiert werden. Die in dieser Dissertation beschriebenen Verfahren verwenden lediglich strukturelle und topologische Informationen der chemischen Bausteine und implizit beschriebenen Moleküle. Für die Zukunft ist aber auch eine Anwendung topologischer Fragmenträume für Verfahren denkbar, die räumliche Konformationen von Molekülen berücksichtigen. In diesem Fall könnten für jedes Fragment bereits Konformationen vorprozessiert oder aus Experimentaldaten extrahiert und in der Datenbank abgespeichert werden.

3. Molekulare Fingerabdrücke in kombinatorischen Bibliotheken

In Kapitel 1 und 2 wurden verschiedene Datenformate [32], [34] zur Abspeicherung von Molekülen vorgestellt. Diese Formate enthalten zumeist Informationen über die Struktur des Moleküls sowie eventuell zusätzlich dessen räumliche Konformation. Oft sind allerdings bei der Auswahl eines Kandidaten für den Wirkstoffentwurf andere chemische Eigenschaften der Moleküle interessant. Um die einfache Wissensextraktion für ein algorithmisches Verfahren im Rahmen des Wirkstoffentwurfs zu ermöglichen, wurden Molekulardeskriptoren entwickelt. [38] Sie beschreiben z.B. Informationen über physikochemische Eigenschaften, [94] enthaltene chemische Substrukturen [95]–[98] oder geometrische Kennzahlen. [99]

In diesem Kapitel wird eine Gruppe von Molekulardeskriptoren behandelt, die als molekulare Fingerabdrücke bezeichnet werden. Ein molekularer Fingerabdruck besteht aus einer Liste von Zahlen, einem Vektor oder einer Bitfolge. Hierbei steht jede Zahl, jeder Eintrag oder jedes gesetzte Bit für eine bestimmte Eigenschaft, die das repräsentierte Molekül aufweist. Diese Darstellungsform, speziell als Bitfolge, ist sehr kompakt und kann dadurch effizient algorithmisch genutzt werden. Deshalb finden molekulare Fingerabdrücke eine breite Anwendung in der Chemieinformatik von der Suche nach ähnlichen Molekülen im Rahmen des virtuellen Screenings, [100], [101] über die Gruppierung von Molekülen einer virtuellen chemischen Bibliothek [40], [102] bis hin zum maschinellen Lernen. [103]

3.1. Etablierte Methoden für molekulare Fingerabdrücke

Alle Verfahren zur Erzeugung molekularer Fingerabdrücke haben die kompakte Darstellungsform durch Bitfolgen, Zahlenlisten oder Vektoren gemein. Allerdings unterschiedenen sie sich in den Moleküleigenschaften, die sie betrachten. Im Folgenden wird ein Überblick

über existierende Verfahren gegeben und die vorgestellten Methoden werden nach der Art von Eigenschaften, die sie bezüglich eines Moleküls einfangen, gruppiert.

Wörterbuch-basierte Methoden für molekulare Fingerabdrücke prüfen die An- oder Abwesenheit einer fest definierten Menge von chemischen Eigenschaften, die nicht vom jeweils betrachteten Molekül abhängt. Jede Eigenschaft wird durch ein Bit in einer Bitfolge repräsentiert, wobei das Bit gesetzt wird, wenn das beschriebene Molekül die entsprechende Eigenschaft aufweist. MACCS Structural Keys [96] verwenden SMARTS [37] Ausdrücke, um das Vorkommen von 166 bzw. 960 chemischer Substrukturen in einem Molekül zu überprüfen. Die PubChem Fingerprint Methode [104] operiert auf die gleiche Weise mithilfe von SMARTS Ausdrücken, enthält aber 881 chemische Substrukturen in seinem Wörterbuch.

Topologische molekulare Fingerabdrücke enkodieren ebenfalls chemische Substrukturen. Allerdings wird nicht die An- oder Abwesenheit festgelegter chemischer Substrukturen aus einem Wörterbuch überprüft. Stattdessen werden alle Subgraphen des Molekulargraphen eines Moleküls enumeriert, die meistens eine bestimmte Graphstruktur aufweisen müssen. Ein Subgraph enthält eine Teilmenge der Schweratome eines Molekulargraphen und zusätzlich eine Teilmenge der Bindungen zwischen diesen Atomen, sodass eine zusammenhängende Graphstruktur entsteht. Unabhängig von den enthaltenen Atomen und Bindungen werden keine zwei Subgraphen als äquivalent betrachtet, wenn sie an einer anderen Position im Molekulargraphen vorkommen. Ein Subgraph wird als *induziert* bezeichnet, wenn alle Bindungen des Moleküls zwischen den Atomen der Teilmenge auch im Subgraphen enthalten sind. Falls in dieser Dissertation nicht spezifiziert, sind induzierte Subgraphen gemeint. Für jeden enumerierten Subgraphen wird mithilfe einer Hashfunktion ein Identifikator in Form einer Zahl generiert. Hierbei fangen die Methoden unterschiedliche Eigenschaften der Atome und Bindungen eines Subgraphen ein. Die so generierten Identifikatoren repräsentieren eine chemische Substruktur und bilden den molekularen Fingerabdruck als sortierte Menge von Zahlen. Zusätzlich kann aus dieser Menge mithilfe einer Division mit Rest eine Bitfolge generiert werden. Dieser Prozess wird als Faltung des Fingerabdrucks bezeichnet.

Der topologische molekulare Fingerabdruck Atom Pairs [95] bildet für alle Paare von Atomen eines Moleküls einen Identifikator aus ihren Elementen und ihrer topologischen Distanz innerhalb des Molekulargraphen. Der Fingerabdruck TopologicalTorsion [97] betrachtet alle Pfade aus vier Atomen und drei Bindungen eines Moleküls. Für jeden dieser Pfade wird ein Identifikator aus den Elementen der Atome, der Anzahl ihrer

Schweratomnachbarn und π-Elektronen gebildet. Der molekulare Fingerabdruck des Unternehmens Daylight [105] enumeriert alle Pfade eines Moleküls mit einer Anzahl von einem bis vier Atomen. Pro Pfad wird ein Identifikator aus dem Elementen der Atomen und den Bindungestypen innerhalb des Pfades generiert. Der Fingerabdruck der Firma OpenEye [106] enumeriert alle Baumstrukturen, die höchstens eine vom Nutzer definierte Anzahl Atome enthalten. Die Atom- und Bindungseigenschaften, die in den Identifikator einer Baumstruktur einfließen sollen, können ebenfalls vom Nutzer spezifiziert werden.

Die quelloffene Python-Bibliothek RDKit [73] enthält ebenfalls einen eigenen topologischen molekularen Fingerabdruck. Für die Erzeugung des RDKit Fingerprint werden alle Subgraphen des Molekulargraphen eines Moleküls enumeriert, die eine Anzahl Bindungen innerhalb eines vom Nutzer definierten Intervalls aufweisen. Für den RDKit Fingerprint eines Moleküls werden auch nicht induzierte Subgraphen enumeriert. Das unterscheidet das Verfahren von der in dieser Dissertation in Kapitel 3.2 vorgestellten Methode zur Berechnung eines molekularen Fingerabdrucks. Der Identifikator eines Subgraphen berücksichtigt das Element und die Aromatizität der enthaltenen Atome sowie die Typen der enthaltenen Bindungen.

Die wohl prominenteste Methode zur Berechnung topologischer molekularer Fingerabdrücke ist die der Extended Connectivity Fingerprints (ECFP) [98]. Hierbei werden sogenannte zirkuläre Subgraphen enumeriert, die aus allen Atomen mit einer Distanz kleiner oder gleich einer oberen Schranke zu einem zentralen Atom bestehen. Für die Berechnung des ECFP eines Moleküls wird pro Schweratom als Zentrum und jede obere Schranke zwischen null und einem durch den Nutzer definierten maximalen Radius ein zirkulärer Subgraph gebildet. In dieser Dissertation wird der Notation aus der Literatur gefolgt und der ECFP eines Moleküls mit maximalem Radius x mit ECFP2x bezeichnet, wobei $2x$ einen Durchmesser also die maximale Distanz zwischen zwei Atomen innerhalb der enumerierten zirkulären Subgraphen bezeichnet. Für jeden Subgraphen wird ein Identifikator durch eine eindeutige Sortierung und Verwendung einer Hashfunktion gebildet. Hierbei wird die Anzahl der Schweratomnachbarn eines Atoms, seine Valenz, seine Anzahl an verbundenen Wasserstoffen, sein Element, sein Gewicht, seine Ladung und seine Zugehörigkeit zu einem Ring berücksichtigt.

Ein Pharmakophor beschreibt die sterischen und elektrostatischen Eigenschaften eines Atoms oder einer chemischen Substruktur, die zu einer pharmakologischen Wirkung des enthaltenen Moleküls führen. [1] Pharmakophor-basierte molekulare Fingerabdrücke enthalten meist Informationen über die topologische oder räumliche Anordnung der

verschiedenen Pharmakophore eines Moleküls. Zur Erzeugung des Functional-Class Fingerprint (FCSFP) [98] werden die gleichen zirkulären Subgraphen enumeriert wie zur Berechnung des ECFP. Allerdings werden für die Berechnung des Identifikatoren eines Subgraphen sechs Pharmakophor-basierte Eigenschaften eines Atoms berücksichtigt. Ob das Atom ein potenzieller Protonenakzeptor oder ein Protonendonator einer Wasserstoffbrückenbindung ist, ob das Atom positiv oder negativ ionisierbar ist und ob es aromatisch oder ein Halogen ist. CATS [78] ist eine Technik bei der, ähnlich wie bei der Erzeugung eines Atom Pairs Fingerabdrucks [95], die paarweise Distanz von Schweratomen eines Moleküls bestimmt wird. Anders als beim Atom Pairs Fingerabdruck werden hier allerdings nur Schweratome betrachtet, die potenzielle Protonenakzeptoren, Protonendonatoren, positiv geladen, negativ geladen oder lipophil sind. Der Identifikator eines Paares von Schweratomen betrachtet die Distanz sowie Klassifizierung der Atome in diese fünf Gruppen. PharmPrint [107] ist eine Methode zur Erzeugung eines Pharmakophor-basierten molekularen Fingerabdrucks. Das Verfahren betrachtet die sechs Pharmakophortypen der CATS-Methode und zusätzlich einen siebten Typ, falls keine der Pharmakophor-Klassifizierungen auf das Atom zutrifft. Nun werden verschiedene räumliche Konformationen des Moleküls generiert und pro Konformation alle Tripel von Atomen gebildet. Abhängig von den Distanzen zwischen den Atomen und ihren Pharmakophortypen wird ein Identifikator gebildet. Durch diese Herangehensweise enthält der molekulare Fingerabdruck sowohl Informationen über pharmakologische sowie räumliche Eigenschaften eines Moleküls.

Eine weitere Klasse molekularer Fingerabdrücke ist die der Interaktionsfingerabdrücke. Methoden zu ihrer Erzeugung betrachten die Wechselwirkungen zwischen einem Molekül und einem Protein. Sie unterscheiden sich damit grundlegend von allen anderen hier beschriebenen Verfahren, die lediglich einzelne Moleküle betrachten. Das Verfahren zur Erzeugung eines Structural Interaction Fingerprint (SIFt) [108] bestimmt Interaktionspunkte in einem Protein-Ligand-Komplex und klassifiziert diese nach sieben Eigenschaften. So entsteht pro Interaktion eine Bitfolge, die in einem Fingerabdruck zusammengefasst werden. Eine weitere Methode ist die der Structural Protein-Ligand Interaction Fingerprints (SPLIF) [109]. Bei der Erzeugung werden Paaren von Ligand- und Proteinatomen bestimmt, die eine oberen Schranke in ihrer räumlichen Distanz nicht überschreiten und als Interaktionsstelle interpretiert werden. Diese werden als Zentrum für zirkulärer Subgraphen verwendet, um Identifikatoren der parametrisierten Methode ECFP2 zu generieren. Die erzeugten Identifikatoren werden zum SPLIF des Protein-Ligand-Komplexes zusammengefasst. Zusätzlich werden pro zirkulärem Subgraphen die

Koordinaten der enthaltenen Atome abgespeichert für eine spätere Überlagerung.

3.2. Connected Subgraph Fingerprints

In diesem Kapitel wird die Methode zur Erzeugung von Connected Subgraph Fingerprints (CSFP) behandelt, wie sie in [D1] eingeführt wurden. Der CSFP eines Moleküls ist ein topologischer molekularer Fingerabdruck, der alle chemischen Substrukturen eines Moleküls betrachtet. Er ist alleinstehend zur Ähnlichkeitssuche im virtuellen Screening geeignet. Seine Hauptaufgabe innerhalb dieser Dissertation ist allerdings die Anwendung als Werkzeug für kombinatorische Verfahren in topologischen Fragmenträumen, die in den Kapiteln 4, 5 und 6 beschrieben wird.

3.2.1. Motivation

Die in der Literatur existierenden Verfahren zur Erzeugung molekularer Fingerabdrücke wurden für die Anwendung in enumerierten Bibliotheken entwickelt. Chemische Bausteine, wie sind in kombinatorischen Bibliotheken repräsentiert werden, enthalten durchschnittlich weniger Atome und Bindungen als ein Molekül einer enumerierten Bibliothek. Für eine feingranulare Beschreibung, sollte ein topologischer molekularer Fingerabdruck also mehr chemische Substrukturen betrachten und diese nach den Eigenschaften ihrer Atome und Bindungen unterscheiden. Zusätzlich muss eine Methode für topologische molekulare Fingerabdrücke im Kontext von kombinatorischen Bibliotheken in der Lage sein, mithilfe der Fingerabdrücke chemischer Bausteine den Fingerabdruck eines abgeleiteten Produkts zu approximieren. Auf diese Weise ist die Methode in der Lage das algorithmische kombinatorische Paradigma zu erfüllen. Diese Beobachtung wird in Kapitel 3.2.5 weiter ausgeführt. Weiterhin sollten die generierten topologischen molekularen Fingerabdrücke in der Lage sein im Rahmen einer Ähnlichkeitssuche Kandidaten für den Wirkstoffentwurf zu identifizieren. Im Folgenden wird der Ansatz zur Generierung von Connected Subgraph Fingerprints (CSFP) vorgestellt und diskutiert, warum er die beschriebenen Anforderungen erfüllt.

3.2.2. Erzeugung

Der CSFP eines Moleküls beschreibt alle seine chemischen Substrukturen, deren Anzahl enthaltener Atome in einem vom Nutzer definieren Intervall $[x, y]$ liegt. Der so generierte CSFP wird als CSFP$x.y$ bezeichnet. Diese Notation gilt auch für alle in Kapitel 3.2.3 eingeführten Varianten.

Analog zu den in Kapitel 3.1 beschriebenen Methoden zur Generierung topologischer molekularer Fingerabdrücke, kann die Erzeugung des CSFP eines Moleküls in zwei Abschnitte unterteilt werden. Zuerst wird eine Menge von Subgraphen des Molekulargraphen enumeriert. Danach wird für jeden enumerierten Subgraphen ein Identifikator berechnet und diese werden zum molekularen Fingerabruck zusammengefasst. Allerdings werden für die Erzeugung des CSFP alle induzierten Subgraphen enumeriert. Dies erfolgt mit dem in [D1] beschriebenen CONSENS-Algorithmus. Hierfür wird zunächst eine beliebige totale Ordnung auf den Schweratomen des Molekulargraphen eingeführt. Danach werden alle zusammenhängenden Atommengen in einem Rücksetzverfahren rekursiv enumeriert, die auschließlich Schweratome beschreiben. In jedem rekursiven Schritt wird die betrachtete Atommenge um einen Kandidaten erweitert, der innerhalb der Ordnung möglichst klein ist und nicht bereits in einem vorherigen Schritt ausgeschlossen wurde. Durch dieses effiziente Verfahren werden keine Atommengen mehrfach betrachtet. Ein Beweis der Korrektheit und der asymptotischen Laufzeit ist in [D1] ausgeführt. Der CONSENS-Algorithmus kann mit beliebigen Filtern für die enumerierten Atommengen verwendet werden. Für die Erzeugung des CSFP wird nach der vom Nutzer spezifizierten unteren und oberen Schranke für die Anzahl der Atom gefiltert. In Kapitel 4.3.1 wird der CONSENS-Algorithmus mit anderen Filtern angewandt. Aus jeder zusammenhängenden Atommenge ergibt sich direkt ein induzierter Subgraph aus allen Bindungen, die zwischen den Atomen der Menge verlaufen. Im Folgenden sind immer induzierte Subgraphen gemeint auch wenn sie nicht explizit als induziert angegeben sind.

In Abbildung 3.1 ist das CONSENS Verfahren beispielhaft beschrieben. Die zusammenhängende Atommenge wird mit dem Atom mit Rang zwei initiiert. Diese Menge bildet bereits ein enumerierten Subgraphen, falls die vom Nutzer angegeben untere Schrank für die Anzahl Atome auf eins gesetzt wurde. Alle Nachbarn des Atoms, die einen höheren Rang in der Ordnung aufweisen, werden der Kandidatenmenge hinzugefügt. Das Atom mit Rang eins ist in der Ordnung kleiner als das zur Initiierung verwendete Atom und wird deshalb für die weitere Enumeration ausgeschlossen. Auf diese Weise wird die zusammenhängende Atommenge der Ränge $\{1, 2\}$ nicht mehrfach enumeriert. In

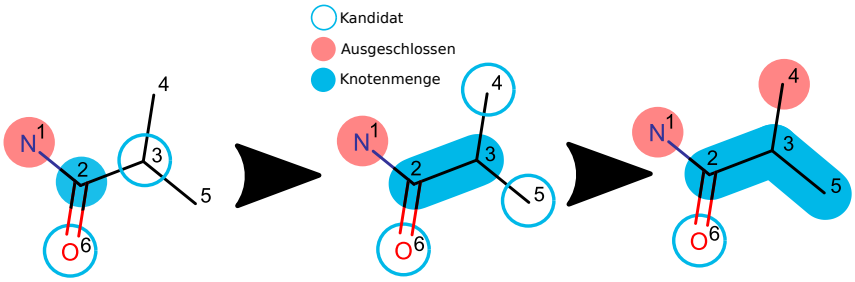


Abbildung 3.1.: Zwei Schritte des CONSENS-Algorithmus angewandt auf Isobutyramid. Die momentan enumerierte zusammenhängende Atommenge ist in türkis und die Menge ausgeschlossener Atome in rot dargestellt. Die Menge der Kandidatenknoten ist türkis umrandet. Die Nummerierung der Atome stellt die eingeführte Ordnung dar. Aus [D1] entnommen und auf Deutsch übersetzt.

den nächsten Schritten werden beispielhaft die Atome mit Rang drei und fünf zur Atommenge hinzugefügt, die im jeweiligen Schritt zur Kandidatenmenge gehören. Zusätzlich werden alle neuen Nachbarn der zusammenhängenden Atommenge zur Kandidatenmenge hinzugefügt, wenn sie noch nicht ausgeschlossen wurden. Bei der Wahl des Atoms mit Rang fünf wird das Kandidatenatom mit Rang vier ausgeschlossen, da es in der Ordnung kleiner ist. Damit wird die doppelte Enumeration der Atommenge mit Rängen $\{2, 3, 4, 5\}$ verhindert.

Im nächsten Schritt wird für jeden Subgraphen ein Identifikator generiert, der aus einer Zahl zwischen 0 und $2^{64} - 1$ besteht. Zunächst wird für jedes einzelne Schweratom des Subgraphen ein eigener Identifikator gebildet. Für diesen wird eine Menge von Eigenschaften zusammengefasst, die durch eine der in Kapitel 3.2.3 beschriebenen Varianten des CSFP definiert ist. Die Atome des Subgraphen werden durch eine Adaption des CANON-Algorithmus [110] in eine kanonische Ordnung überführt. Anschließend werden die Atome, der kanonischen Ordnung folgend, traversiert um den Identifikator des Subgraphen zu generieren. Das genaue Vorgehen ist in Anhang B.2 beschrieben. Dieses Verfahren zur Repräsentation von Subgraphen mithilfe eines Identifikators kann auch auf ausgewählte oder einzelne Subgraphen angewandt werden und wird in den Kapiteln 5 und 6 verwendet.

Tabelle 3.1.: Die betrachteten Subgraphen, Atom- und Bindungseigenschaften der CSFP Varianten. Stereozentren werden nach den CIP-Regeln [111] bestimmt. Bei der Eigenschaft 'generische aromatische Valenz' wird jedem aromatischen Atom die gleiche eindeutige Valenz zugewiesen. Aus [D2] entnommen, auf Deutsch übersetzt, angepasst und erweitert.

Kategorie	Eigenschaft	CSFP	fCSFP	tCSFP	iCSFP	gCSFP	pCSFP
Subgraphen	induziert	x	x	x	x	x	
	Pfad						x
Atom	Element	x	x	x	x	x	x
	Konnektivität	x	x	x		x	x
	Konnektivität in Subgraph	x	x		x		x
	Valenz	x	x			x	x
	Valenz in Subgraph	x	x		x		x
	generische aromatische Valenz					x	
	Aromatizität				x		
	π Elektronen			x			
	Ringzugehörigkeit	x	x				x
	Stereozentrum	x					x
Bindung	Bindungstyp	x	x		x		x

3.2.3. Varianten

Wie in Kapitel 3.2.2 beschrieben, wird pro Schweratom des Molekulargraphen ein Identifikator gebildet. Dieser Identifikator enthält Informationen über das beschriebene Atom. Welche Informationen berücksichtigt werden, bestimmt maßgeblich den Charakter des generierten molekularen Fingerabdrucks. Aus diesem Grund beschreiben wir in [D1] und [D2] mehrere Varianten des CSFP eines Moleküls, die sich aus unterschiedlichen berücksichtigten Atom- und Bindungseigenschaften ergeben. Neben dem klassischen CSFP existieren die folgenden Varianten: Der *fCSFP* ist eine feingranulare Variante, die für die Anwendung in topologischen Fragmenträume optimiert ist. Der *tCSFP* basiert auf den Atomeigenschaften des in Kapitel 3.1 beschriebenen TopologicalTorsion Fingerabdrucks. Der *iCSFP* verwendet ausschließlich Eigenschaften, die unabhängig von der Umgebung des Subgraphen sind. Der *gCSFP* ist eine grobgranulare Variante des CSFP. Die letzte Variante, der *pCSFP*, betrachtet nur Subgraphen mit Pfadstruktur und verwendet die Atom- und Bindungseigenschaften des klassischen CSFP. In Tabelle 3.1 sind die sechs Varianten des CSFP mit ihren betrachteten Eigenschaften beschrieben. Die klassische

CSFP Variante betrachtet die meisten Eigenschaften und liefert so einen sehr feingranularen molekularen Fingerabdruck. Der pCSFP ist auf die Betrachtung von Subgraphen mit Pfadstruktur eingeschränkt, weshalb sein Fingerabdruck eine Teilmenge der Identifikatoren des klassischen CSFP enthält. Die fCSFP Variante unterscheidet sich von der klassischen CSFP Variante nur in der Ausschließung stereochemischer Eigenschaften. Wie in Kapitel 2.3.4 beschrieben, fangen topologische Fragmenträume keine stereochemischen Eigenschaften ein. Da die fCSFP Variante speziell für die Anwendung in diesem Kontext entwickelt wurde, beachtet sie deshalb auch keine stereochemischen Eigenschaften. Die iCSFP Variante betrachtet nur Eigenschaften des Subgraphen selbst und der erzeugte Identifikator ist unabhängig von der Umgebung des Subgraphen. Damit generiert diese Variante einen Molekulardeskriptor, der in einem Kontext angelehnt an die Suche nach maximalen gemeinsamen Teilstrukturen (*maximum common substructure*(MCS)) [36] verwendet werden kann. Die Varianten tCSFP und gCSFP betrachten weniger Eigenschaften als der klassische CSFP oder die fCSFP Variante. Sie bieten damit eine grobgranularere Beschreibung in den von ihnen generierten molekularen Fingerabdrücken, wodurch sie sich potenziell besser für die Suche nach unterschiedlichen Molekülgerüsten eignen. [112]

3.2.4. Evaluation

In [D1] verwenden wir einen von Riniker und Landrum eingeführten Benchmark [112] für molekulare Fingerabdrücke im Rahmen einer Ähnlichkeitssuche. Für die Ähnlichkeitsuche werden für ein Anfragemolekül und alle Moleküle des Datensatzes molekulare Fingerabdrücke berechnet und der paarweise Ähnlichkeitswert zum Anfragemolekül mithilfe des Tanimotokoeffizienten [113] bestimmt. Das Verfahren verwendet Experimentaldaten aus dem Maximum Unbiased Validation (MUV) Datensatz [114], dem Directory of Useful Decoys (DUD) Datensatz [115] und den Datenbanken ChEMBL [43] und ZINC [44]. Zur Bewertung verwenden wir zwei Metriken. Der Anreicherungsfaktor (*early enrichment factor*(EF)) [116] bei 2% misst den Anteil an den 2% Molekülen mit dem höchstens Ähnlichkeitswert, deren Experimentaldaten eine Bioaktivität aufweisen. Der Wert für die Fläche unter der Kurve (*area under the curve* (AUC)) misst das Integral der Isosensitivitätskurve (*ROC curve*). [117] Der AUC-Wert beschreibt die insgesamte Verteilung von Molekülen mit gemessener Bioaktivität innerhalb der nach absteigendem Ähnlichkeitswert sortierten Moleküle. Riniker und Landrum stellten fest, dass der AUC-Wert einer Methode zur Erzeugung eines molekularen Fingerabdrucks mit dessen Potenzial für die Suche nach unterschiedlichen Molekülgerüsten korreliert ist. [112]

Die Evaluation der verschiedenen Varianten des CSFP wurde mithilfe der drei Datensätze und zwei Evaluationsmetriken durchgeführt und mit den Ergebnissen der Methoden zur Erzeugung der Atom Pairs, TopologicalTorsion und ECFP Fingerabdrücke verglichen. Da die fCSFP Variante erst in [D2] eingeführt wurde, wird sie bei dem in [D1] durchgeführten Benchmark nicht berücksichtigt. Es zeigte sich zunächst, dass die Varianten tCSFP und gCSFP durchschnittlich die höchsten AUC-Werte aller CSFP Varianten aufweisen. Dies unterstreicht ihren grobgranularen Charakter und ihre Eignung für die Detektion von bioaktiven Molekülen mit unterschiedlichen Molekülgerüsten. Der CSFP und pCSFP erzielten mit ihrer feingranularen und spezifischen Beschreibung der chemischen Eigenschaften eines Moleküls durchschnittlich die höchsten EF-Werte. In Abbildung

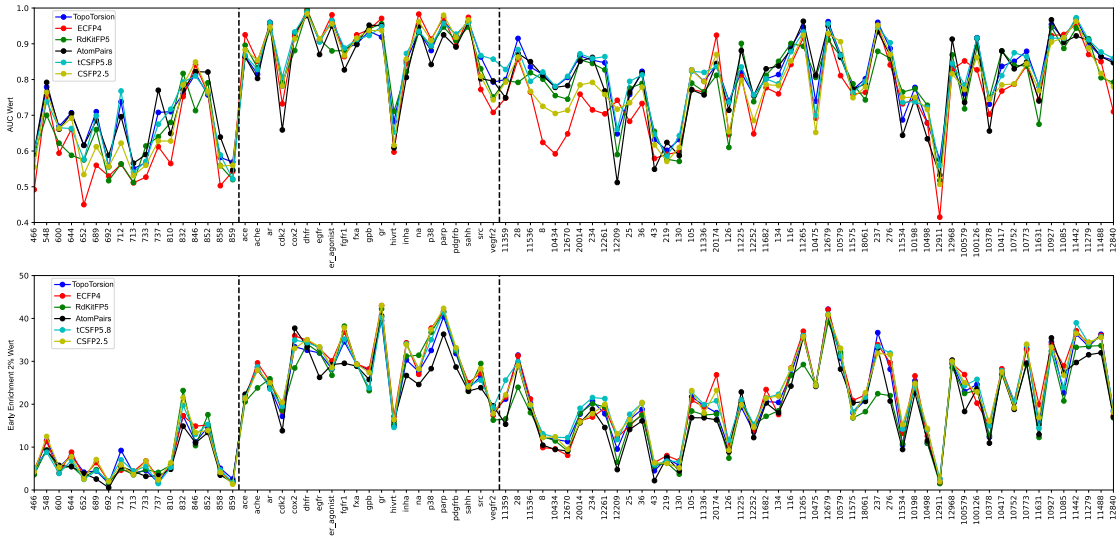


Abbildung 3.2.: Die AUC- (oben) und EF-Werte (unten) der Methoden zur Erzeugung molekularer Fingerabdrücke pro Datensatzeintrag. Links der gestrichelten Linien sind die Einträge des MUV-Datensatzes, in der Mitte die des DUD-Datensatzes und rechts der gestrichelten Linien die Einträge des ChEMBL und ZINC Datensatzes. Der RDKit5 Fingerabdruck betrachtet alle (auch nicht induzierten) Subgraphen mit bis zu fünf Bindungen. Aus [D1] entnommen, angepasst und ins Deutsche übersetzt.

3.2 werden die AUC- und EF-Werte der in [112] am besten bewerteten Methoden und Parametrisierungen mit dem CSFP2.5 und tCSFP5.8 verglichen. Diese zwei Varianten des CSFP und Parametrisierungen wurden wegen ihrer durchschnittlich hohen Werte auf alle drei Datensätzen und Evaluationsmetriken ausgewählt. Die Höhe der Werte hängt deutlich stärker vom jeweiligen Datensatzeintrag ab, als von der verwendeten Methode zur Erzeugung molekularer Fingerabdrücke. Jedoch zeigt dieser Benchmark, dass der CSFP

und seine Varianten durch ihre hohe Parametrisierbarkeit vergleichbare und teilweise bessere Ergebnisse erzielen als die Methoden Atom Pairs, TopologicalTorsion, ECFP oder der RDKit Fingerabdruck. Somit eignet sich die hier vorgestellte Methode zur Ähnlichkeitssuche und wird der Grundstein für die in Kapitel 4 beschriebenen Verfahren sein.

3.2.5. Anwendung in kombinatorischen Bibliotheken

Die Evaluation in Kapitel 3.2.4 zeigt die Anwendbarkeit des CSFP für die Ähnlichkeitssuche in enumerierten Bibliotheken. In Kapitel 4 wird gezeigt, dass der CSFP ebenfalls für die Ähnlichkeitssuche in kombinatorischen Bibliotheken anwendbar ist. Darüber hinaus wird der CSFP aber auch als Werkzeug in anderen Verfahren dienen, die in Kapitel 4, 5 und 6 beschrieben sind. Für diese Verfahren wird eine weitere Eigenschaft des CSFP von zentraler Bedeutung sein. Der CSFP Fingerabdruck eines beliebigen Fragments aus einem topologischen Fragmentraum bildet eine Teilmenge des CSFP Fingerabdrucks jedes Produkts, das mithilfe des Fragments gebildet werden kann. Diese Charakteristik wird im Folgenden als *Fingerabdruck-Teilmengenrelation* bezeichnet. In Kapitel 2.3.2 wurde die Substruktur-Teilmengenrelation eingeführt, derzufolge alle chemischen Substrukturen eines Fragments auch in jedem mit seiner Hilfe gebildeten finalen Molekül enthalten sind. Der CSFP und alle seine Varianten außer der pCSFP Variante betrachten alle Subgraphen, deren Anzahl enthaltener Schweratome innerhalb eines Intervals liegt. Deshalb folgt die Fingerabdruck-Teilmengenrelation für diese molekularen Fingerabdrücke direkt aus der Substruktur-Teilmengenrelation. Der RDKit Fingerabdruck betrachtet alle, auch nicht-induzierten, Subgraphen und erfüllt damit die Fingerabdruck-Teilmengenrelation auch. Der TopologicalTorsion Fingerabdruck und die pCSFP Variante betrachteten nur Subgraphen mit Pfadstruktur. Diese Subgraphen verändern sich ebenfalls nicht bei der Produktbildung und somit gilt für sie die Fingerabdruck-Teilmengenrelation ebenfalls. Der Atom Pairs Fingerabdruck betrachtet die Distanzen zwischen allen Atompaaren. Da alle Ringschlüsse in der Fragmentrepräsentation bereits mithilfe von Linkern und Ringplatzhaltern gebildet werden, ändern sich die Distanzen der Atompaare nicht mehr zwischen Fragmenten und Produkten. Deshalb erfüllt die Atom Pairs Methode die Fingerabdruck-Teilmengenrelation. Da die Identifikatoren eines topologischen molekularen Fingerabdrucks chemische Substrukturen repräsentieren, könnte man annehmen alle topologischen molekularen Fingerabdrücke erfüllen die Fingerabdruck-Teilmengenrelation. Dies ist allerdings nicht der Fall. In Abbildung 3.3 ist ausgeführt, warum die ECFP Methode die Fingerabdruck-Teilmengenrelation nicht erfüllt. Der zirkuläre Subgraph mit

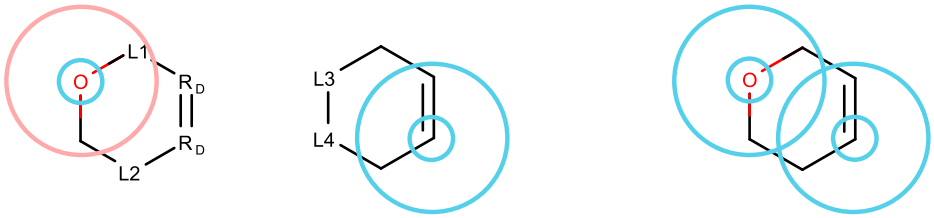


Abbildung 3.3.: Die Fragmentrepräsentationen von Formaldehyd (links), Vinylethylen (mittig) und ihrem Produkt Dihydropyran (rechts) in einer Diels-Alder Reaktion. [91] Einige zirkuläre Subgraphen der ECFP2 Fingerabdrücke sind markiert. Subgraphen deren Identifikatoren im ECFP2 Fingerabdruck des finalen Produkts vorkommen sind türkis, andernfalls sind sie rosa gefärbt. Aus [D1] entnommen und angepasst.

Radius eins und dem Sauerstoffatom als Zentrum enthält in der Fragmentrepräsentation von Formaldehyd nur das Sauerstoffatom selbst und das benachbarte Kohlenstoffatom. Da bei der Kombination der Fragmente ein weiterer Kohlenstoffnachbar hinzukommt, enthält der zirkuläre Subgraph mit Radius eins und dem Sauerstoffatom als Zentrum drei Schweratome in Dihydropyran. Somit ist der Identifikator des Subgraphen im Fragment nicht im ECFP2 Fingerabdruck von Dihydropyran enthalten und die ECFP Methode erfüllt die Fingerabdruck-Teilmengenrelation nicht.

3.2.6. Abgrenzung zu existierenden Verfahren

Die Methode zur Erzeugung des CSFP Fingerabdrucks betrachtet als erstes Verfahren alle induzierten Subgraphen eines Moleküls. Induzierte Subgraphen, die eine Menge von Atomen und alle Bindungen zwischen ihnen repräsentieren, entsprechen in ihrer Darstellung der Intuition einer chemischen Substruktur. Damit ist der CSFP Fingerabdruck ein guter Molekulardeskriptor, um die chemischen Substrukturen eines Moleküls zu beschreiben. Ein generierter Identifikator kann auch einzeln verwendet werden, um eine chemische Substruktur eindeutig zu repräsentieren. Das Verfahren zur Generierung des RDKit Fingerabdrucks betrachtet alle, auch nicht induzierte, Subgraphen und ähnelt damit dem Vorgehen des CSFP wenn auch in Ringsystemen mehr Subgraphen enumeriert werden. Allerdings betrachtet diese Methode weniger Atomeigenschaften, als der feingranulare klassische CSFP und die fCSFP Variante. Zusätzlich werden die Identifikatoren der einzelnen Atome lediglich ihrer Größe nach sortiert und zum Identifikator eines Subgraphen kombiniert. Dadurch entstehen weniger eindeutige Identifikatoren, als mit der adaptieren CANON-Prozedur der CSFP Methode. Der CSFP erzielte höhere Werte im in [D1]

durchgeführten Benchmark [112] und seine Ergebnisse unterschieden sich signifikant von denen des RDKit Fingerabdrucks. Dies könnte auf die zusätzlich betrachteten Atomeigenschaften zurückzuführen sein. Alle anderen topologischen molekularen Fingerabdrücke betrachten nur eine Teilmenge der induzierten Subgraphen eines Molekulargraphen. Dadurch enthalten besonders die Fingerabdrücke von Fragmenten aus topologischen Fragmenträumen mehr Identifikatoren, wenn sie mit der fCSFP Variante berechnet werden. Die prominenteste Methode für topologische molekulare Fingerabdrücke ist die des ECFP. Wir konnten in [D1] und Kapitel 3.2.4 zeigen, dass die CSFP Methode in einem Benchmark zur Ähnlichkeitssuche in enumerierten Bibliotheken vergleichbare und teilweise bessere Ergebnisse erzielt als der ECFP und andere molekulare Fingerabdrücke. Aufgrund seiner hohen Parametrisierbarkeit, Betrachtung aller Subgraphen eines Molekulargraphen und der erfüllten Fingerabdruck-Teilmengenrelation, wird die CSFP Methode der Kernbaustein aller weiterführenden Verfahren dieser Dissertation sein.

3.2.7. Ausblick

Die CSFP Methode wurde in die Codebasis NAOMI [54] implementiert und ist bewusst so aufgebaut, dass weitere Varianten leicht hinzugefügt werden können. Beispielsweise könnte, ähnlich wie bei der pCSFP Variante, die Menge der enumerierten Subgraphen eingeschränkt werden. Es wäre zB. denkbar nur Subgraphen an Pharmakophoren zu enumerieren, um die topologischen und pharmakologischen Eigenschaften eines Moleküls zu betrachten.

In der Codebasis NAOMI existieren effiziente Verfahren für die Suche nach größten gemeinsamen chemischen Substrukturen (*maximum-common-substructure problem*(MCS)) [36] in enumerierten, [118] sowie in kombinatorischen Bibliohteken. [119] Um die Suche in großen Bibliotheken noch effizienter zu gestalten, könnten CSFP Fingerabdrücke mit den passenden betrachteten Atom- und Bindungseigenschaften für alle Moleküle oder Fragmente einer Bibliothek bzw. Fragmentraum präprozessiert werden. Damit könnten Moleküle oder Fragmente schnell ausgeschlossen werden, die mit einem Anfragemolekül keine größere gemeinsame chemische Substruktur teilen.

Die Analyse von Molekülpaaren, die sich nur in einer chemischen Substruktur unterscheiden (*matched molecular pairs*(MMP))[120] ist ein weiteres denkbares Anwendungsgebiet der CSFP Methode. Die sich änderende chemische Substruktur und ihre Umgebung könnte mithilfe eines CSFP Fingerabdrucks feingranular beschrieben und ein Datensatz effizient durchsucht werden. Diese Anwendung ist auch für räumliche MMPs [121]

möglicherweise sogar in Protein-Ligand-Komplexen möglich. Hierfür könnte auch das in Kapitel 3.1 beschriebene Verfahren SPLIF [109] angepasst werden. Die Umgebung der Interaktionspunkte in einem Protein-Ligand-Komplex könnte mithilfe von CSFP Fingerabdrücken beschrieben werden.

Eine Anwendungsgebiet molekulare Fingerabdrücke liegt im maschinellen Lernen. [103] Mit der Betrachtung aller Subgraphen, einer hohen Parametrisierbarkeit und verschiedener Varianten könnten CSFP Fingerabdrücke als Eingabe für Modelle des maschinellen Lernens wie neuronale Netze sein. In diesem Kontext wäre die Untersuchung von Kollisionen der Identifikatoren eines CSFP Fingerabdrucks bei der Faltung zu einer Bitfolge interessant.

4. Topologische Ähnlichkeitssuche in kombinatorischen Bibliotheken

In diesem Kapitel wird das Konzept der Ähnlichkeitssuche in virtuellen chemischen Bibliotheken behandelt. Dabei wird in einer Bibliothek nach Molekülen gesucht, die zu einem gegebenen Anfragemolekül ähnlich sind. Dieser Ansatz basiert auf dem Ähnlichkeitsprinzip von Johnson und Maggiora, dass ähnliche Moleküle auch ähnliche, zB. physikochemische oder pharmakologische, Eigenschaften aufweisen. [122] Der Begriff von Ähnlichkeit zwischen zwei Molekülen wird entscheidend durch den verwendeten Molekulardeskriptor beeinflusst. Zwei Molekülen wird ein Ähnlichkeitswert, meist zwischen null und eins, zugewiesen. Die Ähnlichkeit der zwei Moleküle wird in Abhängigkeit der Höhe des zugewiesenen Werts angenommen, wobei der Grenzwert ab dem zwei Molekül als ähnlich betrachtet werden können stark von der verwendeten Methode und den zwei Molekülen abhängt. [123] Wir geben zunächst einen Überblick über existierende Verfahren zur Ähnlichkeitssuche in enumerierten und kombinatorischen Bibliotheken und stellen dann unsere in [D2] eingeführte Methode vor.

4.1. Bestehende Verfahren zur Ähnlichkeitssuche in enumerierten Bibliotheken

Für die Ähnlichkeitssuche in enumerierten Bibliotheken wurden zahlreiche Methoden entwickelt, die oft Molekulardeskriptoren verwenden und sich durch diese voneinander abgrenzen. [38] Für alle Moleküle einer enumerierten virtuellen chemischen Bibliothek wird einzeln der Ähnlichkeitswert zu einem Anfragemolekül bestimmt. Anschließend werden diejenigen Moleküle mit den höchsten erzielten Werten oder mit Werten über einer vom Nutzer bestimmten unteren Schranke ausgegeben.

In ihrer kompaktesten Form kann die Ähnlichkeit zwischen zwei Molekülen über die Betrachtung einer einzelnen physikochemischen Eigenschaft [41] oder eines Terms

[94] definiert werden. Dixon und Merz entwickelten einen Molekulardeskriptor [124] für dessen Erzeugung eine räumliche Konformation eines Moleküls oder alternativ dessen Molekulargraph in einen ein-dimensionalen Raum abgebildet wird. In diesem Raum wird jedes Atom auf der Basis von Element, Hybridisierung und Anzahl seiner Nachbarn repräsentiert. Für die Ähnlichkeitssuche werden die generierten Molekulardeskriptoren zweier Moleküle aliniert und ihr Ähnlichkeitswert berechnet. Dieses Verfahren erzielte trotz seiner ein-dimensionalen Repräsentation teilweise bessere Ergebnisse bei der Vorhersage von Bioaktivität, als andere Verfahren mit komplexeren Molekulardeskriptoren. [124]

Die Ähnlichkeit zwischen zwei Molekülen kann auch über die relative Größe ihrer größten gemeinsamen chemischen Substruktur (*maximum common substructure*(MCS)) definiert werden. Hierbei wird für ein Molekül nicht erst ein Molekulardeskriptor generiert, sondern ein algorithmisches Verfahren bestimmt direkt mithilfe zweier Molekulargraphen die größte gemeinsame chemische Substruktur. Diese Beschreibung von molekularer Ähnlichkeit existiert bereits seit den 1970er Jahren [125] und es wurden zahlreiche Ansätze zur exakten und approximativen Bestimmungen chemischer Substrukturen mithilfe von induzierten und nicht-induzierten Subgraphen etwickelt. [36], [118] Mit diesem Verfahren werden zwei Moleküle als ähnlich betrachtet, wenn ein großer Teil von ihnen strukturell identisch ist.

Methoden für molekulare Fingerabdrücke zählen zu den weitverbreitetsten Verfahren in der Ähnlichkeitssuche. Der Ähnlichkeitswert zweier Molekül wird meistens auf Basis des Tanimotokoeffizienten [113] ihrer molekularen Fingerabdrücke bestimmt, wobei auch andere Koeffizienten angewendet werden. [126], [127] Hierbei erzielen Methoden die chemische Substrukturen und topologische Eigenschaften betrachten teilweise bessere Ergebnisse als andere Verfahren, die räumliche Konformationen berücksichtigen. [128] In Kapitel 3.1 ist ein Überblick über existierende Verfahren zur Erzeugung molekularer Fingerabdrücke gegeben. Im Gegensatz zur Suche nach einer größten gemeinsamen chemischen Substruktur, legen topologische molekulare Fingerabdrücke für die Ähnlichkeit zweier Moleküle den Fokus auf viele kleinere gemeinsame chemische Substrukturen. Diese können sich theoretisch aber in ihrer Anordnung zwischen den zwei Molekülen unterscheiden.

Eine andere Methode, die die Graphstruktur eines Molekulargraphen betrachtet ist das Verfahren zur Erzeugung von Features Trees [90]. Der generierte Molekulardeskriptor betrachtet einen Graphen mit Baumstruktur, indem Ringe und teilweise Ringsysteme eines Molekulargraphen zu einem Knoten zusammengefasst werden. Jeder Knoten repräsentiert

somit eine chemische Substruktur. Für jeden Knoten werden mithilfe seines FLEXX-Interaktionsprofils [129] pharmakophorische Eigenschaften betrachtet. Zusätzlich werden sterische Eigenschaften durch das approximierte Volumen und Anzahl Atome der repräsentierten chemischen Substruktur eingefangen. Die Ähnlichkeit zwischen zwei Molekülen wird durch eine bestmögliche Paarung von Teilbaumstrukturen ihrer Feature Trees auf Basis der betrachteten pharmakophorische und sterischen Eigenschaften bestimmt. Durch dieses Vorgehen wird die Ähnlichkeit zweier Moleküle durch ihre möglichen Interaktionspunkte mit einem Protein, sterische Informationen sowie eine grobgranulare Betrachtung ihres Molekulargraphen beschrieben. Damit unterscheidet sich die Feature Trees Methode stark von der Ähnlichkeitsbeschreibung topologischer molekularer Fingerabdrücke oder durch eine größte gemeinsame chemische Substruktur.

Weitere existierende Definitionen der Ähnlichkeit zweier Moleküle basiert auf ihrer räumlichen Form. Um die Ähnlichkeit zweier Moleküle zu bestimmen, wird ihre räumliche Konfiguration so gut wie möglich überlagert und der Ähnlichkeitswert abgeleitet. Damit unterscheiden sich diese Ansätze fundamental von den vorangegangen, die auf der topologischen Struktur von Molekulargraphen operieren. Das Programm ROCS [130], [131] ist das prominenste Verfahren zur räumlichen Überlagerung von Molekülen. Es versucht eine optimale Paarung zwischen Teilen der Atome zweier Moleküle zu finden und die Moleküle mithilfe dieser Paarung zu überlagern. Der Matrix-basierte Molekulardeskriptor RVM [132] generiert für ein Molekül das ausgefüllte Volumen mithilfe von Geradenabschnitten, die die Atome des Moleküls schneiden. Zwei RVMs werden dann auf einem Gitter überlagert. Durch die Repräsentation mithilfe einer Matrix wird ein aufwändiges Verfahren zur Paarung vermieden. Andere Verfahren verwenden wie ROCS für die Überlagerung ebenfalls eine Paarung, wobei allerdings keine Atome sondern Fragmente [133] bzw. Pharmakophore [134] zweier Moleküle gepaart und überlagert werden.

Anstatt die gesamte räumliche Komformation eines Moleküls zu betrachten, können für die Definition von Ähnlichkeit auch nur die räumliche Position der Pharmakophore zweier Moleküle betrachtet werden. Hierbei werden einzelne Atome oder funktionelle Gruppen nach ihrer Eignung als Protonenakzeptor oder Protonendonator einer Wasserstoffbrückenbindung charakterisiert, ob sie positiv oder negativ ionisierbar und ob sie aromatisch oder lipophil sind. Anders als bei Pharmakophor-basierten molekularen Fingerabdrücke wie sie in Kapitel 3.1 beschrieben wurden, wird hier allerdings die räumliche Anordnung der Pharmakophor-Punkte betrachtet. Das Programm LigandScout [135] und die Software-Plattform MOE [136] verwenden einen Paarungsansatz, um die Pharmakophor-Punkte zweiere Moleküle bestmöglich zu überlagern und so ihre Ähnlichkeit zu bestimmen. Das

Programm Pharmer [137] indexiert eine Menge von Molekülen mithilfe von Triangulierungen zwischen Tripeln von Pharmakophor-Punkten. Dadurch wird die Ähnlichkeitssuche in größeren enumerierten Bibliotheken effizienter gestaltet.

Zuletzt wird das Verfahren SmallWorld [138] beschreiben. Es verwendet einen Indexierungsansatz, um die Ähnlichkeitssuche in großen enumerierten Bibliotheken zu beschleunigen. Hierfür werden zunächst alle Subgraphen jedes Moleküls enumeriert und kanonisch repräsentiert. Nun wird ein Editierungsgraph aufgebaut, in dem jedes Molekül und jeder eindeutige Subgraph einen Knoten bilden. Zwei Knoten werden durch eine Kante verbunden, wenn die zwei Subgraphen bzw. Moleküle sich nur durch Löschen bzw. Hinzufügen oder Verändern eines Atoms oder einer Bindung unterscheiden. Der Ähnlichkeitswert zwischen zwei Molekülen bzw. Subgraphen entspricht der Distanz zwischen ihren Knoten im Editierungsgraphen. Dieses Ähnlichkeitsmaß wird auch als Grapheditierungsdistanz bezeichnet. Für ein Anfragemolekül werden nun Subgraphen mit absteigender Anzahl Atome und Bindungen enumeriert, bis einer von ihnen im Editierungsgraph der enumerierten virtuellen chemischen Bibliothek gefunden wird. Nun können die ähnlichsten Moleküle mit einer Breitensuche [139] vom Knoten dieses Subgraphen aus im Editierungsgraphen effizient gefunden werden. Mit diesem Ansatz müssen nicht alle Moleküle der Bibliothek auf ihre Ähnlichkeit untersucht werden, allerdings müssen alle Moleküle einer enumerierten Bibliothek und sogar ihre Subgraphen im Editierungsgraphen abgespeichert werden. Damit eignet sich das Verfahren nicht für kombinatorische Bibliotheken, deren Produktmenge zu groß für die Enumeration ist.

4.2. Herausforderungen der Ähnlichkeitssuche in kombinatorischen Bibliotheken und existierende Verfahren

So unterschiedlich die in Kapitel 4.1 beschriebenen Verfahren auch sind, alle bis auf die Methode SmallWorld eint die sequenzielle, unabhängige Berechnung der Ähnlichkeit aller Moleküle einer enumerierten Bibliothek zu einem Anfragemolekül. Um diese Verfahren auf kombinatorische Bibliotheken anzuwenden, müssten alle implizit beschriebenen Produkte enumeriert werden, dies gilt auch für das Verfahren SmallWorld. Speziell für große Bibliotheken mit Milliarden oder Billionen von Produkten würde die Idee der kompakten, kombinatorischen Repräsentation so zunichte gemacht und eine effiziente Ähnlichkeitssuche wäre nicht möglich ohne einen großen Ressourcenaufwand. [88]

Die Methode TOPAS [76] umgeht dieses Problem, indem sie einen evolutionären Algorithmus verwendet, um nur eine zufällige Auswahl von implizit beschriebenen Molekülen zu enumerieren. Die Ähnlichkeit dieser enumerierten Moleküle zum Anfragemolekül werden mithilfe der molekularen Fingerabdrucksmethode CATS [78] bestimmt. In Abhängigkeit der erzielten Ähnlichkeitswerte werden nun im nächsten Schritt des evolutionären Algorithmus neue Moleküle ausgewählt und dieses Vorgehen solange wiederholt, bis ein Konvergenzkriterium erfüllt ist oder die Optimierung beendet wird. Durch diesen Ansatz wird die komplette Enumeration aller Produkte einer kombinatorischen Bibliothek verhindert. Allerdings kann die Verwendung einer Heuristik wie eines evolutionären Algorithmus dazu führen, dass die globalen Optima, also in diesem Fall die Produkte mit dem höchsten Ähnlichkeitswert zur Anfrage, nicht gefunden werden.

Wenn eine Methode zur exakten Ähnlichkeitssuche den kombinatorischen Charakter dieser Bibliotheken ausnutzen möchte, muss sie auf den Daten operieren die explizit enkodiert sind. Dies sind die Repräsentationen chemischer Bausteine und Reaktionen. Trotzdem soll die Methode in der Lage sein die ähnlichsten implizit beschriebenen Produkte zu identifizieren. Dies ist das bereits genannte algorithmische kombinatorische Paradigma. Chemische Bausteine können unähnlich zu einem Anfragemolekül sein, beispielsweise weil sie deutlich weniger Atome enthalten. Ein Produkt, dass aus einer Kombination unähnlicher chemischer Bausteine entsteht, kann dennoch ähnlich zu einem Anfragemolekül sein. Um dieses Problem zu lösen, greifen alle existierenden [53], [75], [119] und in dieser Dissertation beschriebenen, [D2] nicht-heuristischen, algorithmischen Verfahren zur Ähnlichkeitssuche in kombinatorischen Bibliotheken auf eine Idee zurück, die im Folgenden als *Prinzip der Anfragepartitionierung* bezeichnet wird. Hierbei wird der Molekulargraph eines Anfragemoleküls in Subgraphen unterteilt. Es werden die ähnlichsten Paare von Subgraphen und Repräsentationen chemischer Bausteine identifiziert. Anschließend werden die Repräsentationen kombiniert und so die ähnlichsten finalen Moleküle generiert. Durch diesen Ansatz operiert eine Methode zunächst nur auf den chemische Bausteinen und Reaktionen einer kombinatorischen Bibliothek. Erst bei der Kombination chemischer Bausteine werden finale Moleküle betrachtet wodurch nur ein Bruchteil aller implizit beschriebenen Produkte enumeriert werden muss. Alle anderen Moleküle wurden bereits ausgeschlossen, da sie aus mindestens einem chemischen Baustein entstehen, der zu keinem Teil des Anfragemoleküls ähnlich ist.

Das algorithmische Verfahren FTrees-FS [53] verwendet die Methode Feature Trees [90] für die Ähnlichkeitssuche in klassischen Fragmenträumen, wie sie in Kapitel 2.2 beschrieben wurde. Hierbei wird die Eigenschaft klassischer Fragmenträume ausgenutzt

keine Ringschlüsse zwischen Fragmenten zu erlauben. Die azyklischen Kanten des Molekulargraphen eines Anfragemoleküls werden zunächst topologisch sortiert [139] und dann dieser Sortierung folgend prozessiert. Für eine Kante wird nun jeweils das ähnlichste einzelne Fragment oder Fragmentkombination zu beiden Subgraphen bestimmt, die durch Entfernen der Kanten getrennt werden würden. Für einen Subgraphen der Anfrage wird zunächst der Feature Trees Molekulardeskriptor erzeugt und die optimale Paarung mit allen Feature Trees Deskriptoren der Fragmente des Fragmentraums gebildet. Hierbei muss der Feature Tree Knoten eines Atomnachbarn eines Linkers mit dem Knoten des Atoms des Subgraphen an der Schnittkante gepaart werden. Falls nicht alle Knoten des Feature Tree des Subgraphen gepaart sind, werden die gepaarten Knoten mit der größten Distanz zur Schnittkante bestimmt. Die Kanten zu ungepaarten Knoten des Feature Tree und damit zu Knoten oder Ringsystemen des Subgraphen wurden bereits früher in der topologischen Sortierung prozessiert. Deshalb kann mithilfe eines dynamischen Programmieransatzes der optimale Ähnlichkeitswert für den Rest des Subgraphen ausgehend von dieser Kante abgerufen und verwendet werden, um ihn mit der betrachteten Paarung zu kombinieren. Dadurch wird der höchste Ähnlichkeitswert für die aktuell betrachtete Schnittkante und die dazugehörige Fragmentkombination bestimmt. FTrees-FS bezieht seine Beschreibung von Ähnlichkeit aus den Feature Trees Molekulardeskriptoren. Deshalb betrachtet das Verfahren pharmakophorische und sterische Eigenschaften.

Das Verfahren SpaceMACS [119] operiert ebenfalls auf klassischen Fragmenträumen und bestimmt mithilfe eines kombinatorischen Ansatzes die implizit beschriebenen Moleküle mit der größten gemeinsamen chemischen Substruktur zu einem Anfragemolekül. Hierfür verwendet SpaceMACS dieselbe topologische Sortierung der azyklischen Kanten des Molekulargraphen eines Anfragemoleküls wie FTrees-FS und ebenfalls einen dynamischen Programmieransatz. Die generierten Paarungen werden allerdings nicht auf Feature Trees, sondern mithilfe des algorithmischen Verfahrens RIMACS [118] direkt auf den Subgraphen des Molekulargraphen und den Fragmenten gebildet. Hierbei werden nur Atome mit gleichem Element, sowie Bindungen vom gleichen Typ und gleicher Ringzugehörigkeit miteinander gepaart. Zusätzlich ist es möglich anstatt eines Anfragemoleküls einen SMARTS Ausdruck [37] zu spezifizieren. Für den Ausdruck wird intern eine Graphstruktur gebildet und der Paarungsansatz mithilfe dieser Struktur anstatt des Molekulargraphen eines Anfragemoleküls durchgeführt.

Mit der kürzlich entwickelten Pythonbibliothek SynthI [75] ist es neben der in Kapitel 2.2 beschriebenen Funktionalitäten auch möglich eine enumerierte Bibliothek zu generieren, indem chemische Bausteine kombiniert werden, die zu Subgraphen eines Anfragemoleküls

ähnlich sind. Auch in diesem Verfahren wird das Prinzip der Anfragepartitionierung angewendet. Allerdings werden nicht alle Bindungen des Molekulargraphen eines Anfragemoleküls betrachtet, sondern wie in einem retrosynthetischen Ansatz eine Menge von Reaktionen für die Partitionierung und Generierung von Subgraphen verwendet. Für jeden so generierten Subgraphen werden nun alle dazu ähnlichen Repräsentationen chemischer Bausteine, hier Synthons genannt, bestimmt. Für die Definition von Ähnlichkeit gibt es zwei Auswahlmöglichkeiten für den Nutzer. Entweder werden ein Synthon und ein Subgraph des Molekulargraphen des Anfragemoleküls als ähnlich betrachtet, wenn der Tanimotokoeffizient [113] ihrer ECFP Fingerabdrücke [98] über einem vom Nutzer definierten Grenzwert liegt. Oder ihre Ähnlichkeit wird über die Positional Analog Scanning (PAS) Strategie [140] definiert. Hierbei müssen zwei ähnliche Subgraphen in einer Beziehung als Teil- bzw. Supergraph zueinander stehen und sich zusätzlich nur im Vorkommnis einer chemischen Substruktur unterscheiden. Die so gefundenen ähnliche Synthons werden schlussendlich rekombiniert, um eine enumerierte Bibliothek zu erzeugen. Damit weicht das Verfahren von einer Ähnlichkeitssuche ab, da den finalen Molekülen kein Ähnlichkeitswert zugewiesen wird. Dennoch filtert die Methode zumindest auf der Ebene der Synthons nach Ähnlichkeit und auf der zurückgegeben enumerierten virtuellen chemischen Bibliothek kann nachträglich eines der in Kapitel 4.1 beschriebenen Verfahren angewandt werden.

4.3. Beschreibung der SpaceLight Methode

In diesem Kapitel wird das algorithmische Verfahren SpaceLight aus der Publikation [D2] beschrieben. SpaceLight ermöglicht erstmals eine Ähnlichkeitssuche in kombinatorischen virtuellen chemischen Bibliotheken mit einem topologischen Begriff von molekularer Ähnlichkeit mithilfe topologischer molekularer Fingerabdrücke. Das Verfahren verwendet die in Kapitel 2 eingeführten topologischen Fragmenträume, sowie die Methoden ECFP [98] und CSFP [D1] für molekulare Fingerabdrücke.

4.3.1. Partitionierung

SpaceLight verwendet wie die in Kapitel 4.2 beschriebenen Verfahren das Prinzip der Anfragepartitionierung, um nur auf den Fragmenten eines topologischen Fragmentraumes zu operieren. Im Gegensatz zu existierenden Verfahren, betrachtet SpaceLight allerdings auch Partitionen von Ringsystemen des Anfragemoleküls, wenn Reaktionen mit

Ringschlüssen zur Erzeugung des topologischen Fragmentraumes verwendet wurden. Bei der Partitionierung wird darauf geachtet, dass die generierten Partitionen topologisch gleichwertig zu einem Topologiegraphen des topologlischen Fragmentraumes sind. Dazu wird die Anzahl und Größe der Subgraphen einer Partition betrachtet, sowie die zwischen ihnen verlaufenden Bindungen.

Subgraphen des Molekulargraphen des Anfragemoleküls und Fragmente eines topologischen Fragmentraumes erfüllen die *Subgraph-Fragment-Kompatibilität*, wenn sich ihre Anzahl enthaltener Schweratome um höchstens fünf unterscheidet. Zusätzlich muss die Anzahl Bindungen mit genau einem Atom aus dem Subgraphen der Anzahl an Bindungen zu Linkern im Fragment entsprechen. In diesem Fall gleichen sich der Subgraph und das Fragment in Größe und Konnektivität zu anderen Teilen des Anfragemoleküls bzw. der implizit beschriebenen Produkte. Alle weiteren Definitionen von Kompatibilität und des Topologiewerts, sowie des Partitionierungs- und Paarungsschritts von SpaceLight sind in Anhang B.3 gegeben. Zusammengefasst werden in diesem Abschnitt des algorithmischen Verfahrens Partitionen des Anfragemoleküls in Subgraphen generiert, die topologisch gleichwertig zu einer Kombination von Fragmenten aus dem topologischen Fragmentraum sind. Falls keine solche Partitionen gefunden werden, terminiert das Verfahren. In Abbildung 4.1 wird eine Partition des Moleküls CHEMBL1091518 mit topologisch gleichwertiger Paarung zum Topologiegraphen gezeigt. Die Subgraphen der Partition sind zu mindestens einem der Fragmente des gepaarten Knoten kompatibel. Die Einfachbindungen, sowie die zwei Bindungen in aromatischen Ringen zwischen den Subgraphen der Partition entsprechen genau den Bindungstypen der Kanten zwischen den gepaarten Knoten des Topologiegraphen.

4.3.2. Ähnlichkeitsberechnung

Im vorangegangenen Schritt wurden alle Paarungen von topologisch gleichwertigen Partition des Molekulargraphen eines Anfragemoleküls und Topologiegraphen des topologischen Fragmentraumes bestimmt. Dadurch kann sich im nächsten Schritt für die topologische Ähnlichkeitssuche auf die chemischen Substrukturen innerhalb der Fragmente des topologischen Fragmentraumes und Subgraphen der Partitionen beschränkt werden. Im Vergleichsschritt wird die Ähnlichkeit eines Subgraphen einer Partition zu allen kompatiblen Fragmenten, des mit ihm gepaarten Knoten bestimmt. Dafür kann der Nutzer zwischen der Methode ECFP [98] mit Radius null bis 5 sowie den CSFP Varianten fCSFP, iCSFP und tCSFP mit einer oberen Schranke von bis zu sechs Atomen für die Größe der

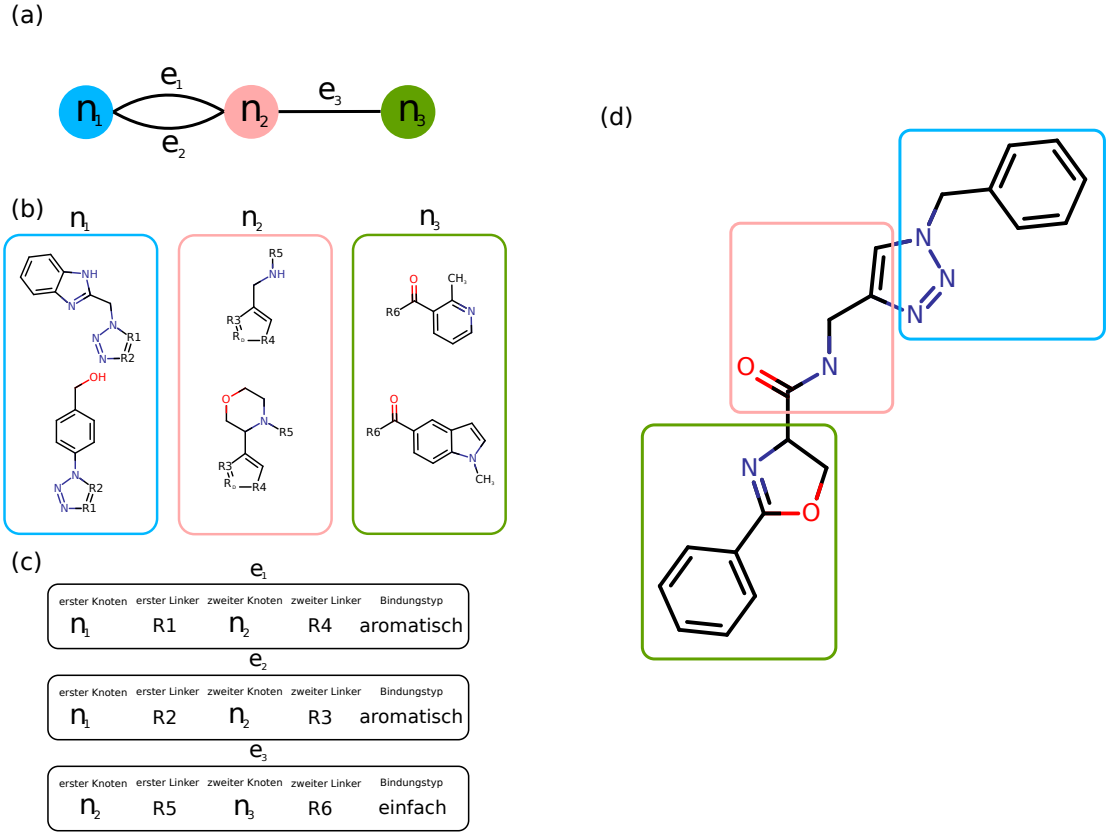


Abbildung 4.1.: Der Topologiegraph aus Abbildung 2.3 in (a)-(c) zusammen mit einer Partition des Moleküls CHEMBL1091518 der ChEMBL Datenbank [43] mit einer Paarung mit Topologiewert eins in (d). Die gefärbten Kästen bezeichnen die Subgraphen der Partition, wobei die Farbe dem gepaarten kompatiblen Knoten entspricht. Aus [D2] entnommen, angepasst und ins Deutsche übersetzt.

betrachteten chemischen Substrukturen wählen. Für alle Fragmente wurden die molekularen Fingerabdrücke bereits berechnet und in einer Datenbank hinterlegt, sodass sie nur noch ausgelesen werden müssen. Für die betrachteten Subgraphen des Molekulargraphen des Anfragemoleküls werden die molekularen Fingerabdrücke berechnet. Hierbei werden nur Atome des Subgraphen direkt für die Berechnung verwendet. Die Konnektivität zu Atomen außerhalb des Subgraphen wird aber dennoch implizit durch die betrachteten Atomeigenschaften wie Konnektivität in den berechneten molekularen Fingerabdrücken berücksichtigt. Damit spielen an den Subgraphen angrenzende Atome eine vergleichbare Rolle zu den Linkern der Fragmente des topologischen Fragmentraumes. Mithilfe des Tanimotokoeffizienten [113] werden nun die k ähnlichsten kompatiblen Fragmente zu dem

gegebenen Subgraphen bestimmt. Hierbei entspricht k dem Minimum aus der Anzahl vom Nutzer geforderten Ergebnisse und der Anzahl kompatibler Fragmente, die im Knoten enthalten sind.

Im Kombinationsschritt werden alle Kombinationen aus den k ähnlichsten Fragmenten für die betrachtete Paarung generiert, die genau ein Fragment pro Knoten des Topologiegraphen enthalten. Der Ähnlichkeitswert einer Fragmentkombination ergibt sich als die gewichtete Summe der Ähnlichkeitswerte der einzelnen Fragmente. Das Gewicht jedes Fragments beträgt $\frac{s}{n}$, wobei s der Anzahl Atome im zum Knoten gepaarten Subgraphen entspricht und n die Anzahl Schweratome im Anfragemolekül ist. Damit erhalten größere Subgraphen ein höheres Gewicht und haben deshalb einen größeren Einfluss auf den Ähnlichkeitswert der Fragmentkombination und der Partition. Dieses Verfahren wird für alle Partitionen und alle ihre topologisch gleichwertigen Paarungen vollzogen. Die gesammelten Fragmentkombinationen werden mit ihren Ähnlichkeitswerten in einer zentralen Datenstruktur abgespeichert. Falls eine Fragmentkombination für unterschiedliche Partitionen oder Paarungen einen Ähnlichkeitswert erhält, wird der höhere Wert priorisiert. Schlussendlich werden für die Fragmentkombinationen mit den höchsten Ähnlichkeitswerten die finalen Produkte gebildet und an den Nutzer zusammen mit ihren erzielten Werten zurückgegeben. Der Kombinationsschritt ist der erste Moment innerhalb des SpaceLight Verfahrens, in dem tatsächlich Produkte betrachtet werden. Da im Vergleichschritt nur eine Auswahl von ähnlichen Fragmenten getroffen wurde, wird hier die vollständige Enumeration aller Produkte vermieden. Die asymptotische Laufzeit des Verfahrens skaliert somit mit der Anzahl der Fragmente eines topologischen Fragmentraums und nicht mit der Anzahl seiner implizit beschriebenen Produkte.

4.3.3. Validierung und Evaluation

In [D2] evaluieren wir drei Eigenschaften der SpaceLight Methode. Zur Evaluation verwenden wir den Enamine REAL Space [81] sowie den KnowledgeSpace [84], wie sie in Kapitel 2.2 beschrieben wurden, in ihrer Repräsentation als topologische Fragmenträume.

Zunächst validieren wir die Fähigkeit von SpaceLight Moleküle zu identifizieren, die in einem topologischen Fragmentraum enthalten sind. Dafür wählen wir aus jedem Topologiegraphen der zwei topologischen Fragmenträume zehn zufällige Produkte aus. Daraus ergeben sich 2320 bzw. 1170 Produkte, die wir als Anfragemoleküle zusammen mit dem jeweiligen topologischen Fragmentraum als Eingabe für SpaceLight verwendet wird. Jedes Anfragemolekül wurde in seinem topologischen Fragmentraum als das ähnlichste Molekül

mit einem Ähnlichkeitswert von eins wiedergefunden. Alle anderen Molekülen des topologischen Fragmentraumes wiesen einen Ähnlichkeitswert echt kleiner eins auf. Methoden für topologische molekulare Fingerabdrücke wie das ECFP- oder CSFP-Verfahren verwenden für die Ähnlichkeitsbeschreibung chemische Substrukturen. Damit ist es möglich das zwei unterschiedliche Moleküle, beispielsweise zwei lange Kohlenstoffketten mit einer unterschiedlich Anzahl Kohlenstoffatome, in SpaceLight und der Ähnlichkeitssuche mithilfe topologischer molekularer Fingerabdrücke im Allgemeinen einen Ähnlichkeitswert von eins zugewiesen bekommen. Bei der Verwendung realer chemischer Daten, wie sie zur Erzeugung des Enamine REAL Space und KnowledgeSpace verwendet wurden, tritt dieser Fall allerdings nicht ein. Damit ist SpaceLight in der Lage das Vorkommen oder die Abwesenheit einzelner Moleküle in großen, nicht-enumerierbaren kombinatorischen Bibliotheken zu überprüfen.

Als nächstes evaluieren wir SpaceLights Beschreibung molekularer Ähnlichkeit. Hierfür existieren im Kontext enumerierter Bibliotheken Benchmarks, wie derjenige [112] den wir in Kapitel 3.2.4 zur Evaluation der CSFP Methode verwendet haben. Wir würden gerne vergleichbar vorgehen, allerdings existieren momentan im Bereich kombinatorischer Bibliotheken keine solchen Benchmarks. Stattdessen untersuchen wir den Grad der Übereinstimmung von SpaceLight mit einer Ähnlichkeitssuche mithilfe der ECFP und CSFP Methode. Da für diese Verfahren die Anwendbarkeit in einem Benchmark bereits gezeigt wurde, folgern wir ein ähnliches Ergebnis für die SpaceLight Methode bei einer ausreichenden Übereinstimmung. Hierbei muss aber zunächst ein Problem überwunden werden. Der Enamine REAL Space mit seinen 20 Milliarden und der KnowledgeSpace mit seinen 10^{14} implizit beschriebenen Molekülen sind zu groß, um komplett enumeriert zu werden. Deshalb wählen wir pro topologischem Fragmentraum drei zufällige Topologiegraphen aus, die höchstens 100.000 implizit beschriebene Moleküle enthalten. Diese Topologiegraphen definieren jeweils einen topologischen Teilraum, der klein genug ist, um vollständig enumeriert zu werden. Eine vergleichbare Strategie der Teilraumbildung wird auch für die in Kapitel 5 und 6 beschriebenen Verfahren angewendet. Für die Evaluation verwenden wir sowohl Moleküle aus den Teilräumen selbst, als auch externe Moleküle aus der ZINC Datenbank. [44] Das genaue Vorgehen ist in [D2] beschrieben. Die Resultate zeigen, dass die Ergebnisse einer Ähnlichkeitssuche mit SpaceLight stark korreliert sind zu den Ergebnissen einer Ähnlichkeitssuche mithilfe der Methoden ECFP und der verwendeten Varianten des CSFP. Diese Korrelation ist allerdings abhängig von der Parametrisierung der Methoden. Deshalb empfehlen wir für die Ähnlichkeitssuche die parametrisierten Methoden ECFP4, ECFP6, fCSFP2.4 und fCSFP2.5. Obwohl Space-

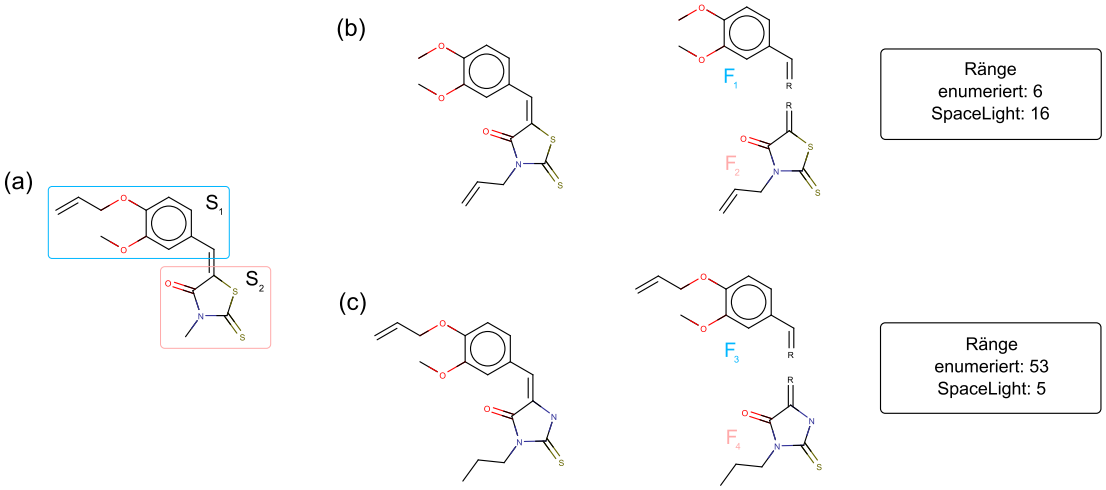


Abbildung 4.2.: (a) Das Molekül ZINC2932278 der ZINC Datenbank [44] mit seiner einzigen topologisch gleichwertigen Partition zum verwendeten Topologiegraphen. In (b) und (c) ist jeweils ein Produkt des Topologiegraphen mit der Fragmentkombination aus der er ensteht gezeigt. Die Farbe der Fragmentnamen stimmt mit der Farbe des Rahmen des gepaarten Subgraphen der Partition aus (a) überein. Für beide Moleküle ist der Rang in einer klassischen Ähnlichkeitsuche mit der ECFP4 Methode in der enumerierten Bibliothek nach 'enumeriert' gegeben. Der Rang des Moleküls in der SpaceLight Suche im topologischen Fragmentraum mit der ECFP4 im Vergleichsschritt ist nach 'SpaceLight' angegeben. Aus [D2] entnommen, angepasst und ins Deutsche übersetzt.

Lights Beschreibung molekularer Ähnlichkeit zu einem großen Ausmaß der von klassischen topologischen molekularer Fingerabdrücken entspricht, weicht sie methodisch durch ihren kombinatorischen Ansatz ab. In Abbildung 4.2 ist dies anhand eines Beispiels verdeutlicht. Das Anfragemolekül ZINC2932278 unterscheidet sich von dem in (b) gezeigten Molekül, um die Position der Alkengruppe. Da ein molekularer Fingerabdruck nur die Existenz einer chemischen Substruktur einfängt, wird der Unterschied in der Position dieser Gruppe in den ECFP4 Fingerabdrücken der zwei Moleküle nicht so stark berücksichtigt, wie in der Betrachtung durch SpaceLight. Da SpaceLight auf den einzelnen Fragmenten des Topologiegraphen operiert, wird der Fingerabdruck des Subgraphen S_1 mit dem des Fragments F_1 verglichen. Im Gegensatz zu F_1, enthält S_1 allerdings die Alkengruppe. Für den Subgraphen S_2 und das Fragment F_2 gilt die umgekehrte Aussage. Durch diese Betrachtung von Subgraphen und Fragmenten spielt die Positionsänderung der Alkengruppe eine größere Rolle für SpaceLight und die Moleküle werden als weniger ähnlich betrachtet. Das Molekül in (c) unterscheidet sich von ZINC2932278 durch die

Verlängerung einer Kohlenstoffkette und den Austausch eines Schwefelatoms mit einem Stickstoffatom in einem der Ringe. Gerade der Austausch des Atoms hat einen großen Einfluss auf den klassischen ECFP4 Fingerabdruck, da es sich um ein zentrales Atom innerhalb des Molekulargraphen handelt. In Folge des Austauschs dieses Atoms verändert sich der Identifikator vieler zirkulärer Subgraphen und dadurch erhält das Moleküle einen niedrigeren Rang von 53 bei der Ähnlichkeitssuche in der enumerierten Liste aller Moleküle des Teilraums. Da SpaceLight die Subgraphen der Partition von ZINC2932278 mit den Fragmenten F_3 und F_4 unabhängig voneinander vergleicht, beeinflusst der Austausch der Atome und die Verlängerung der Kohlenstoffketten nur den Ähnlichkeitswert des Subgrapheb S_2 und des Fragments F_4. S_1 und F_3 werden als identisch betrachtet, da sie die gleiche chemische Struktur aufweisen. Auch hier spielt also die Lokalität chemischer Substrukturen eine größere Rolle für die Ähnlichkeitsbeschreibung durch SpaceLight, wobei dies hier zu einem vergleichsweise höheren Rang des Moleküls führt. Insgesamt kann diese Charakteristik von SpaceLight einen Vorteil gegenüber Ähnlichkeitssuchen mithilfe klassischer topologischer molekularer Fingerabdrücke darstellen.

Im letzten Analyseschritt untersuchen wir das Laufzeitverhalten von SpaceLight. Dafür verwenden wir openSUSE Leap 15 auf einer Intel Core i5-6500 64-Bit-Architektur mit 3,2 GHz und 16 GB Arbeitsspeicher. In Abbildung 4.3 ist die Laufzeit der SpaceLight Methode aufgeteilt auf die algorithmischen Teilschritte, sowie das Einlesen der Information aus der Datenbank, angeben. Der Paarungs- und Kombinationsschritt wurden in der Abbildung nicht berücksichtigt, da die durchschnittliche Laufzeit dieser Teilschritt im Millisekundenbereich liegt und damit im Vergleich vernachlässigbar ist. Zunächst fällt auf, dass die von SpaceLight beanspruchte Laufzeit hauptsächlich auf das Einlesen der Datenbank, sowie den Vergleichsschritt verteilt ist. Wenn drei Prozesse parallel verwendet werden, dann verringert sich die Laufzeit für den Vergleichsschritt auf ca. drei Sekunden oder weniger für beide Fingerabdrucks-Methoden und beide topologischen Fragmenträume. Auffällig ist auch, dass die Laufzeit bei der Verwendung der ECFP4 Methode geringer ist, als bei der fCSFP2.5 Methode. Dies deutet darauf hin, dass der fCSFP2.5 Fingerabdruck eines Fragments durchschnittlich mehr Identifikatoren enthält. Somit muss mehr Zeit auf das Laden der Fingerabdrücke aus der Datenbank, sowie für die Berechnung der fCSFP2.5 Fingerabdrücke der Subgraphen und ihren Vergleich aufgewandt werden. Der Partitionierungsschritt benötigt durchschnittlich weniger als eine Sekunde und hängt nicht von der verwendeten Methode für molekulare Fingerabdrücke ab. Die Hauptbeobachtung ist jedoch der geringe Laufzeitunterschied zwischen dem Enamine REAL Space und KnowledgeSpace. Obwohl der KnowledgeSpace mit 10^{14}

implizit beschriebenen Produkten den Enamine REAL Space um vier Größenordnung übersteigt, benötigt SpaceLight durchschnittlich sogar etwas weniger Laufzeit, wenn er als Eingabe verwendet wird. Dies zeigt, neben der Speichereffizienz kombinatorischer Bibliotheken im Allgemeinen und topologischer Fragmenträume im Speziellen, noch einmal die Motivation für die Verwendung kombinatorischer anstatt enumerierter Verfahren auf. Es ist nicht nur möglich eine weitaus größere Menge an Molekülen zu durchsuchen, [88] die dabei benötigten Ressourcen wachsen nur sublinear mit der Anzahl zu durchsuchender Moleküle.

4.3.4. Fragmente als molekulare Gerüste

Neben der Ähnlichkeitssuche bietet das Software-Modul SpaceLight ebenfalls die Möglichkeit Fragmente zu identifizieren, die eine chemische Substruktur und damit ein Gerüst eines Anfragemoleküls darstellen. Hierfür wird die Fingerabdruck-Teilmengenrelation der CSFP Methode und seiner Varianten ausgenutzt, die in Kapitel 3.2.5 beschrieben wurde. Es wird zunächst eine CSFP Variante und Parametrisierung gewählt und der molekulare Fingerabdruck des Anfragemoleküls berechnet. Nun wird für jedes Fragment des topologischen Fragmentraums überprüft, ob alle Identifikatoren seines molekularen Fingerabdrucks der gleichen CSFP Variante und Parametrisierung im Fingerabdruck des Anfragemoleküls enthalten ist. Alle Fragmente für die dies der Fall ist, werden nach absteigender Anzahl Schweratome sortiert und die k Fragmente mit der höchsten Anzahl an den Nutzer ausgegeben. Hierbei kann k vom Nutzer definiert werden. Durch die Fingerabdruck-Teilmengenrelation der CSFP Methode ist hierbei gesichert, dass alle Fragmente, die molekulare Gerüste darstellen, gefunden werden. Allerdings kann der Fingerabdruck des Anfragemoleküls den eines Fragments enthalten, ohne das dieses eine chemische Substruktur ist. Dies liegt an der substrukturbasierten Beschreibung durch topologische molekulare Fingerabdrücke. Je nachdem, ob die fCSFP, tCSFP oder iCSFP Variante verwendet wird, ergibt sich eine andere Betrachtung molekularer Gerüste. Mithilfe der fCSFP und tCSFP Varianten wird auch die Konfiguration von Linkern der Fragment berücksichtigt, somit spielt auch die Umgebung des Gerüsts innerhalb des Anfragemoleküls eine Rolle. Bei der Verwendung des iCSFP wird, ähnlich einer Suche nach der größten gemeinsamen chemischen Substruktur, [36] nur das Gerüst selbst und nicht dessen Umgebung betrachtet. Durch den effizienten Vergleich von molekularen Fingerabdrücken ist das Verfahren sehr schnell. Um falsch identifizierte molekulare Gerüste auszuschließen, kann nachträglich noch eine Berechnung der größten gemeinsamen chemischen Substruktur zum Anfragemolekül verwendet werden. In Abbildung 4.4 sind zwei Ergebnisse der Gerüstsuche

von SpaceLight im Enamine REAL Space mit Enzalutamid als Anfragemolekül gezeigt. die iCSFP1.5 Methode wurde verwendet, deshalb wird die Umgebung des Gerüsts in Enzalutamid nicht berücksichtigt. In (a) ist das Fragment mit den meisten Schweratome angegeben und es handelt sich um eine chemische Substruktur von Enzalutamid. In (b) ist ein weiteres Ergebnis gezeigt, wobei das Fragment keine chemische Substruktur von Enzalutamid ist. Allerdings sind alle chemischen Substrukturen des Fragments mit maximal fünf Atomen ebenfalls in Enzalutamid enthalten. Das Benzol mit dem jeweils rechten und linken Teil des Fragments ist in (b) in Enzalutamid markiert. Im Fragment ist eine chemische Substruktur mit sechs Atomen markiert, die nicht in Enzalutamid vorkommt. Deshalb würde dieses Fragment bei der Berechnung mithilfe des iCSFP1.6 nicht mehr zu den Resultaten hinzugefügt werden.

4.3.5. Abgrenzung zu existierenden Verfahren

Das beschriebene Verfahren SpaceLight ermöglicht erstmalig die Ähnlichkeitssuche in kombinatorischen Bibliotheken mithilfe topologischer molekularer Fingerabdrücke. Durch seinen kombinatorische Ansatz grenzt sich die SpaceLight methodisch stark von allen existierenden Verfahren zur Ähnlichkeitssuche in enumerierten Bibliotheken ab und ist in der Lage eine weitaus größere Menge von Molekülen zu durchsuchen. Allerdings konnte in [D2] und Kapitel 4.3.3 gezeigt werden, dass die von SpaceLight generierten Ergebnisse auf kleineren, enumerierbaren topologischen Fragmenträumen mit den Ergebnisse einer Ähnlichkeitssuche mithilfe enumerierter Moleküllisten und klassischer topologischer molekularer Fingerabdrücke korreliert ist. Jedoch betrachtet die Methode SpaceLight die Lokalität chemischer Substruktur zu einem größeren Maß.

In der Pythonbibliothek SynthI [75] ist es möglich mithilfe der ECFP Methode [98] eine enumerierte Bibliothek zu erzeugen, die zu einem Anfragemolekül ähnliche Moleküle enthält. Dies geschieht ebenfalls mit einem kombinatorischen Ansatz. SynthI wurde nach der Veröffentlichung von SpaceLight publiziert und verwendet ebenfalls molekulare Fingerabdrücke als eines von zwei Ähnlichkeitsmaßen. Allerdings werden zur Unterteilung der Anfrage eine definierte Menge an Reaktionen in einem retrosynthetischen Ansatz angewendet. Damit wird die Anzahl der möglichen Unterteilungen eingeschränkt und die chemische Umgebung an der Unterteilung entspricht derjenigen in den enumerierten Molekülen. Des Weiteren wird den generierten Molekülen kein Ähnlichkeitswert und damit auch keine Ordnung zugewiesen. Diese kann allerdings mithilfe einer klassischen

sequenziellen Ähnlichkeitssuche auf den enumerierten Ergebnissen nachträglich erzeugt werden.

Die zwei Methoden FTress-FS [53] und SpaceMACS [119] operieren auf klassischen Fragmeträumen, sind aber methodisch wohl die nächsten Verwandten des SpaceLight-Verfahrens. Im Gegensatz zu SpaceLight verwenden sie allerdings einen Paarungsansatz und ausschließlich azyklische Bindungen des Anfragemoleküls zur Partitionierung. In einem Molekulargraphen mit n Atomen existieren maximal $n-1$ viele azyklische Bindungen. [92] Somit existieren auch höchstens $n-1$ Partitionen in zwei Subgraphen. Ein Ring mit n Atomen besitzt bereits n^2-n viele Paare von Bindungen und dadurch ebenso viele Partitionen in zwei Subgraphen. Dieses Beispiel zeigt die erhöhte Anzahl von Partitionen, die von SpaceLight betrachtet werden durch die erlaubte Ringbildung über Fragmentgrenzen. Damit ist SpaceLight auch bisher das einzige Verfahren für die Ähnlichkeitssuche in makrozyklischen kombinatorischen Bibliotheken. Allerdings steigt die benötigte Laufzeit von SpaceLight mit steigender Größe und Dichte der Anfragemoleküle zu einem höheren Maß im Vergleich zu den Methoden FTrees-FS und SpaceMACS, falls Partitionen von Ringen vorgenommen werden. Dies stellt eine potenzielle Begrenzung der Anwendbarkeit von SpaceLight, beispielsweise im Bereich großer makrozyklischer Naturstoffe dar.

Neben diesem methodischen Unterschied, verwenden FTrees-FS, SpaceMACS und SpaceLight eine unterschiedliche Definition molekularer Ähnlichkeit. FTrees-FS benutzt die grobgranulare Graphstruktur und pharmakophorbasierte Beschreibung der Feature Trees Methode. [90] SpaceMACS definiert die Ähnlichkeit zweier Moleküle über große gemeinsame chemische Substrukturen. SpaceLight hingegen betrachtet die Übereinstimmung vieler kleinerer gemeinsamer chemischer Substrukturen. Schmidt et al. konnten in [119] an einem Beispiel zeigen, dass die drei Verfahren unter jeweils 150.000 gefundenen ähnlichen Molekülen nur höchstens 15% ihrer Ergebnisse miteinander gemein hatten. Damit ergänzen sich die drei Methoden mit ihrer ganz eigenen Beschreibung molekularer Ähnlichkeit und bieten eine breite Auswahl für die Ähnlichkeitssuche in kombinatorischen Bibliotheken.

4.3.6. Ausblick

Das SpaceLight Verfahren ermöglicht eine neue Beschreibung molekularer Ähnlichkeit in kombinatorischen Bibliotheken. Die Ergebnisse sind korreliert mit denen einer Ähnlichkeitssuche mithilfe klassischer molekularer Fingerabdrücke, allerdings können sich die absoluten angenommen Ähnlichkeitswerte unterscheiden. Für eine Methode molekularer

Fingerabdrücke wurden bereits Ähnlichkeitsgrenzwerte ermittelt, ab denen mit einer Bioaktivität gerechnet werden kann. [123] Eine ähnliche Untersuchung wäre auch für die SpaceLight Methode denkbar.

Die SpaceLight Methode wurde in die Codebasis NAOMI [54] implementiert, sie ist modular aufgebaut und beide Suchverfahren sowie jeder Teilschritt könnten prinzipiell unabhängig von den anderen Teilschritten verändert werden. Dadurch ergeben sich eine Reihe von Erweiterungen und Anpassungen die mit vergleichsweise geringem Aufwand integriert werden könnten. Beispielsweise könnte für die Bestimmung von Fragmenten, die molekulare Gerüste darstellen, durch einen zusätzlichen algorithmischen Schritt erweitert werden. Die in NAOMI implementierten Methoden zur Bestimmung größter gemeinsamer chemischer Substrukturen [118] könnten auf die gefunden Fragmente angewendet werden, um falsch-positive Ergebnisse wie in Abbildung 4.4 (b) auszuschließen.

Im Partitionierungsschritt der Ähnlichkeitssuche könnten andere Filter für die Partitionsbildung verwendet werden. Hier könnten zB. die chemischen Reaktionen zum Aufbau des topologischen Fragmentraumes mit der chemischen Umgebung von Subgraphen verglichen werden. Damit wären die generierten Partition nicht nur topologisch, sondern auch auf Ebene chemischer Substrukturen, mit Fragmentkombinationen des Raumes vergleichbar. Auch die erlaubte Differenz für die Anzahl Schweratome zwischen kompatiblen Paaren von Fragmenten und Subgraphen könnte angepasst werden. Im Paarungsschritt könnte auch ein Topologiewert kleiner als eins zugelassen werden. Dadurch würde die Zahl der zugelassenen Partitionen erhöht und es kann eine gewichtete Summe für den finalen Ähnlichkeitswert der Fragmentkombinationen verwendet werden. Entsprechende Funktionalität existiert bereits in der Implementation in NAOMI.

Auch im Vergleichsschritt könnten die bestehenden Verfahren für andere Konzepte molekularer Ähnlichkeit angepasst werden. Am einfachsten wäre die Verwendung anderer, beispielsweise pharmakophorbasierter, Methoden für molekulare Fingerabdrücke. Des Weiteren könnten auch Methoden des maschinellen Lernens verwendet werden. Die Modelle müssten allerdings ebenfalls in der Lage sein, auf der Ebene von Fragmenten zu operieren. Sie könnten dann beispielsweise die präprozessierten molekularen Fingerabdrücke der Fragmente als Eingabe verwenden. Es könnten auch andere Klassen von Molekulardeskriptoren angewandt werden. Beispielsweise wäre die Verwendung des räumlichen Molekulardeskriptors RVM [132], wie in Kapitel 4.1 beschrieben, denkbar. Dadurch würde die erste Ähnlichkeitssuche auf Basis der Überlagerung räumlicher Konformationen in kombinatorischen virtuellen chemischen Bibliotheken ermöglicht. Statt

der Unterteilung eines Anfragemoleküls, könnte ein vergleichbares Prinzip angewendet werden, um beispielsweise eine Bindetasche oder ein räumliches Pharmakophormodell zu partitionieren. Da in diesem Kontext allerdings räumliche statt graphstrukturelle Partitionen gebildet werden würden, müssten geometrische statt graphtheoretische Konzepte Anwendung finden.

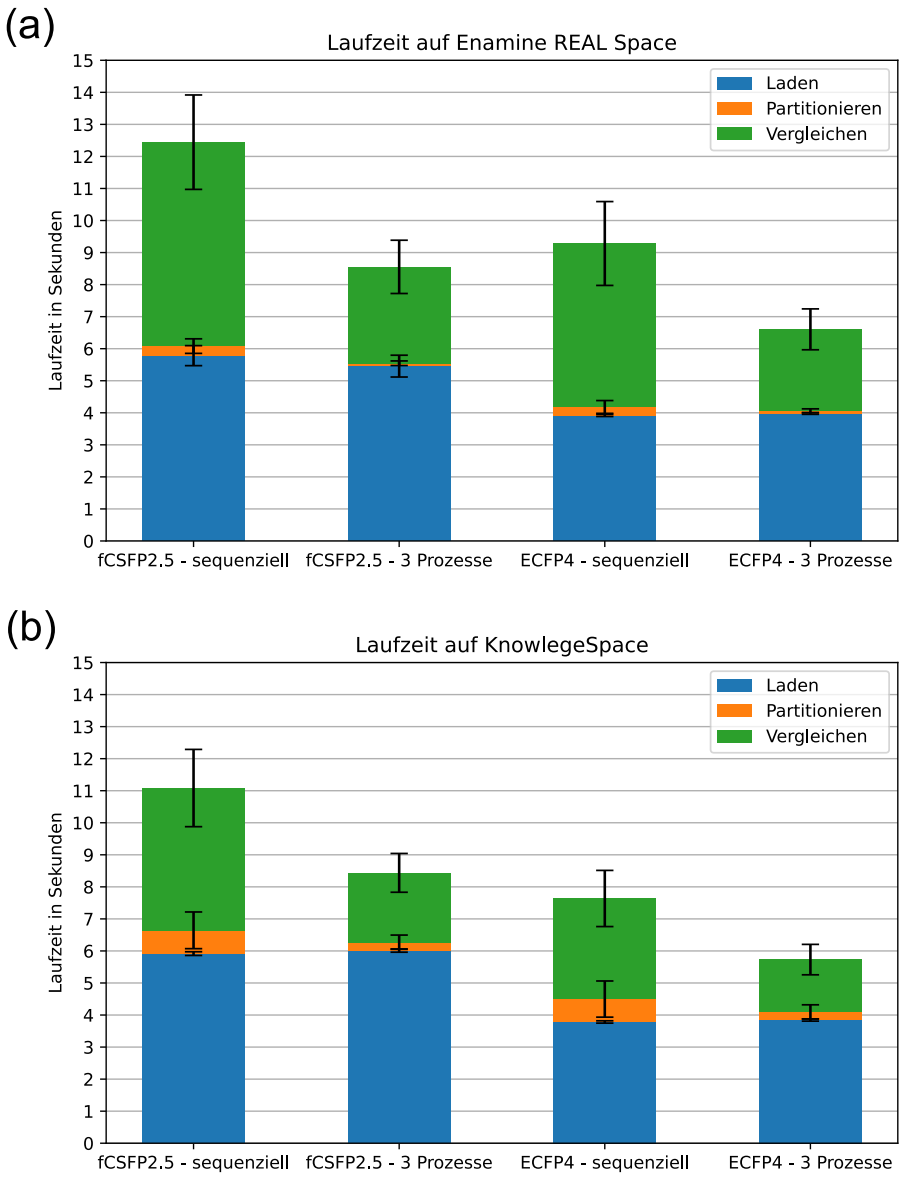


Abbildung 4.3.: Die durchschnittlichen Laufzeiten und Standardabweichungen der SpaceLight Methode mithilfe von 500 zufälligen Anfragemolekülen des "lead-like"Katalogs der ZINC Datenbank. [44] Die Laufzeiten sind nach der verwendeten Zeit für das Laden der benötigten Daten aus der Datenbank, sowie für den Partitionierungs- und Vergleichsschritt aufgeteilt. Für die Suche wurde jeweils ein einzelner Prozess und drei parallele Prozesse, sowie die ECFP4 und fCSFP2.5 Methode verwendet. In (a) sind die Laufzeiten für den Enamine REAL Space [81] angegeben und in (b) die Laufzeiten für den KnowledgeSpace. [84] Aus [D2] entnommen, angepasst und ins Deutsche übersetzt.

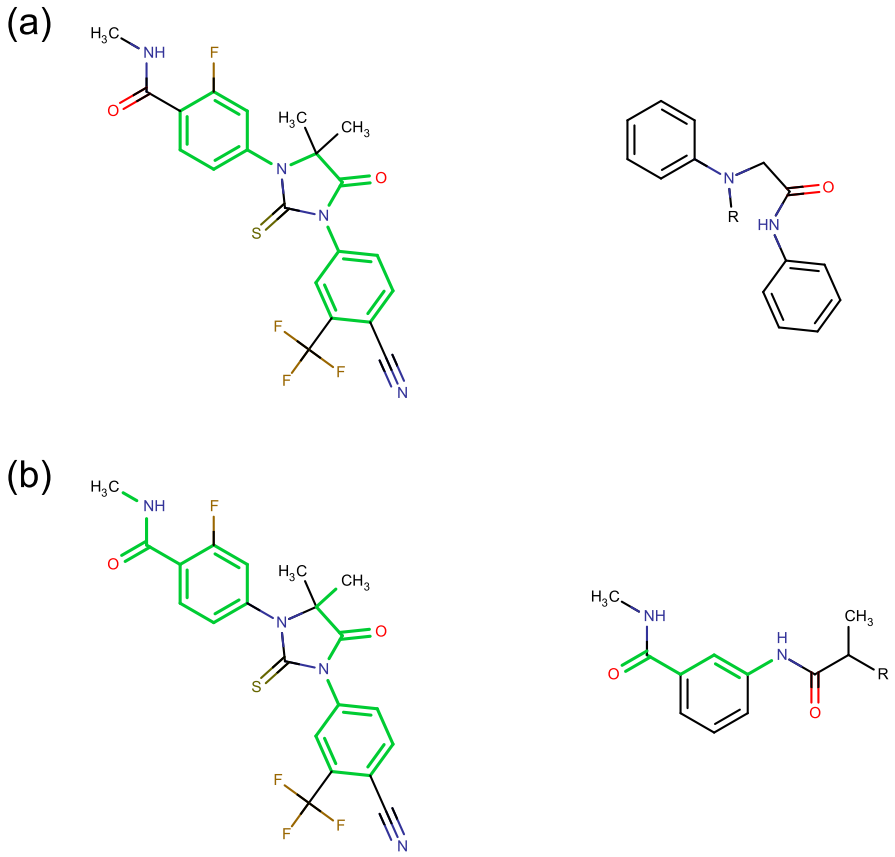


Abbildung 4.4.: Zwei Ergebnisse der Gerüstsuche im Enamine REAL Space mit Enzalutamid als Anfragemolekül und der iCSFP1.5 Methode. (a) Das Ergebnisfragment mit der größten Anzahl Schweratome. In (b) ein weiteres Ergebnis, dass allerdings keine chemische Substruktur darstellt. Die Bindungen der übereinstimmenden Teilstrukturen sind in Enzalutamid grün markiert. Im Fragment in (b) sind die Bindungen einer chemische Substruktur in grün markiert, die nicht in Enzalutamid enthalten ist. Aus [D2] entnommen und angepasst.

5. Schnittmengen kombinatorischer Bibliotheken

Dieses Kapitel handelt von Methoden zur Bestimmung der Schnittmenge gemeinsamer Moleküle verschiedener virtueller chemischer Bibliotheken. Im Gegensatz zum vorangegangen Kapitel 4 geht es bei dieser Fragestellung nicht um Suchverfahren innerhalb eines konkreten Projekts im Wirkstoffentwurf, sondern um einen allgemeinen Vergleich mehrerer virtueller chemischer Bibliotheken.

Die Berechnung einer Schnittmenge kann aus verschiedenen Gründen interessant sein. Zunächst können die Ergebnisse hilfreich bei der Auswahl einer Bibliothek für eine bestimmte Anwendung sein. Falls beispielsweise ein Großteil einer Bibliothek in einer anderen, größeren Bibliothek enthalten ist, sollte eventuell die größere Bibliothek zB. für eine Ähnlichkeitssuche bevorzugt werden. Eine Bibliothek kann gegebenenfalls mithilfe des Komplements einer anderen Bibliothek erweitert werden, um eine größere Abdeckung des chemischen Raumes zu erzielen. In virtuellen chemischen Bibliotheken können auch Informationen über die enthaltenen Moleküle inkludiert werden. Für die Moleküle in der Schnittmenge von Bibliotheken kann dieses Wissen verknüpft werden. Beispielsweise können für Moleküle in der Schnittmenge alternative Synthesewege identifiziert werden, wenn das nötige chemische Wissen hinterlegt ist.

Es existieren eine Reihe öffentlich zugänglicher Plattformen, die Bibliotheken aus verschiedenen Quellen zusammenfassen und verknüpfen. Die so entstandene vereinigte Bibliothek ist nicht nur größer, sie kann auch unterschiedliche Informationen zu den Molekülen der Schnittmengen zusammenführen. Die Datenbank ZINC [44] und eMolecules [59] verknüpfen die Kataloge verschiedener Chemieunternehmen zu einer Bibliothek. Für die Schnittmenge dieser Kataloge kann der Nutzer Preise sowie Lieferzeiten vergleichen und eine Auswahl treffen. Die Datenbanken ChEMBL [43] und PubChem [45] fassen Moleküle und für sie verfügbare Bioaktivitäts- und Experimentaldaten aus Publikationen, klinischen Studien und verwandten Quellen zusammen. Für Moleküle die in mehreren

Quellen vorkommen kann so beispielsweise ermittelt werden, ob und für welche verschiedenen Krankheiten das Molekül als Wirkstoffkandidat verwendet wurde. Zusätzlich kann untersucht werden in welchen unterschiedlichen Publikationen es erforscht wurde bzw. welche bekannten Eigenschaften es besitzt. Die PDBbind Datenbank [141] fasst die Bindungsaffinitäten von Protein-Ligand-Komplexen der PDB Datenbank [35] aus der wissenschaftlichen Literatur zusammen. Falls zu einem Protein-Ligand-Komplex mehrere Publikationen existieren, so kann der Nutzer die Ergebnisse vergleichen.

5.1. Existierende Verfahren zur Schnittmengenberechnung chemischer Bibliotheken

Eine naheliegende Methode zur Schnittmengenberechnung scheint die Verwendung eines in Kapitel 4.1 oder [D2] beschriebenen Verfahrens zur Ähnlichkeitssuche. Zwei Moleküle, die einen Ähnlichkeitswert von eins erhalten, werden als gleichwertig angesehen und zur Schnittmenge hinzugefügt. Dies hat allerdings zwei Nachteile: Zum einen müssen zwei Moleküle nicht identisch sein nur weil es ihre Molekulardeskriptoren sind. Beispielsweise können zwei Moleküle dasselbe Pharmakophormodell aufweisen, obwohl sie strukturell unterschiedlich sind. Zum anderen ist dieser Ansatz nicht unbedingt effizient, da die Bestimmung der Ähnlichkeit zwischen zwei Molekülen eine komplexere Frage, als die nach ihrer Identität darstellt. Darüber hinaus müsste im Fall der Schnittmengenberechnung zweier kombinatorischer Bibliotheken eine von ihnen enumeriert werden, um alle ihre Moleküle als Anfrage zu verwenden. Um diese Problem zu umgehen werden für enumerierte Bibliotheken andere Ansätze verwendet und in Kapitel 5.3 werden eigene Methodiken entwickelt, die speziell auf die Berechnung von Schnittmengen kombinatorischer Bibliotheken angepasst ist. Hierbei wird die CSFP Methode aus Kapitel 3 verwendet. Anstatt molekulare Fingerabdrücke von Produkten zur Überprüfung von Identität einzusetzen, werden mit ihrer Hilfe effizient notwendige Bedingungen für Fragmente getestet, um die Schnittmengenberechnung nicht enumerierbarer topologischer Fragmenträume zu ermöglichen.

Alle existierenden Methoden zur exakten Schnittmengenberechnung operieren auf enumerierten Bibliotheken. Hierfür wird der Schnitt zweier enumerierter Bibliotheken berechnet, indem ihre Moleküle paarweise auf Identität überprüft werden. Zu diesem Zweck ist es notwendig alle Moleküle in eine kanonische Repräsentation zu überführen.

Diese kann für zwei Moleküle verglichen werden und ihre Identität wird bei übereinstimmender Repräsentation festgestellt. Im Folgenden wird deshalb kurz auf existierende Verfahren zur kanonischen Repräsentation von Molekülen eingegangen.

Die Notation als Summenformel eignet sich hierfür nicht, da sie die Struktur von Molekulargraphen nicht berücksichtigt. Beispielsweise haben Ethanol und Dimethylether die gleiche Summenformel C_2H_6O. Im Gegensatz dazu ist es mithilfe der weit verbreiteten Nomenklatur der IUPAC [1] möglich unterschiedliche Notationen für das gleiche Molekül zu bilden. Damit sind beide Verfahren zur Generierung einer kanonischen Repräsentation ungeeignet.

In [110] wurde eine Methode zur Berechnung einer kanonischen Molekülrepräsentation in SMILES Notation [32] entwickelt. Diese Form der kanonischen Repräsentation ist weitverbreitet und leicht lesbar. Ein weiterer Vorteil dieser Darstellung ist die Möglichkeit der Rekonstruktion des repräsentierten Moleküls. Allerdings wurde die SMILES Notation von dem Privatunternehmen Daylight entwickelt und dadurch ist die kanonischen Repräsentation, speziell stereochemischer Eigenschaften, nicht vollkommen standardisiert. Deshalb generieren unterschiedliche Plattformen verschiedene Notationen. [54], [71], [73] Die Notation InChI [33] ist eine von der Vereinigung IUPAC [1] standardisierte Molekülrepräsentation, die auch in kanonischer Form gebildet werden kann. Damit behebt sie Probleme der teilweise proprietären SMILES Notation. Allerdings ist das ursprüngliche Molekül nicht immer aus der InChI Notation rekonstruierbar. [33] Mithilfe der in Kapitel 3 beschriebenen CSFP Methode ist es auch möglich den Identifikator eines kompletten Molekulargraphen zu berechnen und so das Molekül kanonisch zu repräsentieren. Da das Verfahren die CANGEN Methode zur Berechnung der kanonischen SMILES Notation [110] verwendet, sind diese Notationen methodisch verwandt. Allerdings kann das ursprüngliche Molekül nicht aus dem CSFP Identifikator seines Molekulargraphen rekonstruiert werden.

Da die Schnittberechnung mithilfe kanonischer Repräsentationen individueller Moleküle letztlich auf der Enumeration virtueller chemischer Bibliotheken beruht, ist dieser Ansatz für große, nicht enumerierbare kombinatorische Bibliotheken ungeeignet. Es existieren zwar keine Methoden zur exakten Berechnung ihrer Schnittmenge in der Literatur, aber in [142] verwendeten Lessel und Lemmen ein Stichprobenverfahren, um die Schnittmenge der drei kombinatorischen virtuellen chemischen Bibliotheken BICLAIM [85], Enamine REAL Space [81] und KnowledgeSpace [84] zu schätzen. Hierfür verwendeten sie 100 zufällig ausgewählte zugelassene Medikamente als Anfragemoleküle für eine Ähn-

lichkeitssuche mithilfe des Verfahrens FTrees-FS. [53] Für jedes Anfragemolekül wurden die ähnlichsten 10.000 Produkte für jeden der drei kombinatorischen virtuellen chemischen Bibliotheken bestimmt. Die Ergebnisse wurden für jede kombinatorische Bibliothek zu einer enumerierten virtuellen chemischen Teilbibliothek von höchstens 1.000.000 finalen Molekülen zusammengefasst. Für diese enumerierten Teilbibliotheken konnte nun die Schnittmenge mithilfe einer kanonischen Repräsentation bestimmt werden. Weniger als 1.700 Moleküle waren in mehr als einer Bibliothek enthalten und nur drei Moleküle wurden in der gemeinsamen Schnittmenge aller drei Bibliotheken gefunden. Da die erzeugten Teilbibliotheken allerdings um einige Größenordnungen kleiner sind, als die ursprünglichen kombinatorischen Bibliotheken, muss diese Schätzung nicht zwangsläufig eine gute Approximation an ihre exakte Schnittmenge darstellen.

5.2. Herausforderung der Schnittmengenberechnung kombinatorischer Bibliotheken

Die Enumeration aller Moleküle kombinatorischer Bibliotheken ist aufgrund ihrer Größe nicht immer möglich. Deshalb muss ein Verfahren für ihre Schnittberechnung in der Lage sein auf der Repräsentation chemischer Bausteine zu operieren. Ein naheliegender Ansatz ist, ähnlich dem Vorgehen für Moleküle bei der Schnittberechnung enumerierter Bibliotheken, die chemischen Bausteine auf strukturelle Identität zu überprüfen. Dies kann allerdings zu falsch-negativen Resultaten führen, wie in Abbildung 5.1 beispielhaft aufgezeigt ist. Hier werden zwei unterschiedliche Synthesewege gezeigt, die das gleiche Produkt ergeben. Falls die zwei Synthesewege durch die Repräsentation der verwendeten chemischen Bausteine und Reaktionen in zwei kombinatorischen Bibliotheken beschrieben wären, wäre das Produkt in beiden Bibliotheken enthalten und somit Teil ihrer Schnittmenge. Die verwendeten chemischen Bausteine sind allerdings nicht strukturell identisch und somit wäre auch ihre kanonische Repräsentation zB. in SMILES Notation unterschiedlich. Somit kann das simple Überprüfen auf Identität im Kontext der Schnittmengenberechnung kombinatorischer Bibliotheken nicht verwendet werden.

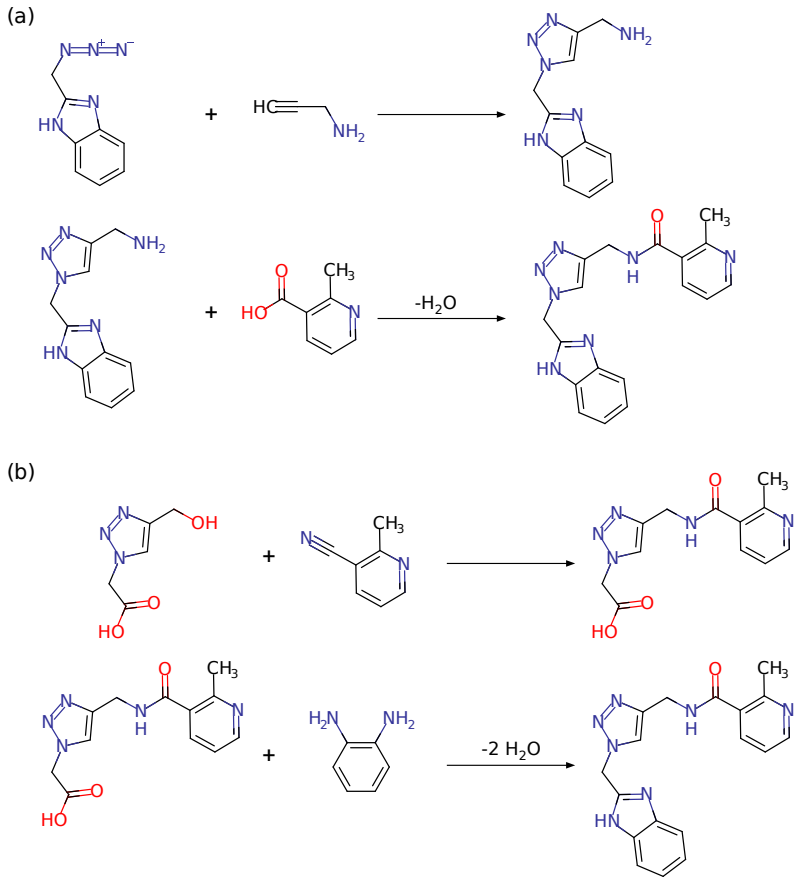


Abbildung 5.1.: Zwei Synthesewege für das gleiche Produkt. In (a) wird zunächst eine Diels-Alder-Reaktion [91] und danach eine Amidkupplung durchgeführt. In (b) folgt eine Kondensationsreaktion auf eine Ritter-Reaktion. Aus [D3] entnommen und angepasst.

5.3. Beschreibung des algorithmischen Verfahrens SpaceCompare

In diesem Kapitel wird die Methode SpaceCompare beschrieben, wie sie in [D3] vorgestellt wurde. SpaceCompare ist erstmals in der Lage die Schnittmenge zweier nicht enumerierbarer kombinatorischer Bibliotheken exakt zu bestimmen, sofern diese Schnittmenge klein genug ist, um sie zu enumerieren. Damit grenzt es sich von allen existierenden Verfahren methodisch stark ab. Auf die genauen Limitierungen von SpaceCompare wird in Kapitel 5.3.5 eingegangen. Das Verfahren operiert auf topologischen Fragmenträumen und betrachtet chemische Substrukturen der im Raum enthaltenen Fragmente und ihrer

Kombinationen. Dafür wird die CSFP Methode aus Kapitel 3 verwendet und dessen Erfüllung der Fingerabdruck-Teilmengenrelation spielt hierbei eine zentrale Rolle.

5.3.1. Arten chemischer Substrukturen

Für die Schnittmengenberechnung verwendet SpaceCompare chemische Substrukturen. Diese können durch Identifikatoren der CSFP Methode repräsentiert und so effizient prozessiert werden. Für das gesamte Kapitel 5 werden zwei chemische Substrukturen als unterschiedlich betrachtet, wenn ihre mit der fCSFP Variante berechneten Identifikatoren sich unterscheiden. Bisher wurden für eine Substruktur die enthaltenen Atom- und Bindungstypen, sowie den unterliegenden Molekulargraphen betrachtet. Jetzt sind zusätzlich die Valenz der enthaltenen Atome sowie deren Aromatizität und Konnektivität innerhalb und außerhalb der Substruktur mit eingeschlossen. Wie das in Kapitel 4 beschriebene Verfahren SpaceLight, soll SpaceCompare zunächst auf der Menge der Fragmente topologischer Fragmenträume operieren. Da aber das letztendliche Ziel die Bestimmung der Schnittmenge der Produkte zweier Räume ist, spielt die Beziehung der verwendeten chemischen Bausteine zu den finalen Produkten eine große Rolle. Wenn beispielsweise eine Substruktur eines chemischen Bausteins während einer Reaktion verändert wird, dann sollte diese Substruktur zur Bestimmung der Schnittmenge nicht verwendet werden.

Gegeben sei ein Syntheseweg eines Produkts und die bei der Synthese genutzten chemischen Bausteine. Jede chemische Substruktur eines Bausteins oder des Produkts fällt in eine von drei Kategorien. Eine chemische Substruktur ist *stabil*, wenn sie sowohl im Produkt, als auch in mindestens einem der chemischen Bausteine vorkommt. Eine chemische Substruktur ist *instabil*, wenn sie in mindestens einem chemischen Baustein existiert, allerdings im Produkt nicht vorkommt. Schlussendlich ist eine Substruktur *kreuzend*, wenn sie im Produkt vorhanden ist, allerdings in keiner der chemischen Bausteine vorkommt.

Mithilfe dieser Kategorisierung chemischer Substrukturen lässt sich auch die in Kapitel 5.2 beschriebene Herausforderung weiter erörtern. Instabile und kreuzende Substrukturen führen zu den beschriebenen falsch-negativen Resultaten, wenn die Identität chemischer Bausteine als Ansatz verwendet wird. Alle chemischen Bausteine beider Synthesewege aus Abbildung 5.1 enthalten instabile chemische Substrukturen. Beispielsweise stellen die Alken- und Carboxygruppe aus (a) und die primären Amine und die Carboxygruppe aus (b) instabile Substrukturen dar. Zusätzlich sind unterschiedliche chemische Substrukturen

des finalen Moleküls kreuzend in (a) und (b). Beispielsweise ist Triazol in (a) eine kreuzende und in (b) eine stabile Substruktur. Wiederum ist Benzimidazol in (a) stabil und in (b) kreuzend. Diese Unterschiede in der Kategorisierung chemischer Substrukturen müssen im weiteren Verlauf berücksichtigt werden.

Stabile Substrukturen können für die Schnittberechnung kombinatorischer Bibliotheken auf Basis chemischer Bausteine ohne Weiteres verwendet werden. Für instabile und kreuzende Substrukturen müssen weitere Beobachtungen und Verfahren aufgestellt werden. Werden anstatt chemischer Bausteine ihre Repräsentation als Fragmente verwendet, kann das Problem instabiler chemischer Substrukturen umgangen werden. In Kapitel 2.3.2 wurde ausgeführt, dass die Fragmente eines topologischen Fragmentraums die Substruktur-Teilmengenrelation erfüllen. Alle chemischen Substrukturen eines Fragments sind auch in allen Produkten enthalten, die mithilfe des Fragments gebildet werden können. Somit enthalten Fragmente, unabhängig vom betrachteten Produkt, keine instabilen Substrukturen. Produkte können aber weiterhin kreuzende Substrukturen aufweisen.

5.3.2. Erweiterte Fingerabdrücke und Überdeckung

Die Betrachtung kreuzender Substrukturen ist für die Schnittmengenberechnung auf Basis chemischer Substrukturen notwendig, wie im vorangegangen Kapitel ausgeführt wurde. Deshalb werden im Rahmen der SpaceCompare Methode alle kreuzenden Substrukturen jedes Produkts eines topologischen Fragmentraums enumeriert. Da theoretisch jedes Produkt mindestens eine eigene kreuzende Substruktur enthalten kann, die in keinem anderen Produkt existiert, kommt dies zunächst asymptotisch einer Enumeration des topologischen Fragmentraums gleich. Zum einen wurden allerdings nur Substrukturen betrachtet, die höchstens sechs Atome enthalten, wodurch die Anzahl Substrukturen begrenzt ist. [D1], [143] Zum anderen schneiden kreuzende Substrukturen ihrer Definition nach zwei oder mehr Fragmente und liegen damit im Bereich einer Reaktionsstelle. Da die gleichen Reaktionen auf alle chemischen Bausteine angewandt werden, die zu einem Topologiegraphen und seinen Fragmenten führen, begrenzt dies die Menge an unterschiedlichen kreuzenden Substrukturen weiter, wenn reale Syntheseprotokolle zur Erzeugung eines topologischen Fragmentraums verwendet werden. Die mögliche Anzahl kreuzender Substrukturen kann also als begrenzt angenommen werden. Falls es nun möglich ist bei der Enumeration aller kreuzenden Substrukturen die Enumeration aller Produkte zu vermeiden, skaliert die asymptotische Laufzeit des Verfahrens mit der Anzahl der Fragment eines topologischen Fragmentraums, anstatt mit der Anzahl

seiner Produkte. Dies wird sich auch in den Laufzeiten der SpaceCompare Methode in Kapitel 5.3.4 zeigen. In Anhang B.4 wird eine effiziente Methode zur Enumeration aller kreuzender Substrukturen eines topologischen Fragmentraums beschrieben. Bei der Enumeration werden alle kreuzenden Substrukturen aus unterschiedlichen Produkten zusammengefasst, die den gleichen fCSFP Identifikator aufweisen. Zusätzlich werden alle Fragmente abgespeichert, die einen Schnitt mit der zusammengefassten kreuzenden Substruktur aufweisen.

Nun werden für jedes Fragment eines topologischen Fragmentraums dessen eigene, stabile Substrukturen und alle das Fragment schneidende kreuzende Substrukturen in einer Menge zusammengefasst. Hierbei werden nur Substrukturen mit höchstens sechs enthaltenen Atomen betrachtet. Alle fCSFP Identifikatoren der Substrukturen dieser Menge ergeben den *erweiterten Fingerabdruck* eines Fragments. Erweiterte Fingerabdrücke aller Fragmente können nach der Enumeration aller kreuzenden Substrukturen durch das Verfahren aus Anhang B.4 einfach gebildet werden. Für die eigenen Substrukturen der Fragmente können die präprozessiertee fCSFP1.6 Fingerabdrücke der Fragmente verwendet werden. Diese werden dann durch die fCSFP Identifikatoren der kreuzenden Substrukturen für die abgespeicherten schneidenden Fragmente erweitert.

Nach ihrer Erzeugung sind die erweiterten Fingerabdrücke somit eine Eigenschaft einzelner Fragmente und unabhängig von bestimmten Produkten. Dies ist, dem algorithmischen kombinatorischen Paradigma folgend, allgemein wichtig, um die vollständige Enumeration kombinatorischer Bibliotheken möglichst zu vermeiden und auf der Ebene chemischer Bausteine zu operieren. Erweiterter Fingerabdrücke beschrieben sowohl stabile, als auch kreuzende Substrukturen, weshalb ihnen eine zentrale Rolle im algorithmischen Verfahren SpaceCompare zukommt. Zwei Fragmentkombination, die dasselbe Produkt erzeugen, können nun in Beziehung gesetzt werden. Seien dazu $\{x_1, \ldots, x_n\}$ und $\{y_1, \ldots, y_m\}$ zwei Fragmentkombinationen, die beide das Produkt z erzeugen. Sei weiterhin $F(x)$ der fCSFP1.6 Fingerabdruck des Produkts oder Fragments x und $E(x)$ der erweiterte Fingerabdruck des Fragments x. Beide seien hier als Mengen von Identifikatoren repräsentiert. Aufgrund der Erfüllung der Fingerabdruck-Teilmengenrelation des CSFP und seiner Varianten gilt

$$F(x_1) \cup \cdots \cup F(x_n) \subseteq F(z) \tag{5.1}$$

Da jede kreuzende Substruktur des Produkts z bezüglich der Kombination $\{y_1, \ldots, y_m\}$ mindestens zwei der Fragmente y_i schneidet, ist auch jede kreuzende Substruktur in min-

destens zwei ihrer erweiterten Fingerabdrücke enthalten. Da alle übrigen Substrukturen von z bezüglich $\{y_1, \ldots, y_m\}$ stabil sind, gilt

$$F(z) \subseteq E(y_1) \cup \cdots \cup E(y_m) \tag{5.2}$$

Da die erweiterten Fingerabdrücke kreuzende Substrukturen unterschiedlicher Produkte enthalten können, kann hier die rechte Seite von 5.2 auch eine echt Obermenge darstellen. Werden 5.1 und 5.2 zusammengefasst, ergibt sich

$$F(x_1) \cup \cdots \cup F(x_n) \subseteq E(y_1) \cup \cdots \cup E(y_m) \tag{5.3}$$

Die Beziehung 5.3 lässt sich auch mit invertierten Rollen von $\{x_1, \ldots, x_n\}$ und $\{y_1, \ldots, y_m\}$ aufstellen. Die Fragmentkombination $y = \{y_1, \ldots y_j\}$ *überdeckt* die Fragmentkombination $x = \{x_1, \ldots x_i\}$, wenn $F(x_1) \cup \cdots \cup F(x_i) \subseteq E(y_1) \cup \cdots \cup E(y_j)$. Hierbei kann sowohl x als auch y aus nur einem Fragment bestehen. y ist eine *minimale Überdeckung* von x, falls x von keiner echten Teilkombination $y' \subset y$ überdeckt wird. Es existiert nun eine notwendige Bedingung an zwei Fragmentkombinationen, wenn sie dasselbe Produkt erzeugen und damit im Schnitt zweier topologischer Fragmenträume liegen. Der Vollständigkeit halber wird darauf hingewiesen, dass es sich hier um keine hinreichende Bedingung handelt, wie in Kapitel 5.3.4 gezeigt wird.

5.3.3. Algorithmische Schritte der SpaceCompare Methode

Die Methode SpaceCompare leitet sich nun direkt aus dieser Definition ab und ist in [D3] ausführlich beschrieben. Die Eingabe besteht aus zwei topologischen Fragmenträumen, für die ihre Schnittmenge der gemeinsamen finalen Moleküle bestimmt wird. Der Raum mit einer geringeren Anzahl implizit beschriebener Produkte sei als kleiner bezeichnet und der anderen Raum als größer. SpaceCompare beinhaltet zwei algorithmische Schritte. In beide Schritten werden initial die zwei Räume in einer Reihenfolge verarbeitet. Danach wird das Verfahren mit invertierten Rollen der zwei Räume erneut durchgeführt.

Im Überdeckungsschritt, werden alle einzelnen Fragmente des größeren Raums identifiziert, die durch eine Fragmentkombination aus dem kleineren Raum überdeckt werden können. Dafür werden zunächst die erweiterten Fingerabdrücke der Fragmente des kleineren Raumes berechnet. Sie werden in einer Datenstruktur abgespeichert, die die erweiterten Fingerabdrücke indexiert nach Identifikatoren enthält. Mithilfe dieser Datenstruktur werden dann die Überdeckungen bestimmt. Danach wird das Vorgehen mit

vertauschten Rollen des kleinen und größeren Raums erneut durchgeführt. Allerdings werden nun nur noch die erweiterten Fingerabdrücke der Fragmente des größeren Raums verwendet, die selbst überdeckt wurden. Deshalb werden auch nur kreuzende Substrukturen enumeriert, die mithilfe von Kombinationen dieser Fragmente erzeugt werden können. Der Überdeckungsschritt endet mit zwei Datenstrukturen, eine für jeden Raum, aus überdeckten Fragmenten und allen ihren minimalen Überdeckungen aus dem anderen topologischen Fragmentraum.

Im Kombinationsschritt werden nun die überdeckten Fragmente rekursiv zu insgesamt überdeckten Fragmentkombinationen zusammengefügt. Hierbei können nur Fragmentkombinationen, sowohl überdeckt, als auch überdeckend, vereinigt werden, wenn sie aus dem gleichen Topologiegraphen stammen und für keinen Knoten unterschiedliche Fragmente enthalten. Nur in so einem Fall kann aus der vereinigten Kombination ein valides Produkt entstehen. Auf diese Weise identifizierte überdeckte Fragmentkombinationen, die pro Knoten ein Fragment enthalten, erfüllen also genau die Bedingung aus Beziehung 5.3. Sie werden als Kandidaten für die Schnittmenge abgespeichert und die kanonische SMILES Notation [110] ihres Produkts bestimmt. Auch der Kombinationsschritt wird, wie der Überdeckungsschritt, ebenfalls in einer inversen Reihenfolge durchgeführt. Eine Fragmentkombination wird im inversen Schritt nur dann als Kandidat abgespeichert, wenn die kanonische SMILES Notation des Produkts bereits für einen Kandidaten des initialen Kombinationsschritts abgespeichert wurde. Somit ergibt die Kandidatenmenge des inversen Kombinationsschritts die exakte Schnittmenge der zwei topologischen Fragmenträume.

5.3.4. Validierung und Evaluation

SpaceCompare wird mithilfe enumerierbare topologische Fragmenträume validiert, die Teilräume prominenter kombinatorischer Bibliotheken darstellen. Die von SpaceCompare berechneten Schnittmengen dieser Teilräume werden mit der Ausgabe von in Kapitel 5.1 beschriebenen Verfahren für die enumerierte Menge ihrer finalen Moleküle verglichen. Insgesamt neun paarweise Schnittmengen werden mithilfe der kanonischen SMILES Notation und der ECFP Fingerabdruck-Methode [98] berechnet. Das genaue Vorgehen ist in [D3] beschrieben. SpaceCompare generiert die gleiche Ausgabe wie die beiden auf Enumeration basierenden Ansätze für alle neun Schnittmengen.

Zunächst wird den in Tabelle 5.1 angegebenen Ressourcenaufwand von SpaceCompare besprochen. Die Experimente wurden mit openSUSE Leap 15 auf einer Intel Xeon

Tabelle 5.1.: Die von SpaceCompare benötigte Laufzeit in Stunden und Speicher in Gigabyte für die paarweisen Schnittmengen des REAL Space [46], GalaXi [82] und KnowledgeSpace [84]. Der Laufzeitbedarf ist aufgeteilt in den Überdeckungs- und Kombinationsschritt der Methode. Aus [D3] entnommen, angepasst und ins Deutsche übersetzt.

Schnittmenge	Laufzeiten		Speicherbedarf
	Überdeckungsschritt	Kombinationsschritt	
REAL vs Knowledge	18.735 h	44.084 h	181.8 GB
REAL vs GalaXi	2.394 h	20.200 h	57.4 GB
GalaXi vs Knowledge	0.426 h	0.020 h	17.3 GB

E5-4620 64-Bit-Architektur mit 2,2 GHz und 250 GB Arbeitsspeicher durchgeführt. SpaceCompare verwendete zur Berechnung 12 parallele Prozesse. Der KnowledgeSpace mit 10^{14} Produkten ist der größte der drei Räume. Der REAL Space mit 10^{10} Produkten übersteigt die Größe des GalaXi mit 10^{9} Produkten um grob eine Größenordnung. Die von SpaceCompare benötigten Ressourcen folgen dieser Reihenfolge nicht. Zwar weist die Schnittmengenberechnung des REAL Space und KnowledgeSpace den größten Bedarf an Speicher und Laufzeit auf, allerdings ist die Berechnung des Schnitts von KnowledgeSpace und GalaXi bei weitem die schnellste und speicherärmste. Dies zeigt, dass der Ressourcenbedarf von SpaceCompare nicht direkt mit der Anzahl implizit beschriebener Produkte der verwendeten topologischen Fragmenträume skaliert. Für die zwei Berechnungen auf dem REAL Space benötigt SpaceCompare weniger Laufzeit für den Überdeckungs- als den Kombinationsschritt. Der Überdeckungsschritt beinhaltet die Enumeration kreuzender Substrukturen. Dies weist darauf hin, dass die Enumeration kreuzender Substrukturen bei den hier durchgeführten Experimenten nicht der Enumeration aller Produkte gleichkommt.

In Abbildung 5.2 ist die Anzahl der möglichen verbleibenden Kandidaten für die Schnittmenge der drei Berechnungen gezeigt. Mithilfe dieser Werte lässt sich das Laufzeitverhalten von SpaceCompare grob erläutern. Für die Schnittmengenberechnung von REAL Space und KnowledgeSpace werden im Überdeckungsschritt so viele Fragmente des KnowledgeSpace nicht überdeckt, dass dessen Kandidatenmenge um vier Größenordnungen reduziert wird. Dadurch wird die Enumeration der 10^{14} implizit beschriebenen Moleküle vermieden. Allerdings ist die verbleibende Kandidatenmenge mit ungefähr 10^{10} Molekülen die größte der drei Berechnungen. Bei der Schnittberechnung von GalaXi und KnowledgeSpace ist die verbleibende Kandidatenmenge nach dem Überdeckungsschritt am kleinsten. Dies spiegelt das relative Laufzeitverhalten sowie den Speicherbedarf von

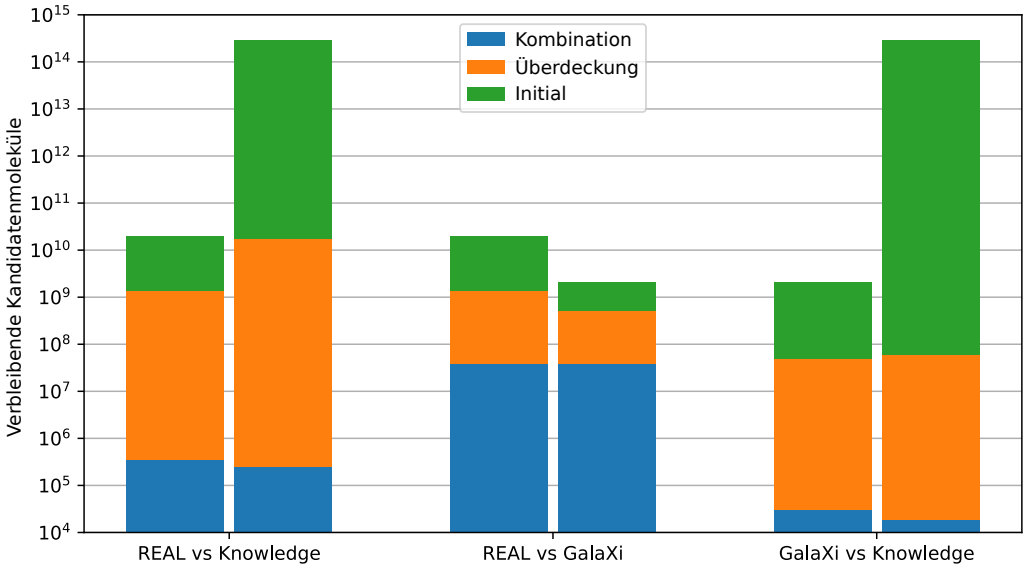


Abbildung 5.2.: Anzahl der möglichen Kandidatenmoleküle der Schnittmenge für die drei paarweisen Schnittmengen des REAL Space [46], GalaXi [82] und KnowledgeSpace [84] unterteilt nach den algorithmischen Schritten von SpaceCompare. Die gesamten Balken geben die Anzahl der Produkte in der jeweiligen kombinatorischen Bibliothek an. Der blaue und orangene Balken zusammen ergeben die Anzahl der Kombinationen aus überdeckten Fragmenten nach dem Überdeckungsschritt. Der blaue Balken gibt die Anzahl der Kandidaten nach dem Kombinationsschritt an. Auf der y-Achse werden logarithmische Einheiten verwendet. Aus [D3] entnommen, angepasst und ins Deutsche übersetzt.

SpaceCompare für die drei Rechnungen wider. Insgesamt ist die Reduktion der Kandidatenmenge im Überdeckungsschritt für die Schnittmenge von REAL Space und GalaXi am geringsten, was auf die relative Größe ihrer Schnittmenge zurückzuführen ist. Allgemein gibt die Höhe des kleineren blaue Balkens die Größe der paarweisen Schnittmenge an, da die Kandidatenmenge des inversen Kombinationsschritts bereits die berechnete Schnittmenge ist. Die Kandidatenmenge des initialen Kombinationsschritt ergibt die Menge an Produkten, deren zugrundeliegende Fragmentkombination überdeckt werden konnte. Für alle drei Berechnungen ist der Höhenunterschied der zwei blauen Balken relativ klein. Dies zeigt, dass unsere Definition von Überdeckung mithilfe chemischer Substrukturen bereits eine gute Approximation an die exakte Schnittmenge zweier kombinatorischer Bibliotheken darstellt. Damit ist die Überdeckung von Fragmentkombination ein starkes Indiz, jedoch nicht in jedem Fall ein hinreichendes Kriterium für die Identität der abgeleiteten Produkte.

5.3.5. Grenzen des Verfahrens

Die Abbildung 5.1 zeigt, dass der Ressourcenbedarf von SpaceCompare stark von den zwei topologischen Fragmenträumen der Eingabe abhängt. In diesem Kapitel werden die Grenzen der Anwendbarkeit der Methode besprochen. Diese Begrenzungen lassen sich in zwei Kategorien unterteilen.

Die erste Kategorie von Begrenzungen ergeben sich aus der Problemstellung selbst und sind unabhängig von dem algorithmischen Ansatz, den SpaceCompare verfolgt. Die Schnittmenge zweier nicht enumerierbarer kombinatorischer Bibliotheken kann selbst zu groß zur Enumeration sein. Zusätzlich muss sie nicht unbedingt eine zugrundeliegende kombinatorische Struktur aufweisen, die die Repräsentation durch eine enumerierbare Menge von chemischen Bausteinen und Reaktionen erlaubt. Also kann die Schnittmenge weder in enumerierter, noch in kombinatorischer Weise ausgegeben werden. In diesem Fall ist das Problem der Schnittmengenberechnung, unabhängig von der gewählten algorithmischen Methode, nicht lösbar.

Die zweite Kategorie von Begrenzung ergibt sich aus der algorithmischen Methode SpaceCompare selbst. Zunächst kann bereits ein Fragment durch eine nicht enumerierbare und nicht kombinatorische Menge Fragmentkombinationen minimal überdeckt werden. Zusätzlich kann der durch überdeckte Fragmente aufgespannte kombinatorische Raum nicht enumerierbar sein. Zwar wird bei den rekursive Aufrufen im Kombinationsschritt auf frühestmögliche Abbruchkriterien geachtet, aber dennoch lässt sich die komplette Enumeration nicht zwingend vermeiden. Deshalb skaliert die asymptotische Laufzeit des Verfahrens auch mit der Anzahl der Produkte der zwei betrachteten kombinatorischen Bibliotheken. Das Resultat des Überdeckungsschritts stellt nach Meinung des Autors sowohl die größte Leistung, als auch die größte Limitierung für SpaceCompare dar. In Abbildung 5.2 ist die potenziell große Diskrepanz zwischen der kompletten Anzahl von Produkten, der Kandidatenmenge nach dem Überdeckungsschritt und der tatsächlichen Schnittmenge zu sehen. Der Überdeckungsschritt reduziert die Kandidatenmenge für den KnowledgeSpace in der Berechnung mit dem REAL Space um vier Größenordnung und zeigt damit die Stärke der SpaceCompare Methode. Allerdings beträgt auch der Unterschied zur tatsächlichen Größe der Schnittmenge grob vier Größenordnungen. Damit ist das Ergebnis nach dem Überdeckungsschritt keine gute Approximation an die tatsächliche Schnittmenge. Wäre die Kandidatenmenge nach dem Überdeckungsschritt um eine Größenordnung höher, dann würde die Speicherkapazität der verwendeten Maschine eventuell schon nicht mehr für die Berechnung ausreichen.

5.4. Schnittmenge prominenter kombinatorischer Bibliotheken

In diesem Abschnitt wird SpaceCompare angewendet, um die paarweise und dreifache Schnittmenge der drei kombinatorischen Bibliotheken REAL Space [81], GalaXi [82] und CHEMriya [83] zu untersuchen. Die Bibliotheken wurden in Kapitel 2.2 beschrieben und weisen die Besonderheit auf, dass ihre Produkte von den drei Chemiekonzernen bezogen werden können.

In Abbildung 5.3 (a) ist die Schnittmengen der drei Bibliotheken gezeigt. Die größte paarweise Schnittmenge weisen der REAL Space und GalaXi mit 38 Millionen gemeinsamen Molekülen auf. In absoluten Zahlen besitzt der REAL Space die größten Schnittmengen mit den anderen beiden Räumen. Da er aber auch die größte Anzahl implizit beschriebener Produkte enthält, weist GalaXi die größte relative Schnittmenge auf. Dennoch beträgt auch diese weniger als 2% seiner gesamten Produktmenge. Zu jedem Produkt einer kombinatorischen Bibliothek lässt sich ein Syntheseweg durch die enkodierten Repräsentationen chemischer Bausteine und Reaktionen bestimmen. Wie in Kapitel 5.1 beschrieben, lassen sich die hinterlegten Informationen zweier virtueller chemischer Bibliotheken für ihre Schnittmenge verknüpfen. In hier vorliegenden Fall bedeutet dies, dass die Synthesewege der drei Chemiekonzerne verglichen werden können. Aus Abbildung 5.3 (b) lässt sich schließen, dass mehr als zwei Drittel der dreifachen Schnittmenge der Bibliotheken wiederum in der dreifachen Schnittmenge der gezeigten Teilräume enthalten ist. Diese beschreiben alle eine Amid-Kupplung, weshalb die drei Konzerne für einen Großteil ihrer dreifachen Schnittmenge den gleichen Reaktionstyp verwenden. Für 55.728 dieser Moleküle nutzen sie ebenfalls die gleichen chemischen Bausteine, weshalb die gewählten Synthesewege in diesen Fällen übereinstimmen. Das rechte obere Molekül in Abbildung 5.3 (b) ist ein Beispiel dafür. Die verbleibende 214 Moleküle in der dreifachen Schnittmenge der Teilräume werden zwar alle mithilfe einer Amid-Kupplung synthetisiert, allerdings mithilfe unterschiedlicher chemischer Bausteine. Wie im unteren rechten Beispiel in (b) zu sehen, bildet der Syntheseweg in GalaXi eine andere Amidbindung, als im REAL Space und CHEMriya. 4.672 Moleküle sind in keinem der Teilräume, aber in der dreifachen Schnittmenge der gesamten Bibliotheken enthalten. Im Beispielmolekül links in (b) bildet CHEMriya während der Synthese das Chinolin und REAL Space und GalaXi formen die Bindung am Amin.

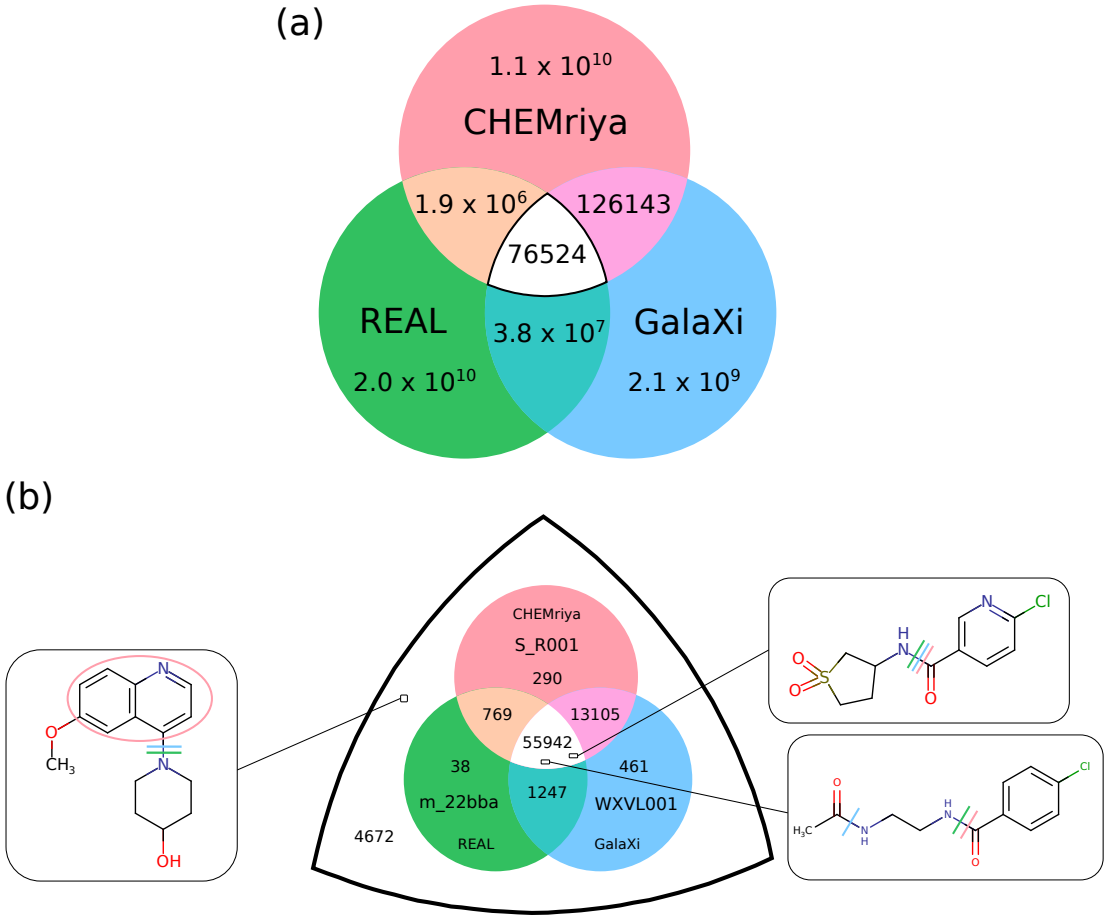


Abbildung 5.3.: (a) Venn-Diagramm der paarweisen und dreifachen Schnittmengen des REAL Space in grün, CHEMriya in magenta und GalaXi in türkis. (b) Analyse der dreifachen Schnittmenge der drei Bibliotheken. Die schwarz umrandete Fläche beschreibt die gesamte dreifache Schnittmenge. Das Venn-Diagramm innerhalb der Fläche zeigt die Schnittmenge der topologischen Fragmentteilräume m_22bba des REAL Space, S_R001 des CHEMriya und WXVL001 des GalaXi. Der Bereich innerhalb der Fläche, aber außerhalb des Venn-Diagramms beschreibt Moleküle des dreifachen Schnitts, die in keiner der drei Teilräume enthalten sind, links in (b) ist eines dieser Moleküle abgebildet. Die beiden Moleküle rechts sind in der dreifachen Schnittmenge der Teilräume enthalten. Die farbigen Striche markieren die Bindung, die im farblich dazu passenden Raum bei der Reaktion gebildet wird. Der farbige Kreis am linken Molekül markiert das Ringsystem, dass im CHEMriya bei der Reaktion gebildet wird. Aus [D3] entnommen und angepasst.

5.5. Ausblick

Mit der algorithmischen Methode SpaceCompare ist es erstmals möglich die exakte Schnittmenge kombinatorischer Bibliotheken zu bestimmen, ohne zwingend alle ihre Produkte zu enumerieren. SpaceCompare wurde modular in die Codebasis NAOMI [54] integriert und lässt sich somit anpassen und erweitern. Um die verbleibende Kandidatenmenge nach dem Überdeckungsschritt zu reduzieren, könnte mithilfe einer MCS-Strategie [36] überprüft werden, ob das überdeckte Fragment eine Substruktur der überdeckenden Fragmentkombination darstellt. Ist dies nicht der Fall, dann könnte die Überdeckung verworfen werden. Die Berechnung anderer Resultate ist auch umsetzbar mit dem bestehenden algorithmischen Ansatz. Beispielweise könnte ein topologischer Teilraum aus allen nicht überdeckten Fragmenten erzeugt werden. Werden nur diese Fragmente eines topologischen Fragmentraums gewählt, dann stellen die Produkte dieses kombinatorischen Teilraums eine Teilmenge des Komplements mit dem anderen Raum dar. Werden die Fragmente beider Räume gewählt, ergibt sich eine kombinatorische Teilmenge der symmetrischen Differenz der Produkte beider Räume. Für diese Berechnungen wird nur der Überdeckungsschritt ohne dessen Datenstruktur der erzeugten Überdeckungen benötigt. Dies würde den Ressourcenaufwand der Methode wahrscheinlich drastisch senken und die Laufzeit dieses Ansatzes skaliert asymptotisch nicht mehr mit der Enumeration der gesamten Produktmenge der Bibliothek.

Mithilfe von SpaceCompare wurde gezeigt, dass die Schnittmenge prominenter Bibliotheken, trotz ihrer Größe, prozentual gering ist. Für die Moleküle der Schnittmenge konnten die in den Bibliotheken enkodierten Synthesewege verglichen werden. Falls andere Informationen in den kombinatorischen Bibliotheken hinterlegt wären, ließen sich diese ebenfalls verknüpfen. Allerdings müssten Informationen zu Reaktionen und chemischen Bausteinen abgespeichert werden, da Produkte nicht unbedingt enumeriert werden können. Beispielweise könnten Preise der chemischen Bausteine und Reaktionen oder die prognostizierte Ausbeute eines chemischen Bausteins innerhalb einer Reaktion hinterlegt werden. Auf diese Weise könnten die Preise oder Ausbeuten unterschiedlicher Synthesewege für die Molekül in der Schnittmenge verglichen werden. Für die chemischen Bausteine könnten auch Experimentaldaten aus dem Fragment-basierten Wirkstoffentwurf [144] abgespeichert und verknüpft werden.

6. Physikochemische Eigenschaftsverteilungen kombinatorischer Bibliotheken

In diesem Kapitel werden Verteilungen von Eigenschaften der Moleküle einer physischen oder virtuellen chemischen Bibliothek behandelt. Wie bei der Bestimmung von Schnittmengen aus dem vorangegangen Kapitel 5, handelt es sich also wiederum um allgemeine Charakteristiken chemischer Bibliotheken. Die generierten Verteilungen können verwendet werden, um die Eignung von Bibliotheken generell oder für bestimmte Projekte des Wirkstoffentwurfs zu untersuchen. Auf der Basis physikochemischer Eigenschaften wurden beispielsweise approximative Faustregeln entwickelt, die für die Abschätzung von oraler Bioverfügbarkeit [145], [146] oder die Eignung als chemischer Baustein [147] im Wirkstoffentwurf verwendet werden können.

6.1. Physikochemische Eigenschaften und existierende Berechnungsverfahren

Für enumerierte physische und virtuelle chemische Bibliotheken lassen sich die für jedes Moleküle einzeln bestimmten Eigenschaften direkt zu Verteilungen zusammenfassen. Für kombinatorische virtuelle chemische Bibliotheken gibt es bisher keine Verfahren zur Berechnung von Eigenschaftsverteilungen, die nicht auf der Enumeration aller Produkte basieren. Deshalb werden im Folgenden Arten physikochemische Eigenschaften individueller Moleküle und teilweise Verfahren zu deren approximativer Berechnung besprochen.

Das Molekulargewicht ist die Summe der Massen aller Atome eines Moleküls. [1] Als Teil der von Lipinski et al. definierten 5er-Regel (*rule of five*) [145] wird oft eine obere Schranke

von 500 Dalton für das Molekulargewicht von Kandidaten für den Wirkstoffentwurf angesetzt. Die Anzahl der Schweratome eines Moleküls wird ebenfalls oft als physikochemische Eigenschaft betrachtet, wobei 36 Schweratome ungefähr mit einem Molekulargewicht von 500 Dalton korrespondieren [148] und deshalb auch als Obergrenze, allerdings nicht in der 5er-Regel, verwendet werden. Die Anzahl der potenziellen Protonenakzeptoren und Protonendonatoren einer Wasserstoffbrückenbindung [145] beschreibt pharmakologische Eigenschaften eines Moleküls. Als Teil der 5er-Regel wird eine obere Schranke von zehn Protonenakzeptoren und fünf Protonendonatoren für Moleküle angesetzt. Der dekadische Logarithmus des Octanol-Wasser-Koeffizienten, mit log P bezeichnet, beschreibt die Löslichkeit eines Moleküls in Wasser. [149], [150] Die Wasserlöslichkeit eines Moleküls kann für seine Eignung als Kandidat im Wirkstoffentwurf entscheidend sein [145]. In der 5er-Regel wird ein Wert von fünf als obere Schranke für den log P angenommen. Insgesamt erfüllt ein Molekül die 5er-Regel, wenn es höchstens eine der vier genannten oberen Schranken verletzt. Es existieren allerdings auch zugelassene Wirkstoffe mit einer breiten Anwendung, die diese Regel verletzen. [151]–[153]

Über die Jahre wurden verschiedene Methoden entwickelt, um den log P eines Moleküls rechnerisch zu approximieren anstatt ihn experimentell im Labor bestimmen zu müssen. In [154] wurde eine Korrelation zwischen dem Wiener Index, [155] einer der frühesten topologischen Deskriptoren, und dem log P eines Moleküls gefunden. In [42] wurde eine Methode entwickelt, um den log P eines Moleküls mithilfe eines strukturell ähnlichen Moleküls mit bekanntem log P zu approximieren. Klopman et al. beschreiben in [156] eine Methode den approximativen log P eines Moleküls durch eine Auswahl seiner funktionellen Gruppen abzuleiten. Zusätzlich wird ein korrektiver Term verwendet, der die intramolekulare Interaktion zwischen funktionellen Gruppen berücksichtigt. Dieser wurde in eine späteren Publikation überarbeitet. [157] Wildman und Crippen stellten in [41] eine Berechnungsmethode vor, die eine Erweiterung älterer Methodiken [158]–[160] ist. In diesem Ansatz wird jedem Atom eines Moleküls auf der Basis seines Elements und seiner chemischen Umgebung ein Wert zugewiesen. Diese Werte werden aufsummiert und ergeben eine Approximation des log P des Moleküls. Diese approximative Rechnung ist aufgrund ihrer Effizienz weit verbreitet und konnte im Vergleich mit anderen Methoden in einem Benchmark gute Ergebnisse erzielen. [161] Diese Approximation wird in dem in diesem Kapitel vorgestellten algorithmischen Verfahren verwendet und im Folgenden mit *aLogP* bezeichnet.

Des Weiteren existieren physikochemische Eigenschaften, die die räumliche Konformation oder Flexibilität eines Moleküls beschreiben. Ein rotierbare Bindung führt zu

unterschiedlichen räumlichen Konformationen eines Moleküls, wobei unterschiedliche Definitionen für die Rotierbarkeit einer Bindung existieren. [146], [162] Die Anzahl rotierbarer Bindungen eines Moleküls beschreibt somit dessen räumliche Flexibilität und wird als Gradmesser für dessen orale Bioverfügbarkeit verwendet. [146] Der polare Oberflächenbereich (*polar surface area*(PSA)) einer räumlichen Konformation eines Moleküls beschreibt räumliche pharmakologische Eigenschaften und kann beispielsweise verwendet werden, um seine Fähigkeit abzuschätzen die Blut-Hirn-Schranke zu überwinden. [163] Der topologische polare Oberflächenbereich (*topological polar surface area*(TPSA)) [164] approximiert den PSA-Wert einer Konformation auf Basis funktioneller Gruppen, vermeidet die Betrachtung räumlicher Konformationen und steigert somit die Effizienz der Berechnung. Eine weitere Eigenschaft der räumlichen Konformation von Molekülen ist die Beschreibung ihrer Form. Dafür werden die Verhältnisse zwischen den Trägheitsmomenten bezügliche ihrer drei Hauptrotationsachsen bestimmt und in einem zweidimensionalen Vektor dargestellt. Mithilfe dieser Vektoren kann die Form eines Moleküls dann grob als kugel-, scheiben- oder stabförmig kategorisiert werden. [165]

6.2. Herausforderungen bei der Berechnung von Eigenschaftsverteilungen kombinatorischer Bibliotheken

Für die in Kapitel 5 beschriebene Schnittmengenberechnung, wurde in [142] ein Stichprobenverfahren entwickelt, um die exakte Schnittmenge abzuschätzen. Ein ähnlicher Ansatz könnte auch für Eigenschaftsverteilungen verwendet werden, indem die Eigenschaften für eine enumerierbare Teilmenge von Produkten bestimmt und zusammengefasst werden. Allerdings stellt die so erzeugte Verteilung wiederum nur eine Schätzung dar, die nicht zwangsläufig eine gute Approximation an die exakte Verteilung aller Produkte einer kombinatorischen Bibliothek sein muss. Wie für die Problemstellungen aus den Kapiteln 4 und 5 muss auch in diesem Kontext eine exakte Methode das algorithmische kombinatorische Paradigma erfüllen. Das Verfahren muss also die komplette Enumeration aller implizit beschriebenen Produkte vermeiden und die einzelne Bestimmung der Eigenschaften aller Produkte ist nicht anwendbar. Stattdessen sollte die Methode auf den Repräsentationen chemischer Bausteine und Reaktionen operieren.

Eine erste Idee kann es sein, die Eigenschaftswerte der einzelnen chemischen Bausteine zu bestimmen und alle Werte von Bausteinen zusammenzufassen, die innerhalb einer Reaktion an der gleichen Stelle verwendet werden können. So könnte die Eigenschaftsverteilung

der Produkte aus der Kombination der Verteilungen der Bausteine bestimmt werden. Die Eigenschaftswerte der Verteilungen werden addiert und die Anzahl der Vorkommnisse der Werte multipliziert. Durch dieses Zusammenfassen chemischer Bausteine wird die Enumeration der Bibliothek verhindert. Allerdings setzt dieses Verfahren voraus, dass sich der Eigenschaftswert eines Produkts in jedem Fall durch die Summe der Eigenschaftswerte seiner chemischen Bausteine ausdrücken lässt.

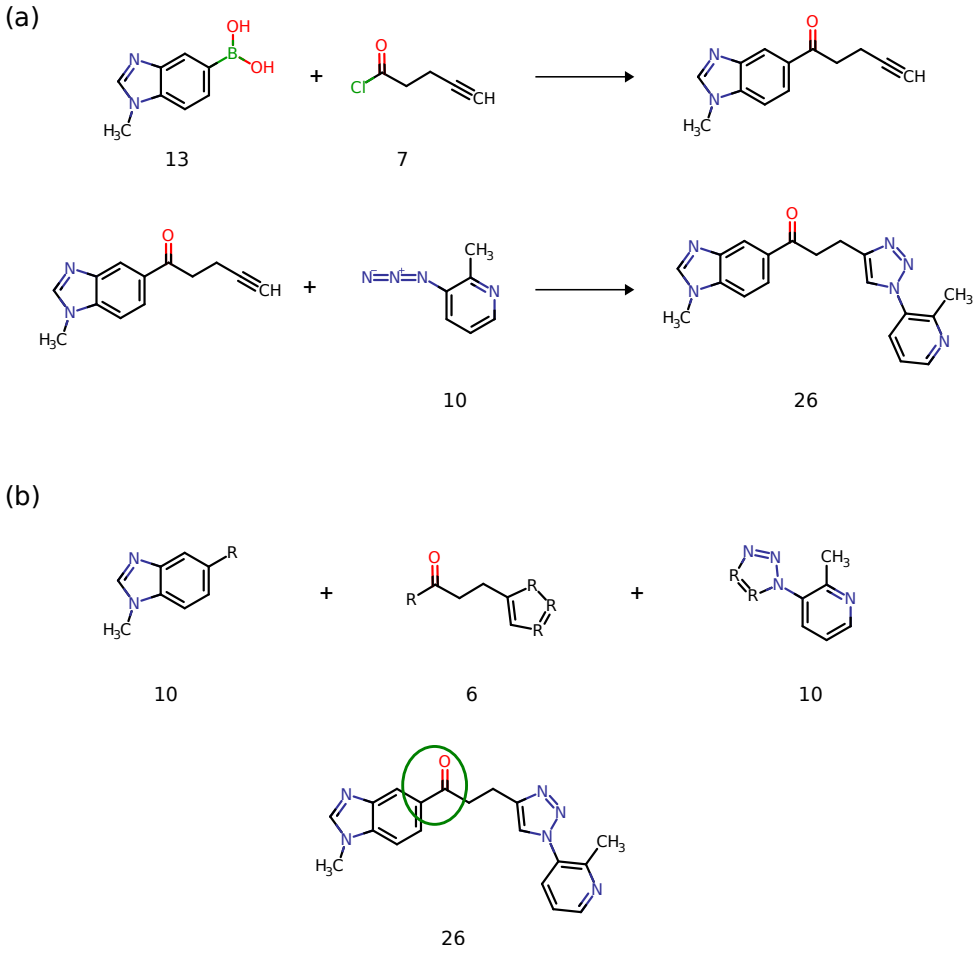


Abbildung 6.1.: (a) Ein Syntheseweg mit drei chemischen Bausteinen und zwei konsekutiven Reaktionen. (b) Die Fragmentrepräsentationen der chemischen Bausteine des Synthesewegs aus (a) zusammen mit dem Produkt. Die im grün markierten Bereich enthaltene chemische Substruktur bestimmt den aLogP-Atomwert des Sauerstoffatoms im Produkt. Für alle chemischen Bausteine, Fragmentrepräsentationen und das Produkt ist die Anzahl enthaltener Schweratome angegeben. Aus [D4] entnommen und angepasst.

Dies ist aber nicht immer der Fall wie in Abbildung 6.1 zu sehen ist. Die Summe der in den drei chemischen Bausteinen enthaltenen Schweratome unterscheidet sich von der Anzahl im Produkte enthaltenen Schweratome. Um den Ansatz anzupassen kann, anstatt auf chemischen Bausteinen, auf den Fragmenten eines topologischen Fragmentraums operiert werden. Wie in (b) zu sehen ist, stimmt die Summe der in den Fragmenten enthaltenen Schweratome nun mit der des Produkts überein. Aus der in Kapitel 2.3.2 beschriebenen Substruktur-Teilmengenrelation folgt direkt, dass diese Übereinstimmung für alle Produkte und ihre Fragmentkombinationen eines topologischen Fragmentraums gilt.

Allgemein wird eine physikochemische Eigenschaft als *additiv* bezeichnet, wenn für ein beliebiges Produkt eines topologischen Fragmentraums dessen Eigenschaftswert mit der Summe von Eigenschaftswerten seiner Fragmente übereinstimmt. Falls alle betrachteten physikochemischen Eigenschaften additiv sind, existiert somit bereits eine Strategie zur exakten Berechnung von Eigenschaftsverteilungen ohne komplette Enumeration eines topologischen Fragmentraums.

Strategie für additive Eigenschaften

1. Bestimme die Eigenschaftswerte aller Fragmente eines topologischen Fragmentsraums
2. Fasse die Eigenschaftswerte aller Fragmente eines Topologieknotens in einer Verteilung zusammen
3. *Multipliziere* die Verteilungen der Knoten eines Topologiegraphen, indem alle Kombinationen von Werten addiert und ihre Häufigkeiten multipliziert werden. Es entsteht eine Eigenschaftsverteilung pro Topologiegraph
4. *Addiere* die Verteilungen aller Topologiegraphen, indem die Werte in einer gesamten Verteilung zusammengefasst werden. Kommt ein Eigenschaftswert in mehreren Verteilungen vor, werden die Häufigkeiten addiert

Allerdings handelt es sich beim aLogP um eine nicht additive Eigenschaft. Jedem Atom wird ein Atomwert zugewiesen, der sich aus den Eigenschaften des Atoms selbst, aber auch aus seiner chemischen Umgebung ableitet. Für das Sauerstoffatom im Produkt aus Abbildung 6.1 ergibt sich dessen Atomwert aus der es enthaltenden Carbonylgruppe mit benachbartem aromatischen Kohlenstoff. Die korrespondierende, in (b) markierte Substruktur ist allerdings in keinem der Fragmente komplett enthalten. Dem Sauerstoffatom würde im Fragment selbst entweder kein oder der falsche Wert zugewiesen werden. Insgesamt kann der aLogP-Wert eines einzelnen Fragments nicht immer bestimmt werden, ohne dessen Umgebung innerhalb der Produkte zu betrachten. Auf diesen Umstand, also die Behandlung nicht additiver physikochemischer Eigenschaften, wird in der vorgestellten Methode eingegangen.

6.3. Das algorithmische Verfahren SpaceProp

In diesem Kapitel wird die algorithmische Methode SpaceProp besprochen, wie sie in [D4] vorgestellt wurde. Mit SpaceProp ist es erstmals möglich exakte Verteilungen physikochemischer Eigenschaften aller Produkte einer kombinatorischen Bibliothek zu berechnen. Damit grenzt sich das Verfahren voll allen existierenden Methoden stark ab. In seiner Implementation in der Codebasis NAOMI [54] ist SpaceProp in der Lage Verteilungen für die folgenden fünf Eigenschaften zu berechnen: aLogP, Molekulargewicht und Anzahl der enthaltenen potenziellen Protonenakzeptoren und Protonendonatoren einer Wasserstoffbrückenbindung sowie der Anzahl der enthaltenen Schweratome. Das

methodische Vorgehen allgemein beschrieben und lässt sich grundsätzlich auch auf andere Eigenschaften anwenden. Auf diesen Umstand wird in Kapitel 6.6 eingegangen.

6.3.1. Interne und externe Eigenschaftskomponente und Randinformation

Um nicht-additive physikochemische Eigenschaften ohne vollständige Enumeration behandeln zu können werden die Fragmente einer Kombination und das korrespondiere Produkt in einen unabhängigen und eine abhängigen Bereich unterteilt. für den unabhängigen Bereich lässt sich der Beitrag zum gesamten Eigenschaftswert allein mithilfe von Informationen bezüglich eines einzelnen Fragments bestimmen. Für den abhängigen Teil ist dies nicht der Fall. Alle fünf hier behandelten Eigenschaften basieren auf der Summe von Werten einzelner Atome. Deshalb sind die nun folgenden Definitionen ebenfalls mithilfe von Atomen formuliert. Sie könnten theoretisch allerdings auch auf Basis chemischer Bindungen oder Substrukturen definiert werden.

Gegeben sein ein Produkt eines topologischen Fragmentraums und dessen zugrundeliegende Fragmentkombination. Ein Atom wird als *intern* bezeichnet, wenn sein korrekter Atomwert im Produkt bereits aus dem einzelnen es enthaltenden Fragment ableitbar ist. Ansonsten wird das Atom als *extern* bezeichnet. Die *interne Eigenschaftskomponente* eines Fragments ist die Summe der Eigenschaftswerte aller internen Atome des Fragments. Die *externe Eigenschaftskomponente* ist die Summe der Atomwerte externer Atome im Produkt. Somit ergibt sich der Eigenschaftswert des Produkts aus der Summe der internen Eigenschaftskomponenten der Fragmente und der externen Eigenschaftskomponente des Produkts. Ist die externe Eigenschaftskomponente für alle Produkte aller topologischen Fragmenträume gleich null, ist dies äquivalent zur Additivität der betrachteten Eigenschaft. Für den aLogP und die Fragmentkombination in Abbildung 6.1 (b) sind das Sauerstoffatom und das aromatische Kohlenstoffatom im grün markierten Bereich extern und alle anderen Atome sind intern. Für die Anzahl enthaltener Schweratome sind alle Atome intern, da diese Eigenschaft additiv ist.

Die *Randinformation* eines Fragments besteht aus allen seinen externen Atomen, sowie den Informationen die benötigt werden, um die Atomwerte externer Atome des Fragments selbst und anderer Fragmente zu bestimmen. Diese etwas wage Definition hängt stark von der konkret betrachteten Eigenschaft ab. Die Randinformation wird in der Regel Aspekte der chemischen Umgebung von Linkern beschreiben. In Abbildung 6.2 ist die Randinformation des zweiten Fragments aus Abbildung 6.1 (b) für den aLogP gezeigt. Alle Schweratome, die zu einem Linker benachbart sind, sind in der Randinformation enthalten.

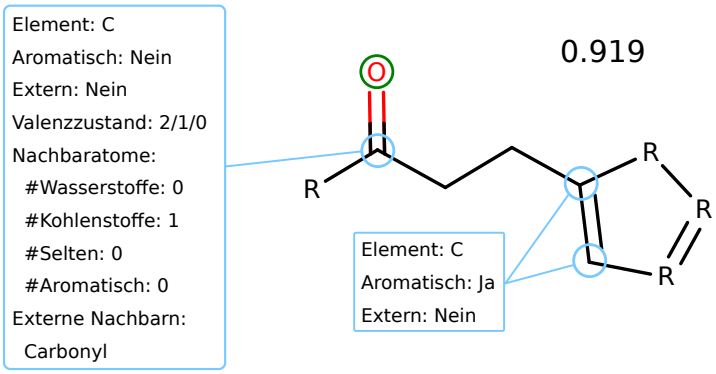


Abbildung 6.2.: Die Randinformation des zweiten Fragments aus Abbildung 6.1 (b). Oben rechts ist die interne aLogP-Komponente angegeben. Das externe Sauerstoffatom und alle blau markierten Atome, sowie die blau eingerahmten Angaben bilden die Randinformation. Der Valenzzustand repräsentiert die zwei Einfach- und eine Doppelbindung des markierten Kohlenstoffatoms. Für die Nachbaratome des markierten Kohlenstoffatoms werden jeweils die Anzahl Wasserstoff- und Kohlenstoffatome, sowie die Anzahl aromatischer und 'seltener' Atome gezählt. Ein Atom wird als 'selten' deklariert, wenn es kein Kohlenstoff-, Sauerstoff-, Stickstoff-,Wasserstoff- oder Phosphoratom und kein Halogen ist. Aus [D4] entnommen, ins Deutsche übersetzt und angepasst.

Die Atomwerte des aLogP sind allgemein nur von den direkten Nachbarn und im Falle von Wasserstoff- und terminalen Sauerstoffatomen teilweise von Atomen mit Distanz zwei abhängig. Deshalb sind alle externen Atome entweder selbst Nachbarn von Linkern, oder haben zu ihnen eine Distanz von zwei. Somit lässt sich die gesamte Randinformation als Information der direkten Nachbarn von Linkern und deren Nachbarschaft ausdrücken. Für jeden internen Linknachbarn wird dessen Element und Aromatizität abgespeichert. Diese Information wird später benötigt, um die Atomwerte der externen Atome aus anderen Fragmenten zu bestimmen. Für Linknachbarn, die selbst extern sind oder die einen externen Nachbarn haben, wird zusätzlich ihr Valenzzustand, die Anzahl ihrer benachbarten Wasserstoff- und Kohlenstoffatome sowie die Anzahl der aromatischen und seltenen Nachbarn hinterlegt. Zusätzlich wird noch die Art externer Nachbarn abgespeichert, in diesem Fall ein Sauerstoffatom einer Carbonylgruppe. Mithilfe dieser Information kann später der Atomwert dieses Sauerstoffatoms bestimmt werden.

6.3.2. Verteilungsberechnung für nicht additive Eigenschaften

Für additive Eigenschaften wurde bereits in Kapitel 6.2 ein Verfahren entwickelt. Das Vorgehen von SpaceProp für nicht additive Eigenschaften ist dazu verwandt. Allerdings können die Werte aller Fragmente eines Topologieknotens nicht unbedingt in einer Verteilung zusammengefasst werden. Stattdessen wird eine Eigenschaft der Randbedingung ausgenutzt. Verschiedene Fragmentkombinationen, die die gleiche kombinierte Randinformation aufweisen, haben auch die gleiche externe Eigenschaftskomponente. Das bedeutet, der Unterschied in den Eigenschaftswerten ihrer Produkte ergibt sich ausschließlich aus den internen Eigenschaftskomponenten der einzelnen Fragmente. Deshalb können die internen Eigenschaftskomponenten von Fragmenten aus einem Topologieknoten zu einer Verteilung zusammengefasst werden, wenn sie die gleiche Randinformation aufweisen.

In Abbildung 6.3 ist eine Kombination aus Randinformationen des aLogP und dazugehörige Fragmente angegeben. Die externe aLogP Komponente ist die gleiche für alle zwölf Produkte, die aus den Fragmenten entstehen. Um die externe aLogP Komponente zu bestimmen, werden die Randinformationen von Linknachbarn aufgelöst, die entweder selbst extern sind oder einen externen Nachbarn haben. Dafür wird die Randinformationen der anderen Fragmente verwendet. Beispielsweise ergibt sich für das externe Kohlenstoffatom in der türkis markierten Gruppe von Fragmenten ein weiterer impliziter Kohlenstoffnachbar in den Fragmenten der magenta markierten Gruppe. Daraus leitet sich sein Atomtyp C21 nach der Definition von Wildman und Crippen eindeutig ab. [41] Ähnlich wird mit dem Linknachbarn der Fragmente der magenta markierten Gruppe verfahren, der ein externes Sauerstoffatom als Nachbarn besitzt. Da der Linknachbar nun implizit zu einem aromatischen Kohlenstoffatom benachbart ist, leitet sich der Atomtyp O10 für das externe Sauerstoffatom ab.

Insgesamt verfährt das algorithmische Verfahren SpaceProp für nicht additive physikochemische Eigenschaften wie folgt, wobei die Definitionen von Addition und Multiplikation von Verteilungen aus der Strategie für additive Eigenschaften aus Kapitel 6.2 übernommen wurde.

Strategie für nicht-additive Eigenschaften

1. Bestimme die Randinformationen und internen Eigenschaftskomponenten aller Fragmente eines topologischen Fragmentsraums

2. Fasse die internen Eigenschaftskomponenten aller Fragmente eines Topologieknotens mit der gleichen Randinformation in einer Verteilung zusammen

3. Bilde für einen Topologiegraphen alle Kombinationen von Verteilungen, sodass genau eine Verteilung pro Topologieknoten gewählt wird. Löse die Randinformationen der Kombination auf und bestimme die externe Eigenschaftskomponente. Multipliziere alle Verteilungen der Kombination und addiere die externe Eigenschaftskomponente auf alle Werte.

4. Addiere alle diese Verteilungen, um die gesamte Eigenschaftsverteilung der Produkte des Topologiegraphen zu erhalten.

5. Addiere alle Verteilungen aller Topologiegraphen für die Eigenschaftsverteilung aller implizit beschriebenen Moleküle des topologischen Fragmentraums

Für die Ausgabe werden die Molekulargewichte der finalen Verteilung ganzzahlig und die Werte der aLogP-Verteilung auf eine Nachkommastelle gerundet. Alle weiteren Eigenschaftswerte sind ganzzahlig und bleiben unverändert.

6.3.3. Approximativer Ansatz

Das Verfahren SpaceProp ist in der Lage Fragmente mit der gleichen internen Eigenschaftskomponente und der gleichen Randinformation als äquivalent zu behandeln und somit die vollständige Enumeration eines topologischen Fragmentraums möglichst zu verhindern. Wie viele Fragmente tatsächlich diese Gleichheit erfüllen, hängt allerdings stark von der betrachteten physikochemischen Eigenschaft ab. Die Anzahl der potenziellen Protonenakzeptoren und Protonendonatoren einer Wasserstoffbrückenbindung sind ganzzahlige Werte die realistischerweise zwischen null und 40 liegen für im Wirkstoffentwurf verwendete Moleküle. Somit ist der Wertebereich interner Eigenschaftskomponenten für diese beiden Eigenschaften begrenzt und es können viele Fragmente von SpaceProp als äquivalent betrachtet und gleichzeitig verarbeitet werden. Im Gegensatz dazu sind das Molekulargewicht und der aLogP eines Moleküls Dezimalzahlen mit bis zu vier bzw. fünf Nachkommastellen. Also ist die potenzielle Bandbreite interner Eigenschaftskomponenten viel größer und es können weniger oder möglicherweise gar keine Fragmente als äquivalent betrachtet werden. Deshalb skaliert die asymptotische Laufzeit der exakten Verteilungsberechnung durch SpaceProp mit der Anzahl der Produkte einer kombinatorischen Bibliothek.

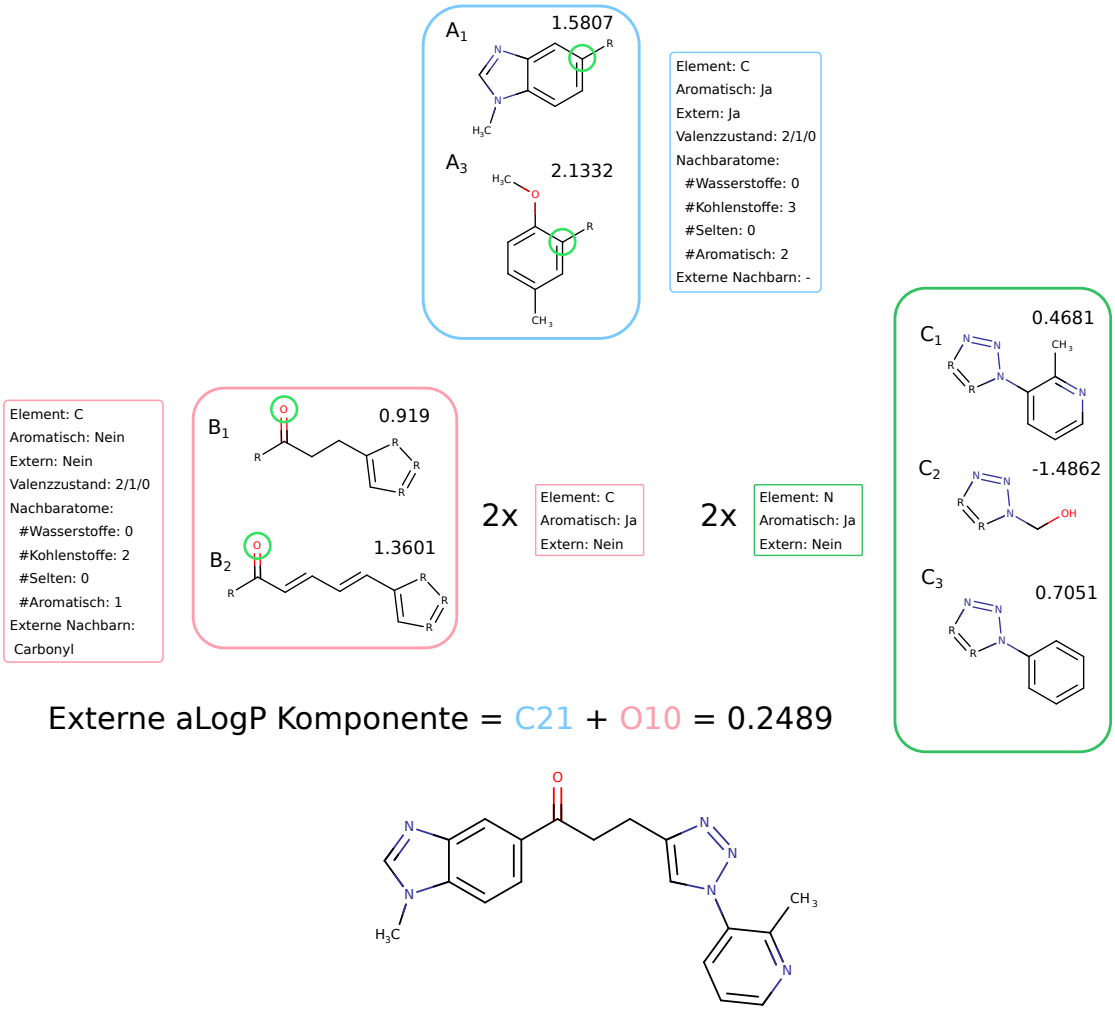


Abbildung 6.3.: Fragmente mit gleicher Randinformation für den aLogP. Die farblichen Rahmen geben die Gruppierung von Fragmenten mit gleicher Randinformationen an. Die Randinformationen wurden aufgelöst mit den Informationen der anderen Fragmente und ihre Notation entspricht der in Abbildung 6.2. Die grün umrandeten Atome sind extern. Ihre abgeleiteten Atomwerte haben die gleiche Farbe, wie die Rahmen der gruppierten Fragmente und ergeben die externe aLogP Komponente. Es ist das Produkt angezeigt, dass sich aus den ersten Fragmenten der Gruppen ergibt. Sein aLogP Wert ist angegeben und die internen aLogP Komponenten der Fragmente haben die gleiche Farbe wie die Rahmen ihre Gruppe. Aus [D4] entnommen, ins Deutsche übersetzt und angepasst.

Um den Anteil äquivalenter Fragmente und damit die Effizienz der Methode für das Molekulargewicht und den aLogP zu erhöhen, wurde neben dem exakten Ansatz von SpaceProp zusätzlich ein approximatives Verfahren entwickelt. Hierbei werden die interne Eigenschaftskomponenten aller Fragmente für das Molekulargewicht auf zwei Nachkommastellen und für den aLogP auf drei Nachkommastellen gerundet. In allen anderen Aspekten entspricht die approximative Variante von SpaceProp dem in Kapitel 6.3.2 beschriebenen exakten Verfahren. Durch das Runden kann das berechnete approximative Molekulargewicht für eine Fragmentkombination mit x Fragmenten einen Fehler von $0,005x$ enthalten. Für den aLogP beträgt dieser Fehler $0,0005x$. Da ein Produkt in aller Regel aus weniger als 200 Fragmenten besteht, entspricht der einzig mögliche Rundungsfehler für ein Produkt dem Rundungsgrad der Ausgabe. Damit beträgt dieser Fehler für das Molekulargewicht eins und für den aLogP 0,1. Die Auswirkungen des approximativen Ansatzes sowohl für Präzision und Laufzeitverhalten werden im folgenden Kapitel analysiert.

6.3.4. Validierung und Evaluation

Zur Validierung von SpaceProp wird nach einem ähnlichen Ansatz wie in Kapitel 5.3.4 verfahren. Zunächst werden jeweils zwei enumerierbare topologische Teilfragmenträume des REAL Space [81], GalaXi [82], CHEMriya [83] und des KnowledgeSpace [84] erzeugt. Der Aufbau dieser Räume wurde in Kapitel 2.2 beschrieben. Für die insgesamt acht Teilräume werden jeweils alle Produkte enumeriert und ihr Molekulargewicht, aLogP und die Anzahl der enthaltenen potenziellen Protonenakzeptoren und Protonendonatoren einer Wasserstoffbrückenbindung sowie der enthaltenen Schweratome berechnet. Die auf diese Weise generierten Verteilungen stimmen mit den Ergebnisse der exakten SpaceProp Methode für alle insgesamt 40 Verteilungen überein. Das genaue Vorgehen ist in [D4] beschrieben.

Die Laufzeit der Methode wird mit Ubuntu 20.04 LTS auf einer i5-4690K 64-Bit-Architektur mit 3,5 GHz und 16 GB Arbeitsspeicher analysiert. Für die Berechnung wurden vier Prozesse parallel verwendet. Die Ergebnisse sind in Tabelle 6.1 gezeigt. Der KnowledgeSpace enthält die meisten Produkte und die exakte SpaceProp Variante benötigt die längste Laufzeit für die Berechnung seiner Eigenschaftsverteilungen. Andersherum enthält GalaXi die geringste Anzahl von Produkten und SpaceProp benötigt die geringste Laufzeit für beide Varianten. Allerdings übersteigt die Anzahl der Moleküle des KnowledgeSpace die von GalaXi um fünf Größenordnungen. Die Laufzeit von SpaceProp

Tabelle 6.1.: Laufzeiten der SpaceProp Methode in der exakten und approximativen Variante für den REAL Space, GalaXi, CHEMriya und KnowledgeSpace. Zusätzlich ist der Fehler in den berechneten Verteilungen der approximativen Variante für den aLogp und das Molekulargewicht (MW) angegeben. Hierfür wurden alle Abweichungen in den Häufigkeiten für die exakte und approximative Verteilung aufsummiert und durch die Anzahl in der Bibliothek enthaltenen Produkte geteilt. Diese Zahl ergibt den Prozentwert. Der Prozentwert wurde mit der Höhe des einzig möglichen Rundungsfehlers multipliziert, um den durchschnittlichen Fehler im Eigenschaftswert über alle Moleküle zu bestimmen. Diese Zahl ist in Klammern angegeben. Aus [D4] entnommen und ins Deutsche übersetzt.

Bibliothek	Exakt	Approx.	aLogP Fehler	MW Fehler
REAL Space	191 s	130 s	0,009% (9×10^{-6})	0,821% (0,00821)
GalaXi	27 s	10 s	0,004% (4×10^{-6})	0,618% (0,00618)
CHEMriya	269 s	17 s	0,003% (3×10^{-6})	0,087% (0,00087)
KnowledgeSpace	7353 s	77 s	0,016% ($1,6 \times 10^{-5}$)	0,041% (0,00041)

steigt allerdings nur um weniger als drei Größenordnungen für den KnowledgeSpace im Vergleich zu GalaXi. Dies weist darauf hin, dass auch in der exakten Variante von SpaceProp in diesen Fällen keine vollständige Enumeration stattfindet. In der approximativen Variante schrumpft der Unterschied zwischen den Laufzeiten von SpaceProp für die vier kombinatorischen Bibliotheken zusammen. Tatsächlich benötigt SpaceProp sogar eine längere Laufzeit für den REAL Space im Vergleich zum KnowledgeSpace, obwohl letzterer eine um vier Größenordnungen höhere Anzahl an implizit beschriebenen Molekülen enthält. Insgesamt berechnet SpaceProp in der approximativen Variante die Eigenschaftsverteilungen aller vier Bibliotheken in jeweils weniger als drei Minuten, GalaXi und CHEMriya benötigen sogar weniger als 20 Sekunden.

Der prozentuale Fehler im approximativen Ansatz, der durch das Runden der internen Eigenschaftskomponenten entsteht, beträgt auf allen vier kombinatorischen virtuellen chemischen Bibliotheken für das Molekulargewicht weniger als 1%. Für den aLogP ist der prozentuale Fehler sogar kleiner als 0,1%. Der durchschnittliche Rundungsfehler ist ebenfalls sehr klein. Der Vollständigkeit halber soll hier erwähnt werden, dass sich zwei Rundungsfehler in entgegengesetzter Richtung ausgleichen somit im Unterschied der exakten und approximativen Verteilung nicht auftauchen. Dadurch kann der durchschnittliche Rundungsfehler größer sein, allerdings führen diese ausgleichenden Rundungsfehler eben zu keinem Fehler in der approximativen Verteilung. Insgesamt lässt sich sagen, dass der approximative Ansatz Verteilungen generiert, die sehr nah an den exakten Verteilungen

sind. Damit kann diese Variante gut verwendet werden, um die Laufzeit von SpaceProp für große kombinatorische Bibliotheken zu minimieren.

6.3.5. Grenzen des Verfahrens

Die Methode SpaceProp wurde bewusst so entwickelt und beschrieben, dass auch Verteilungen anderer physikochemischer Eigenschaften damit berechnet werden können. Eine Voraussetzung der Anwendbarkeit von SpaceProp ist allerdings die klare Definierbarkeit von interner und externer Eigenschaftskomponente sowie der Randinformation. Dies ist nicht unbedingt für jede Eigenschaft möglich. Beispielsweise sind die Hauptrotationsachsen eines Produkts nicht direkt aus den einzelnen Fragmenten ersichtlich. Somit könnte keine interne Eigenschaftskomponente bestimmt werden, um die Form des Moleküls anhand seiner Trägheitsmomente zu beschreiben. [165]

Eine weitere Einschränkung von SpaceProp ergibt sich aus dem möglichen Wertebereich einer physikochemischen Eigenschaft. Durch das Runden interner Eigenschaftskomponenten in der approximativen Variante von SpaceProp wird der mögliche Wertebereich für das Molekulargewicht und den aLogP jeweils um zwei Größenordnungen verringert. Aus Abbildung 6.1 kann entnommen werden, dass diese Einschränkung zu einer deutlich niedrigeren Laufzeit speziell für den KnowledgeSpace mit 10^{14} Produkten führt. Die Ergebnisse weisen aber auch im Umkehrschluss darauf hin, dass die Laufzeit von SpaceProp auf dem KnowledgeSpace für eine physikochemische Eigenschaft mit beispielsweise einem um zwei Größenordnung höheren Wertebereich als das ungerundete Molekulargewicht eventuell schon unpraktikabel lang wäre. Würde die Kombination aus Molekulargewicht und aLogP als zwei-dimensionale physikochemische Eigenschaft betrachtet werden, wäre der Wertebereich noch größer und eine effiziente exakte Verteilungsberechnung durch SpaceProp wahrscheinlich nicht mehr möglich. Für die Kombination aller vier Eigenschaften der 5er-Regel gilt Ähnliches. Diese Kombination wäre allerdings nötig, um die exakte Anzahl von Produkten einer kombinatorischen Bibliothek zu bestimmen, die die 5er-Regel erfüllen.

6.4. Eigenschaftsverteilungen prominenter kombinatorischer Bibliotheken

Mit SpaceProp ist es nun erstmals möglich exakte physikochemische Eigenschaftsverteilungen aller Produkte einer kombinatorischen Bibliothek zu bestimmen, auch wenn diese nicht praktikabel enumerierbar ist. Im Folgenden wird SpaceProp angewendet, um die exakten Verteilungen der fünf von der Methode betrachteten physikochemischen Eigenschaften für den REAL Space [81], GalaXi [82], CHEMriya [83] und KnowledgeSpace [84] zu bestimmen.

In Abbildung 6.4 sind die Histogramme der Eigenschaftsverteilungen gezeigt. Um die Verteilungen der vier kombinatorischen Bibliotheken vergleichbar zu machen, ist die Größe der Histogrammklassen relativ angegeben. Der REAL Space und GalaXi weisen für alle fünf Eigenschaften durchschnittlich die niedrigsten Werte auf. Die Produkte des CHEMriya haben einen höheren durchschnittlichen Eigenschaftswert für alle fünf Eigenschaften und sogar den höchstens für den aLogP. In Kapitel 5.4 wurde mit der Methode SpaceCompare der paarweise und gesamte Schnitt von REAL Space, GalaXi und CHEMriya bestimmt. Hier wies CHEMriya prozentual und absolut die kleinste Schnittmenge mit den anderen zwei Bibliotheken auf. Dies spiegelt sich nun in der starken Abweichung der Histogramme zwischen CHEMriya einerseits und REAL Space und GalaXi andererseits wider. In [D4] schätzen wir den Anteil von Produkten des REAL Space ab, die die 5er-Regel [145] erfüllen und geben für REAL Space, GalaXi und CHEMriya enthaltende Beispielmoleküle an, die trotz Verletzung der 5er-Regel zu bekannten Molekülklassen für Medikamente gehören.

6.5. Optimierung von Eigenschaftsverteilungen

Anhand der von SpaceProp generierten Verteilungen ist es nun auch möglich zu untersuchen, ob die durchschnittlichen Eigenschaften von Produkten einer kombinatorischen Bibliothek für eine bestimmte Anwendung im Wirkstoffentwurf geeignet sind. Dies kann eine wichtige Voraussetzung sein, um beispielsweise mithilfe einer Ähnlichkeitssuche wie SpaceLight interessante Moleküle zu extrahieren, die eventuell als Kandidaten für einen Wirkstoff untersucht werden können. Für die Abschätzung der oralen Bioverfügbarkeit werden oft vor allem obere Schranken für Eigenschaftswerte betrachtet. [145], [146] Deshalb wurden im Rahmen von SpaceProp Funktionalitäten implementiert, um

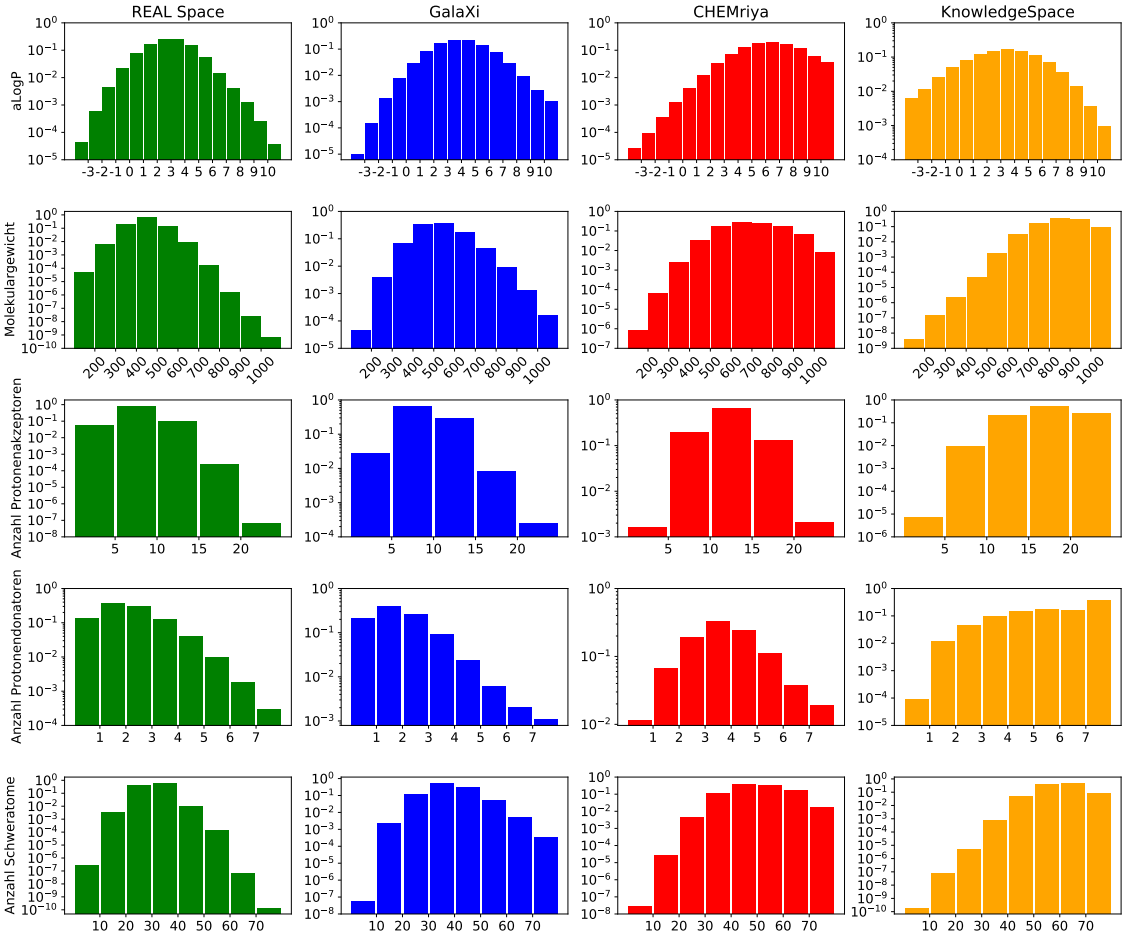


Abbildung 6.4.: Die Eigenschaftsverteilungen des REAL Space, GalaXi, CHEMriya und des KnowledgeSpace für alle fünf von SpaceProp berücksichtigten Eigenschaften. Die Anzahl Protonenakzeptoren und -donatoren stehen für die Anzahl potenzieller Protonenakzeptoren und -donatoren einer Wasserstoffbrückenbindung. Die y-Achse ist in logarithmischer Skala notiert und gibt die anteilige Größe der Klasse im Vergleich zur gesamten Anzahl Produkte in der kombinatorischen Bibliothek an. Die Klassen des Histogramms beinhalten alle Moleküle mit einem Wert größer oder gleich dem Wert auf der x-Achse links der Klasse (falls vorhanden) und kleiner als der Wert rechts der Klasse auf der x-Achse (falls vorhanden). Aus [D4] entnommen, angepasst und ins Deutsche übersetzt.

kombinatorische Teilbibliotheken mit geringeren durchschnittlichen Eigenschaftswerten zu erzeugen. Hierfür wird eine der fünf von SpaceProp betrachteten physikochemischen Eigenschaften ausgewählt und die interne Eigenschaftskomponente jedes Fragments eines topologischen Fragmentraums bestimmt. Dann wird ein vom Nutzer definierter Anteil

der Fragmente mit den höchsten internen Eigenschaftskomponenten entfernt. Mit den verbleibenden Fragmenten wird ein topologischer Teilraum erzeugt. Die Annahme ist, dass hierdurch auch die durchschnittlichen Eigenschaftswerte des Teilraums im Vergleich zum ursprünglichen Raum sinken. In Kapitel 6.6 wird auf mögliche Erweiterungen dieses Ansatzes eingegangen. Das dieses Verfahren keine Multiplikation von Verteilungen benötigt, skaliert es direkt mit der Anzahl Fragmente eines topologischen Fragmentraums und hängt nicht von der Produktmenge ab.

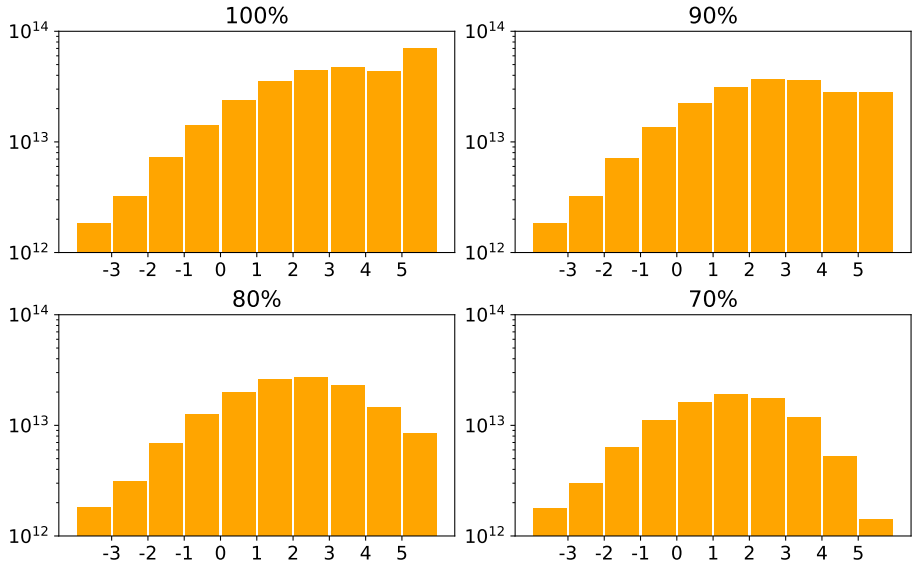


Abbildung 6.5.: aLogP-Verteilungen für Teilbibliotheken des KnowledgeSpace. Über jedem Histogramm ist der Prozentsatz verbleibender Fragmente zur Erzeugung des Teilraums angegeben. Das Histogramm oben links zeigt die aLogP-Verteilung der Produkte des gesamten KnowledgeSpace. Die y-Achse gibt die absolute Größe der Klassen des Histogramms in logarithmischer Skala an. Aus [D4] entnommen und angepasst.

In Abbildung 6.5 sind die Ergebnisse des Optimierungsverfahrens für den KnowledgeSpace bezüglich des aLogP gezeigt. Je höher der Anteil entfernter Fragmente mit einer hohen internen aLogP-Komponente ist, desto geringer fällt der durchschnittliche aLogP-Wert der Produkte des Teilraums aus. Für den gesamten KnowledgeSpace liegt der durchschnittliche aLogP-Wert über fünf und damit über der oberen Schranke der 5er-Regel für den log P. Nach Entfernen von 30% aller Fragmente haben nur noch weniger als 10% der finalen Moleküle des erzeugten Teilraums einen aLogP von fünf oder höher. Allerdings enthält der Teilraum auch circa eine um eine Größenordnung geringere Anzahl von Produkten. Insgesamt zeigen die Ergebnisse, dass sich auch die durchschnittlichen

Eigenschaftswerte von Produkten durch das Entfernen von Fragmenten mit hoher interner Eigenschaftskomponente gezielt verringern lassen. Zusätzlich ist zu sehen, dass die Klassen mit niedrigem aLogP-Wert weitestgehend unverändert bleiben. Somit werden nur wenige Produkte mit niedrigem aLogP-Wert entfernt was auf die Selektivität des Optimierungsverfahrens in diesem Kontext hinweist.

6.6. Ausblick

Das algorithmische Verfahren SpaceProp wurde bewusst modular in der Codebasis NAOMI [54] implementiert und die Methode so beschrieben, dass sie prinzipiell leicht um weitere physikochemische Eigenschaften erweitert werden. Hierbei muss allerdings auf die in Kapitel 6.3.5 beschrieben Einschränkungen durch die Definierbarkeit interner und externen Eigenschaftskomponenten sowie die eventuelle Verwendung der approximativen Variante von SpaceProp für Eigenschaften mit großem Wertebereich geachtet werden. Zwei Eigenschaften, die sich für die Integration im Rahmen der SpaceProp-Methode anbieten sind die Anzahl rotierbarer Bindungen [146] eines Moleküls und sein TPSA-Wert. [164] Beide basieren auf der Betrachtung einzelner Bindungen bzw. Atome und ihrer chemischen Umgebung. Damit wäre das Vorgehen bei der Bildung interner Eigenschaftskomponente und Randinformationen methodisch ähnlich zum Ansatz für den aLogP. Ein TPSA-Wert ist eine Dezimalzahl mit maximal zwei Nachkommastellen und die Anzahl der rotierbaren Bindungen ist ganzzahlig. Somit haben beide einen eingeschränkten Wertebereich und die exakte Variante von SpaceProp kann auch für große kombinatorische Bibliotheken verwendet werden.

Das in Kapitel 6.5 vorgestellte Optimierungsverfahren entfernt die Fragmente eines topologischen Fragmentraums mit den höchsten internen Eigenschaftskomponenten. Obwohl im Beispiel des KnowledgeSpace und aLogP der Fall, muss dies nicht immer eine selektive Entfernung von Produkten mit hohem Eigenschaftswert zur Folge haben. Beispielsweise kann eine niedrige externe Eigenschaftskomponente oder interne Eigenschaftskomponente anderer Fragmente zu insgesamt niedrigen Eigenschaftswerten führen, obwohl die interne Eigenschaftskomponente eines Fragments hoch ist. Um dies zu verhindern, könnten beispielsweise zunächst wie im algorithmischen Vorgehen von SpaceProp Fragmente aus einem Topologieknoten mit der gleichen Randinformation gruppiert werden. Pro Gruppe werden nun die Fragmente mit den höchsten internen Eigenschaftskomponenten bestimmt. Auf diese Weise können ohne Enumeration die höchsten Eigenschaftswerte

von Produkten pro Topologiegraph oder sogar pro Kombination von Randinformationen bestimmt werden. Nur Falls die erzielten Eigenschaftswerte im Vergleich zu anderen Topologiegraphen oder Kombinationen von Randinformationen hoch sind, werden nun Fragmente entfernt. Es könnte ebenfalls von Interesse sein, nicht nur Produkte mit hohem Eigenschaftswert, sondern allgemein Ausreißer zu entfernen, um die Varianz in der Eigenschaftsverteilung zu verringern. Hierfür könnten auch Fragmente mit besonders niedriger interner Eigenschaftskomponente aus dem topologischen Fragmentraum entfernt werden.

7. Zusammenfassung und Ausblick

Die in dieser Dissertation beschriebenen algorithmischen Verfahren dienen der Ähnlichkeitssuche in und Analyse von kombinatorischen virtuellen chemischen Bibliotheken. Jede vorgestellte Methode bietet für ihre jeweilige Problemstellung neuartige Funktionalitäten, die bisher nicht für kombinatorische Bibliotheken existierten. Diese Bibliotheken verwenden chemische Bausteine und Reaktionen, um einen potenziell sehr großen kombinatorischen Raum von Produkten implizit zu beschreiben. Alle in dieser Dissertation beschriebenen Verfahren verwenden dieses algorithmische kombinatorische Paradigma, indem sie auf Repräsentationen chemischer Bausteine operieren und gleichzeitig die Menge implizit beschriebener Produkte durchsuchen oder analysieren.

Topologische Fragmenträume bilden eine neuartige, kompakte Darstellungsform kombinatorischer Bibliotheken, die als Grundlage für alle weiteren Verfahren in dieser Dissertation verwendet wird. Die CSFP Methode liefert einen feingranularen molekularen Fingerabdruck, der durch seine mathematischen Eigenschaften speziell im Rahmen kombinatorischer Bibliotheken ein gutes Hilfsmittel darstellt. SpaceLight ermöglicht erstmals eine topologische Ähnlichkeitssuche in kombinatorischen Bibliotheken, die auf chemischen Substrukturen basiert, wie sie für enumerierte Bibliotheken zu den Standardwerkzeugen der Chemieinformatik gehört. Allerdings ist SpaceLight durch seinen kombinatorischen Ansatz in der Lage, 100 Billionen Produkte innerhalb von Sekunden zu durchsuchen und grenzt sich damit von existierenden topologischen Suchverfahren ab. Mit SpaceCompare ist es möglich die Schnittmenge kombinatorischer Bibliotheken zu berechnen und mit SpaceProp können physikochemische Eigenschaftsverteilungen bestimmt werden. Mithilfe dieser Verfahren können nicht-enumerierbare kombinatorische Bibliotheken erstmals umfassend analysiert, verglichen und optimiert werden.

In ihrer Gesamtheit erweitern diese Methodiken den Anwendungsschatz für kombinatorische virtuelle chemische Bibliotheken. Zusammen mit den existierenden Verfahren FTrees-FS und SpaceMACS, sind nun viele der klassischen, auf Molekulargraphen basierenden Methoden der Chemieinformatik für kombinatorische Bibliotheken verfügbar. Für

zukünftige Forschungsprojekte bietet sich der Schritt zu Verfahren an, die auf räumlichen Konformationen von Fragmenten topologischer oder klassischer Fragmenträume operieren. Hierfür könnte beispielsweise der Molekulardeskriptor *ray volume matrix* (RVM) auf Fragmente, und eine Adaption des Partitionierungsansatz von SpaceLight auf die Unterteilung von Bindetaschen angewendet werden. Mit so einem Ansatz könnte erstmal das Docking für kombinatorische Bibliotheken ermöglicht werden. Denkbar wäre auch den Paarungsansatz der Verfahren FTrees-FS und SpaceMACS anzupassen, um damit eine Ähnlichkeitssuche auf Basis räumlicher Überlagerungen zu entwickeln. Im Hinblick auf das gestiegene generelle Interesse an kombinatorischen Bibliotheken stellt die Entwicklung neuer algorithmischer Verfahren ein zukunftsträchtiges Forschungsfeld mit vielen interessanten offenen Fragestellungen und einem wichtigen Beitrag für den Medikamententwurf dar.

Glossar

Die folgenden Definitionen gelten ausschließlich für diese Dissertation und können von den Definitionen aus anderen Quellen abweichen.

A | **B** | **C** | **D** | **E** | **F** | **G** | **I** | **K** | **L** | **M** | **P** | **R** | **S** | **T** | **U** | **V**

A

additive physikochemische Eigenschaft Eine physikochemische Eigenschaft ist *additiv*, wenn für ein beliebiges Produkt eines topologischen Fragmentraums dessen Eigenschaftswert mit der Summe von Eigenschaftswerten seiner Fragmente übereinstimmt.

algorithmisches kombinatorisches Paradigma Ein algorithmisches Verfahren, dass die computergestützte Durchsuchung oder Analyse von kombinatorischen virtuellen chemischen Bibliotheken ermöglicht, sollte auf der Menge der repräsentierten chemischen Bausteine und Verknüpfungsregeln operieren. Gleichzeitig muss es in der Lage sein den kombinatorischen Produktraum zu durchsuchen bzw zu analysieren..

B

Baustein-Produkt-Substruktur-Kategorien Für einen Syntheseweg ergeben sich drei Kategorien chemischer Substrukturen eines Bausteins oder des Produkts. Eine chemische Substruktur ist *stabil*, wenn sie sowohl im Produkt, als auch in mindestens einem der chemischen Bausteine vorkommt. Eine chemische Substruktur ist *instabil*, wenn sie in mindestens einem chemischen Baustein existiert, allerdings im Produkt nicht vorkommt. Schlussendlich ist eine Substruktur *kreuzend*, wenn sie im Produkt besteht, allerdings in keiner der chemischen Bausteine vorkommt.

Bindungstyp Der Typ einer chemischen kovalenten Bindung zwischen zwei Atomen. Falls die Bindung innerhalb eines aromatischen Rings verläuft, ist ihr Typ 'aromatisch'. Ansonsten ist ihr Typ 'einfach', 'zweifach' oder 'dreifach'.

C

chemische Substruktur Ein zusammenhängender Teil eines Moleküls, der durch die enhaltenen Atome und Bindungen definiert ist. Im Gegensatz zu Subgraphen kann ein Molekül mehrere äquivalente chemische Substrukturen enthalten, die an anderer Stelle im Molekül vorkommen. Ab Kapitel 5 unterscheiden wir chemische Substrukturen zusätzlich nach der Valenz der enthaltenen Atome sowie deren Aromatizität und Konnektivität innerhalb und außerhalb der Substruktur.

chemischer Baustein Ein Molekül, dass in einem Syntheseprotokoll zur Erzeugung eines Produktmoleküls verwendet wird.

D

DNA-kodierte Bibliothek Eine kombinatorische physische chemische Bibliothek, deren Moleküle mit einem Marker aus DNA-Fragmenten verknüpft sind. Jedes DNA-Fragment repräsentiert einen chemischen Baustein, der zur Synthese des Moleküls verwendet wurde.

E

enumerierte virtuelle chemische Bibliothek Eine virtuelle chemische Bibliothek in der jedes Molekül durch einen eigenen Eintrag repräsentiert ist.

erweiterter Fingerabdruck Beschreibt alle chemischen Substrukturen eines Fragments sowie alle kreuzenden Substrukturen aller Produkte, die mithilfe des Fragments gebildet werden können und das Fragment schneiden. Es werden nur Substrukturen mit höchstens sechs Atomen betrachtet. Der erweiterte Fingerabdruck enthält den fCSFP Identifikator jeder dieser Substrukturen.

externe Eigenschaftskomponente Die Summe der Atomwerte aller externen Atome eines Produkts bzw. der zugrundeliegenden Fragmentkombination.

F

fCSFP Variante Unterscheidet sich von der klassischen CSFP Variante nur in der Vernachlässigung stereochemischer Eigenschaften.

Fingerabdruck-Teilmengenrelation Gilt für eine Methode zur Erzeugung topologischer molekularer Fingerabdrücke, wenn für ein beliebiges Fragment eines topologischen Fragmentraums alle Identifikatoren seines Fingerabdrucks im Fingerabdruck aller finalen Produkte enthalten ist, die mit der Hilfe des Fragments gebildet werden können..

Fragment Eine virtuelle Repräsentation eines chemischen Bausteins, die Linker enthält und an die Konfiguration des Bausteins im finalen Produktmolekül angepasst ist. Fragmente eines topologischen Fragmentraums können zusätzlich Ringplatzhalter enthalten.

Fragmentraum Eine Darstellungsform einer kombinatorischen virtuellen chemischen Bibliothek. Ein Fragmentraum enthält Fragmente und Verknüpfungsregeln aus kompatiblem Paaren von Linkern.

G

gCSFP Variante Betrachtet nur wenige Atomeigenschaften und liefer damit eine grobgranularere Beschreibung.

I

iCSFP Variante Betrachtet nur Eigenschaften des Subgraphen selbst und der erzeugte Identifikator ist unabhängig von der Umgebung des Subgraphen.

Identifikator Eine Zahl, die beispielsweise mithilfe einer Hashfunktion generiert wurde, um eine chemische Substruktur zu repräsentieren.

interne Eigenschaftskomponente Die Summe der Atomwerte aller internen Atome eines Fragments.

internes Atom Ein Atom eines Fragments ist *intern* bezüglich einer physikochemischen Eigenschaft, wenn sein Atomwert ableitbar ist, ohne die Umgebung des Fragments zu betrachten. Andernfalls wird es als *extern* bezeichnet.

K

klassische CSFP Variante Betrachtet die meisten Atom- und Bindungseigenschaften und liefert so einen sehr feingranularen molekularen Fingerabdruck.

kombinatorische physische chemische Bibliothek Eine physische chemische Bibliothek, deren Moleküle mithilfe von chemischen Bausteinen und auf sie angewandter chemischer Reaktionen kombinatorisch synthetisiert werden.

kombinatorische virtuelle chemische Bibliothek Eine virtuelle chemische Bibliothek, die Repräsentationen chemischer Bausteine und Verknüpfungsregeln enthält. Die dadurch beschriebenen Produkte werden nicht explizit aufgelistet.

L

Linker Ein Platzhalter, der eine Verknüpfungsstelle in einem Fragment markiert und durch einen Typ identifiziert werden kann.

M

Molekulardeskriptor Eine virtuelle Repräsentation eines Moleküls. Sie beschreibt chemische Eigenschaften wie zB. dessen Topologie, das Vorkommen einer chemischen Substruktur oder Pharmakophors oder physikochemische Eigenschaften.

molekularer Fingerabdruck Ein Molekulardeskriptor der chemische Eigenschaften als Liste von Zahlen, Vektor oder Bitfolge repräsentiert. Durch diese kompakte Form können molekulare Fingerabrücke besonders effizient algorithmisch behandelt werden.

Molekulargraph Der Graph der sich aus den Atomen eines Moleküls als Knoten und den Bindungen zwischen ihnen als Kanten ergibt. Es werden keine zwei Atome oder Bindungen als äquivalent betrachtet.

P

Partition-Topologiegraph-Kompatibilität Eine Partition des Molekulargraphen eines Anfragemoleküls ist zu einem Topologiegraphen eines topologischen Fragmentraumes *kompatibel*, wenn eine Paarung zwischen kompatiblen Subgraphen der Partition und Knoten des Topologiegraphen existiert.

pCSFP Variante Betrachtet die gleichen Atom- und Bindungseigenschaften wie die klassische CSFP Variante, enumeriert aber nur Subgraphen mit Pfadstruktur.

physische chemische Bibliothek Besteht aus einer Menge von Molekülen, die physisch zB. in einer Lösung vorliegen.

Prinzip der Anfragepartitionierung Die Anfrage, beispielsweise der Molekulargraph eines Anfragemoleküls, wird unterteilt und die generierten Partitionen zusammen mit den Repräsentationen chemischer Bausteine einer kombinatorischen virtuellen chemischen Bibliothek algorithmisch verarbeitet, um beispielsweise die ähnlichsten implizit beschriebenen Produkte zu identifizieren.

R

Randinformation Besteht aus allen externen Atomen eines Fragments, sowie den Informationen die benötigt werden, um die Atomwerte externer Atome des Fragments selbst und anderer Fragmente zu bestimmen.

Ringplatzhalter Ein Platzhalter, der in einem Ring eines Fragments enthalten ist. Im Gegensatz zu einem Linker hat ein Ringplatzhalter keinen Typ und dient lediglich dazu die Größe eines Rings im Produktmolekül zu imitieren.

S

Subgraph Enthält eine Teilmenge der Atome eines Molekulargraphen und zusätzlich eine Teilmenge der Bindungen zwischen diesen Atomen, sodass eine zusammenhängende Graphstruktur entsteht. Wie die Knoten eines Molekulargraphen, sind auch keine zwei Subgraphen äquivalent, wenn sie nicht die gleichen Atomen und Bindungen an den gleichen Positionen enthalten. Ein Subgraph wird als *induziert* bezeichnet, wenn alle Bindungen des Moleküls zwischen den Atomen der Teilmenge auch im Subgraphen enthalten sind. Falls nicht weiter spezifiziert, sind induzierte Subgraphen gemeint.

Subgraph-Fragment-Kompatibilität Ein Subgraph des Molekulargraphen eines Anfragemoleküls und ein Fragment eines topologischen Fragmentraumes sind *kompatibel*, wenn sich ihre Anzahl enthaltener Schweratome um höchstens fünf unterscheidet. Zusätzlich muss die Anzahl Bindungen mit genau einem Atom aus dem Subgraphen der Anzahl an Bindungen zu Linkern im Fragment entsprechen.

Subgraph-Knoten-Kompatibilität Ein Subgraph des Molekulargraphen eines Anfragemoleküls und ein Knoten eines Topologiegraphen sind *kompatibel*, wenn der Subgraph zu mindestens einem der im Knoten enthaltenen Fragmente kompatibel ist.

T

tCSFP Variante Betrachtet die gleichen Eigenschaften wie der TopologicalTorsion Fingerabdruck [97].

Topologiegraph Eine Graphstruktur, die ein Syntheseprotokoll mit variablen chemischen Bausteinen beschreibt. Ein Topologiegraph besteht aus Topologieknoten und Topologiekanten und ist Teil eines topologischen Fragmentraums.

Topologiekante Eine Kante eines Topologiegraphen, die eine geformte Bindung während des beschriebenen Syntheseprotokolls repräsentiert. Sie enthält die zwei adjazenten Knoten, sowie einen Linktypen pro Knoten und den Bindungstypen. Bei der Generierung von Produkten werden aus den Fragmenten der Knoten jeweils der Linker des angegeben Typs entfernt und eine Bindung des angegeben Typs zwischen ihren Schweratomnachbarn geknüpft.

Topologieknoten Ein Knoten eines Topologiegraphen der eine Menge von Fragmenten mit der gleichen Konfiguration von Linkern enthält.

Topologiewert Beschreibt die Ähnlichkeit eines kompatiblen Paares von Partition des Molekulargraphen eines Anfragemoleküls und Topologiegraphen eines topologischen Fragmentraumes bezüglich der Komposition ihrer Bindungen zwischen den Subgraphen der Partition und Kanten des Topologiegraphen. Eine genaue Definition ist in Anhang B.3 gegeben. Wir bezeichnen eine Paarung als *topologisch gleichwertig*, wenn die seinen Topologiewert von eins aufweist.

topologischer Fragmentraum Eine Darstellungsform einer kombinatorischen virtuellen chemischen Bibliothek. Der topologische Fragmentraum baut auf dem klassischen Fragmentraum auf und enthält Fragmente und Topologiegraphen.

U

Ueberdeckung Die Fragmentkombination $y = \{y_1, \ldots y_j\}$ *überdeckt* die Fragmentkombination $x = \{x_1, \ldots x_i\}$, wenn $F(x_1) \cup \cdots \cup F(x_i) \subseteq E(y_1) \cup \cdots \cup E(y_j)$. Für ein Fragment a ist $F(a)$ dessen fCSFP1.6 Fingerabdruck und $E(a)$ sein erweiterter Fingerabdruck. Sowohl x als auch y kann aus nur einem Fragment bestehen. y ist eine *minimale Überdeckung* von x, falls x von keiner echten Teilkombination $y' \subset y$ überdeckt wird.

V

virtuelle chemische Bibliothek Ein Datensatz, der eine Menge von Molekülen virtuell beschreibt und computergestützt prozessiert werden kann.

Literatur

Literaturverzeichnis externer Quellen

[1] A. D. McNaught und A. Wilkinson, *Compendium of Chemical Terminology*. Blackwell Science Oxford, 1997, Bd. 1669.

[2] H.-J. Böhm, G. Klebe und H. Kubinyi, *Wirkstoffdesign: Der Weg zum Arzneimittel*. Spektrum Akad. Verlag, 1996.

[3] R. S. Bohacek, C. McMartin und W. C. Guida, „The Art and Practice of Structure-Based Drug Design: a Molecular Modeling Perspective," *Med. Res. Rev.*, Jg. 16, Nr. 1, S. 3–50, 1996.

[4] S. H. Sleigh und C. L. Barton, „Repurposing Strategies for Therapeutics," *Pharmaceutical Medicine*, Jg. 24, Nr. 3, S. 151–159, 2010.

[5] R. Macarron, M. N. Banks, D. Bojanic, D. J. Burns, D. A. Cirovic, T. Garyantes, D. V. Green, R. P. Hertzberg, W. P. Janzen, J. W. Paslay u. a., „Impact of High-Throughput Screening in Biomedical Research," *Nat. Rev. Drug Discov.*, Jg. 10, Nr. 3, S. 188–195, 2011.

[6] R. P. Hertzberg und A. J. Pope, „High-Throughput Screening: New Technology for the 21st Century," *Curr. Opin. Chem. Biol.*, Jg. 4, Nr. 4, S. 445–451, 2000.

[7] D. M. Volochnyuk, S. V. Ryabukhin, Y. S. Moroz, O. Savych, A. Chuprina, D. Horvath, Y. Zabolotna, A. Varnek und D. B. Judd, „Evolution of Commercially Available Compounds for HTS," *Drug Discov. Today*, Jg. 24, Nr. 2, S. 390–402, 2019.

[8] S. Michael, D. Auld, C. Klumpp, A. Jadhav, W. Zheng, N. Thorne, C. P. Austin, J. Inglese und A. Simeonov, „A Robotic Platform for Quantitative High-Throughput Screening," *Assay Drug Dev. Technol.*, Jg. 6, Nr. 5, S. 637–657, 2008.

[9] Y. Chen und H. Tang, „High-Throughput Screening Assays to Identify Small Molecules Preventing Photoreceptor Degeneration Caused by the Rhodopsin P23H Mutation," in *Rhodopsin*, Springer, 2015, S. 369–390.

[10] M. R. Player, „Target-Based Compound Library Design and Synthesis,“ *Drug Discov. Today: Targets*, Jg. 2, Nr. 3, S. 48–50, 2004.

[11] T. R. Webb, „Current Directions in the Evolution of Compound Libraries,“ *Curr. Opin. Drug. Discov. Devel.*, Jg. 8, Nr. 3, S. 303–308, 2005.

[12] A. Furka, „Study on the Possibilities of Systematic Searching for Pharmaceutically Useful Peptides,“ *Notarized Report (File number 36237/1982, in Hungarian)*, 1982.

[13] Á. Furka, „Combinatorial Chemistry: 20 Years on. . .,“ *Drug Discov. Today*, Jg. 7, Nr. 1, S. 1–4, 2002.

[14] T. Kodadek, „The Rise, Fall and Reinvention of Combinatorial Chemistry,“ *ChemComm*, Jg. 47, Nr. 35, S. 9757–9763, 2011.

[15] R. Liu, X. Li und K. S. Lam, „Combinatorial Chemistry in Drug Discovery,“ *Curr. Opin. Chem. Biol.*, Jg. 38, S. 117–126, 2017.

[16] H. M. Geysen, R. H. Meloen und S. J. Barteling, „Use of Peptide Synthesis to Probe Viral Antigens for Epitopes to a Resolution of a Single Amino Acid,“ *Proc. Natl. Acad. Sci. U.S.A.*, Jg. 81, Nr. 13, S. 3998–4002, 1984.

[17] A. Furka, F. Sebestyén, M. Asgedom und G. Dibó, „General Method for Rapid Synthesis of Multicomponent Peptide Mixtures,“ *Int. J. Pept. Protein Res.*, Jg. 37, Nr. 6, S. 487–493, 1991.

[18] R. B. Merrifield, „Solid Phase Synthesis,“ *Science*, Jg. 232, S. 341–348, 1986.

[19] R. A. Houghten, „General Method for the Rapid Solid-Phase Synthesis of Large Numbers of Peptides: Specificity of Antigen-Antibody Interaction at the Level of Individual Amino Acids,“ *Proc. Natl. Acad. Sci. U.S.A.*, Jg. 82, Nr. 15, S. 5131–5135, 1985.

[20] A. Frankel, S. W. Millward und R. W. Roberts, „Encodamers: Unnatural Peptide Oligomers Encoded in RNA,“ *Chem. Biol.*, Jg. 10, Nr. 11, S. 1043–1050, 2003.

[21] D. Morton, S. Leach, C. Cordier, S. Warriner und A. Nelson, „Synthesis of Natural-Product-Like Molecules with Over Eighty Distinct Scaffolds,“ *Angew. Chem.*, Jg. 121, Nr. 1, S. 110–115, 2009.

[22] M. Ohlmeyer, R. N. Swanson, L. W. Dillard, J. C. Reader, G. Asouline, R. Kobayashi, M. Wigler und W. C. Still, „Complex Synthetic Chemical Libraries Indexed with Molecular Tags,“ *Proc. Natl. Acad. Sci. U.S.A.*, Jg. 90, Nr. 23, S. 10 922–10 926, 1993.

[23] S. Brenner und R. A. Lerner, „Encoded Combinatorial Chemistry.," *Proc. Natl. Acad. Sci. U.S.A.*, Jg. 89, Nr. 12, S. 5381–5383, 1992.

[24] M. A. Clark, R. A. Acharya, C. C. Arico-Muendel, S. L. Belyanskaya, D. R. Benjamin, N. R. Carlson, P. A. Centrella, C. H. Chiu, S. P. Creaser, J. W. Cuozzo u. a., „Design, Synthesis and Selection of DNA-Encoded Small-Molecule Libraries," *Nat. Chem. Biol.*, Jg. 5, Nr. 9, S. 647–654, 2009.

[25] M. C. Needels, D. G. Jones, E. H. Tate, G. L. Heinkel, L. M. Kochersperger, W. J. Dower, R. W. Barrett und M. A. Gallop, „Generation and Screening of an Oligonucleotide-Encoded Synthetic Peptide Library," *Proc. Natl. Acad. Sci. U.S.A.*, Jg. 90, Nr. 22, S. 10 700–10 704, 1993.

[26] R. M. Weisinger, S. J. Wrenn und P. B. Harbury, „Highly Parallel Translation of DNA Sequences Into Small Molecules," *PLoS One*, Jg. 7, Nr. 3, e28056, 2012.

[27] W. Decurtins, M. Wichert, R. M. Franzini, F. Buller, M. A. Stravs, Y. Zhang, D. Neri und J. Scheuermann, „Automated Screening for Small Organic Ligands Using DNA-Encoded Chemical Libraries," *Nat. Protoc.*, Jg. 11, Nr. 4, S. 764–780, 2016.

[28] R. M. Franzini und C. Randolph, „Chemical Space of DNA-Encoded Libraries," *J. Med. Chem.*, Jg. 59, Nr. 14, S. 6629–6644, 2016.

[29] R. A. Goodnow Jr, C. E. Dumelin und A. D. Keefe, „DNA-encoded Chemistry: Enabling the Deeper Sampling of Chemical Space," *Nat. Rev. Drug Discov.*, Jg. 16, Nr. 2, S. 131–147, 2017.

[30] L. C. Ray und R. A. Kirsch, „Finding Chemical Records by Digital Computers," *Science*, Jg. 126, Nr. 3278, S. 814–819, 1957.

[31] A. M. Moore, „A Line-Formula Chemical Notation," *J. Am. Chem. Soc.*, Jg. 77, Nr. 7, S. 2032–2032, 1955.

[32] C. A. James, D. Weininger und J. Delany, „Daylight Theory Manual. Daylight Chemical Information Systems," *Inc., Irvine, CA*, 1995.

[33] S. R. Heller, A. McNaught, I. Pletnev, S. Stein und D. Tchekhovskoi, „InChI, the IUPAC International Chemical Identifier," *J. Cheminformatics*, Jg. 7, Nr. 1, S. 1–34, 2015.

[34] A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. Gushurst, D. L. Grier, B. A. Leland und J. Laufer, „Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited," *J. Chem. Inf. Comput. Sci.*, Jg. 32, Nr. 3, S. 244–255, 1992.

[35] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov und P. E. Bourne, „The Protein Data Bank," *Nucleic Acids Res.*, Jg. 28, Nr. 1, S. 235–242, 2000.

[36] J. W. Raymond und P. Willett, „Maximum Common Subgraph Isomorphism Algorithms for the Matching of Chemical Structures," *J. Comput.Aided Mol. Des.*, Jg. 16, Nr. 7, S. 521–533, 2002.

[37] C. James, D. Weininger und J. Delany, *SMARTS Theory. Daylight Theory Manual*, 2000.

[38] R. Todeschini und V. Consonni, *Handbook of Molecular Descriptors*. Weinheim: John Wiley & Sons, 2008, Bd. 11.

[39] D. B. Kitchen, H. Decornez, J. R. Furr und J. Bajorath, „Docking and Scoring in Virtual Screening for Drug Discovery: Methods and Applications," *Nat. Rev. Drug Discov.*, Jg. 3, Nr. 11, S. 935–949, 2004.

[40] J. D. MacCuish und N. E. MacCuish, „Chemoinformatics Applications of Cluster Analysis," *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, Jg. 4, Nr. 1, S. 34–48, 2014.

[41] S. A. Wildman und G. M. Crippen, „Prediction of Physicochemical Parameters by Atomic Contributions," *J. Chem. Inf. Model.*, Jg. 39, Nr. 5, S. 868–873, 1999.

[42] A. Leo, P. Jow, C. Silipo und C. Hansch, „Calculation of Hydrophobic Constant (log P) From π and f Constants," *J. Med. Chem.*, Jg. 18, Nr. 9, S. 865–868, 1975.

[43] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani und J. P. Overington, „ChEMBL: a Large-scale Bioactivity Database for Drug Discovery," *Nucleic Acids Res.*, Jg. 40, Nr. D1, S. D1100–D1107, 2011.

[44] J. J. Irwin und B. K. Shoichet, „ZINC- a Free Database of Commercially Available Compounds for Virtual Screening," *J. Chem. Inf. Model.*, Jg. 45, Nr. 1, S. 177–182, 2005.

[45] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang und S. H. Bryant, „PubChem: a Public Information System for Analyzing Bioactivities of Small Molecules," *Nucleic Acids Res.*, Jg. 37, W623–W633, 2009.

[46] A. Shivanyuk, S. Ryabukhin, A. Tolmachev, A. Bogolyubsky, D. Mykytenko, A. Chupryna, W. Heilman und A. Kostyuk, „Enamine Real Database: Making Chemical Diversity Real," *Chim. Oggi – Chem. Today*, Jg. 25, Nr. 6, S. 58–59, 2007.

[47] W. P. Walters, „Virtual Chemical Libraries,“ *J. Med. Chem.*, Jg. 62, Nr. 3, S. 1116–1124, 2018.

[48] J. J. Irwin, K. G. Tang, J. Young, C. Dandarchuluun, B. R. Wong, M. Khurelbaatar, Y. S. Moroz, J. Mayfield und R. A. Sayle, „ZINC20-A Free Ultralarge-scale Chemical Database for Ligand Discovery,“ *J. Chem. Inf. Model.*, Jg. 60, Nr. 12, S. 6065–6073, 2020.

[49] *OpenEye Large Scale Virtual Screening*, last accessed on 22/06/2020. Adresse: https://www.eyesopen.com/large-scale-virtual-screening.

[50] A. R. Leach und M. M. Hann, „The *in silico* World of Virtual Libraries,“ *Drug Discov. Today*, Jg. 5, Nr. 8, S. 326–336, 2000.

[51] J. M. Blaney und E. J. Martin, „Computational Approaches for Combinatorial Library Design and Molecular Diversity Analysis,“ *Curr. Opin. Chem. Biol.*, Jg. 1, Nr. 1, S. 54–59, 1997.

[52] D. H. Drewry und S. S. Young, „Approaches to the Design of Combinatorial Libraries,“ *Chemom. Intell. Lab. Syst.*, Jg. 48, Nr. 1, S. 1–20, 1999.

[53] M. Rarey und M. Stahl, „Similarity Searching in Large Combinatorial Chemistry Spaces,“ *J. Comput. Aided Mol. Des.*, Jg. 15, Nr. 6, S. 497–520, 2001.

[54] S. Urbaczek, A. Kolodzik, J. R. Fischer, T. Lippert, S. Heuser, I. Groth, T. Schulz-Gasch und M. Rarey, „NAOMI: On the Almost Trivial Task of Reading Molecules from Different File Formats,“ *J. Chem. Inf. Model.*, Jg. 51, Nr. 12, S. 3199–3207, 2011.

[55] *Enamine Building Blocks*, last accessed on 18/01/2022. Adresse: https://enamine.net/building-blocks.

[56] *Otava Building Blocks*, last accessed on 18/01/2022. Adresse: https://otavachemicals.com/products/chemical-building-blocks.

[57] *Sigma-Aldrich Building Blocks*, last accessed on 18/01/2022. Adresse: https://www.sigmaaldrich.com/DE/de/products/chemistry-and-biochemicals/building-blocks.

[58] *ChemBridge Building Blocks*, last accessed on 20/01/2022. Adresse: https://www.chembridge.com/building_blocks/.

[59] *eMolecules*, last accessed on 05/08/2021. Adresse: https://www.emolecules.com/.

[60] M. Hartenfeller, M. Eberle, P. Meier, C. Nieto-Oberhuber, K.-H. Altmann, G. Schneider, E. Jacoby und S. Renner, „A Collection of Robust Organic Synthesis Reactions for *in silico* Molecule Design," *J. Chem. Inf. Model.*, Jg. 51, Nr. 12, S. 3093–3098, 2011.

[61] E. Corey, „Computer-Assisted Analysis of Complex Synthetic Problems," *Q Rev Chem Soc*, Jg. 25, Nr. 4, S. 455–482, 1971.

[62] D. A. Pensak und E. J. Corey, „LHASA — Logic and Heuristics Applied to Synthetic Analysis," in ACS Publications, 1977.

[63] E. J. Corey, A. K. Long und S. D. Rubenstein, „Computer-Assisted Analysis in Organic Synthesis," *Science*, Jg. 228, Nr. 4698, S. 408–418, 1985.

[64] J. Gasteiger, M. Marsili, M. Hutchings, H. Saller, P. Loew, P. Röse und K. Rafeiner, „Models for the Representation of Knowledge About Chemical Reactions," *J. Chem. Inf. Comput. Sci.*, Jg. 30, Nr. 4, S. 467–476, 1990.

[65] X. Q. Lewell, D. B. Judd, S. P. Watson und M. M. Hann, „RECAP Retrosynthetic Combinatorial Analysis Procedure: a Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry," *J. Chem. Inf. Comput. Sci.*, Jg. 38, Nr. 3, S. 511–522, 1998.

[66] J. Degen, C. Wegscheid-Gerlach, A. Zaliani und M. Rarey, „On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces," *ChemMedChem*, Jg. 3, Nr. 10, S. 1503, 2008.

[67] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano und T. Laino, „Predicting Retrosynthetic Pathways Using Transformer-Based Models and a Hyper-Graph Exploration Strategy," *Chem.*, Jg. 11, Nr. 12, S. 3316–3325, 2020.

[68] M. H. Segler und M. P. Waller, „Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction," *Chem. Eur. J.*, Jg. 23, Nr. 25, S. 5966–5971, 2017.

[69] M. H. Segler, M. Preuss und M. P. Waller, „Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI," *Nature*, Jg. 555, Nr. 7698, S. 604–610, 2018.

[70] H. Patel, W.-D. Ihlenfeldt, P. N. Judson, Y. S. Moroz, Y. Pevzner, M. L. Peach, V. Delannée, N. I. Tarasova und M. C. Nicklaus, „SAVI, *in silico* Generation of Billions of Easily Synthesizable Compounds Through Expert-System Type Rules," *Sci. Data*, Jg. 7, Nr. 1, S. 1–14, 2020.

[71] W. D. Ihlenfeldt, Y. Takahashi, H. Abe und S.-i. Sasaki, „Computation and Management of Chemical Properties in CACTVS: An Extensible Networked Epproach Toward Modularity and Compatibility," *J. Chem. Inf. Comput. Sci.*, Jg. 34, Nr. 1, S. 109–116, 1994.

[72] F. Chevillard und P. Kolb, „SCUBIDOO: A Large yet Screenable and Easily Searchable Database of Computationally Created Chemical Compounds Optimized Toward High Likelihood of Synthetic Tractability," *J. Chem. Inf. Model.*, Jg. 55, Nr. 9, S. 1824–1835, 2015.

[73] G. Landrum, *RDKit: Open-source Cheminformatics, version 2018.09.1*, last accessed on 09/09/2018. Adresse: https://www.rdkit.org.

[74] J. B. Baell und G. A. Holloway, „New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays," *J. Med. Chem.*, Jg. 53, Nr. 7, S. 2719–2740, 2010.

[75] Y. Zabolotna, D. M. Volochnyuk, S. V. Ryabukhin, K. Gavrylenko, D. Horvath, O. Klimchuk, O. Oksiuta, G. Marcou und A. Varnek, „SynthI: A New Open-Source Tool for Synthon-Based Library Design," *J. Chem. Inf. Model.*, 2021.

[76] G. Schneider, M.-L. Lee, M. Stahl und P. Schneider, „De Novo Design of Molecular Architectures by Evolutionary Assembly of Drug-Derived Building Blocks," *J. Comput. Aided Mol. Des.*, Jg. 14, Nr. 5, S. 487–494, 2000.

[77] *World Drug Index*, last accessed on 21/01/2022. Adresse: https://www.daylight.com/products/wdi.html.

[78] G. Schneider, W. Neidhart, T. Giller und G. Schmid, „Grundgerüstwechsel (Scaffold-Hopping) durch topologische Pharmakophorsuche: ein Beitrag zum virtuellen Screening," *Angew. Chem.*, Jg. 111, Nr. 19, S. 3068–3070, 1999.

[79] *OpenChemLib*, last accessed on 16/02/2022. Adresse: https://github.com/Actelion/openchemlib.

[80] J. Wahl und T. Sander, „Fully Automated Creation of Virtual Chemical Fragment Spaces Using the Open-Source Library OpenChemLib," *J. Chem. Inf. Model.*, 2022.

[81] *Enamine REAL Space*, last accessed on 05/08/2021. Adresse: https://enamine.net/library-synthesis/real-compounds/real-space-navigator.

[82] *GalaXi Space*, last accessed on 05/08/2021. Adresse: https://www.labnetwork.com/frontend-app/p/#!/library/virtual.

[83] *CHEMriya Space*, last accessed on 23/08/2021. Adresse: https://www.otavachemicals.com/products/chemriya.

[84] C. Detering, H. Claussen, M. Gastreich und C. Lemmen, „KnowledgeSpace-a Publicly Available Virtual Chemistry Space," *J. Cheminformatics*, Jg. 2, Nr. 1, S. 1–1, 2010.

[85] U. Lessel, B. Wellenzohn, M. Lilienthal und H. Claussen, „Searching Fragment Spaces with Feature Trees," *J. Chem. Inf. Model.*, Jg. 49, Nr. 2, S. 270–279, 2009.

[86] W. A. Warr, *NIH Meeting Ultra-Large Databases in Chemistry*, 2020. Adresse: https://chemrxiv.org/articles/preprint/Report_on_an_NIH_Workshop_on_Ultralarge_Chemistry_Databases/14554803 (besucht am 12. 08. 2021).

[87] C. Grebner, *Webinar: Exploration and Mining of Large Virtual Chemical Spaces*, Mölndal, Sweden, 2018. Adresse: https://youtu.be/fMrI11SXwpU.

[88] T. Hoffmann und M. Gastreich, „The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries," *Drug Discov. Today*, 2019.

[89] F.-M. Klingler, M. Gastreich, O. O. Grygorenko, O. Savych, P. Borysko, A. Griniukova, K. E. Gubina, C. Lemmen und Y. S. Moroz, „SAR by Space: Enriching Hit Sets from the Chemical Space," *Molecules*, Jg. 24, Nr. 17, S. 3096, 2019.

[90] M. Rarey und J. S. Dixon, „Feature Trees: a New Molecular Similarity Measure Based on Tree Matching," *J. Comput. Aided Mol. Des.*, Jg. 12, Nr. 5, S. 471–490, 1998.

[91] O. Diels und K. Alder, „Synthesen in der Hydroaromatischen Reihe," *Justus Liebigs Annalen der Chemie*, Jg. 460, Nr. 1, S. 98–122, 1928.

[92] R. Diestel, „The Basics," in *Graph Theory*, Springer, 2017, S. 1–34.

[93] *SQLite*, last accessed on 28/01/2022. Adresse: https://www.sqlite.org/index.html.

[94] L. P. Hammett, „The Effect of Structure Upon the Reactions of Organic Compounds. Benzene Derivatives," *J. Am. Chem. Soc.*, Jg. 59, Nr. 1, S. 96–103, 1937.

[95] R. E. Carhart, D. H. Smith und R. Venkataraghavan, „Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications," *J. Chem. Inf. Comput. Sci.*, Jg. 25, Nr. 2, S. 64–73, 1985.

[96] „MACCS Structural Keys," *Accelrys: San Diego*, 2011.

[97] R. Nilakantan, N. Bauman, J. S. Dixon und R. Venkataraghavan, „Topological Torsion: a New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors," *J. Chem. Inf. Comput. Sci.*, Jg. 27, Nr. 2, S. 82–85, 1987.

[98] D. Rogers und M. Hahn, „Extended-Connectivity Fingerprints," *J. Chem. Inf. Model.*, Jg. 50, Nr. 5, S. 742–754, 2010.

[99] P. Torkington, „The Moments of Inertia of Molecules with Internal Rotation," *J. Chem. Phys.*, Jg. 18, Nr. 4, S. 407–413, 1950.

[100] A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé und G. Pujadas, „Molecular Fingerprint Similarity Search in Virtual Screening," *Methods*, Jg. 71, S. 58–63, 2015.

[101] I. Muegge und P. Mukherjee, „An Overview of Molecular Fingerprint Similarity Search in Virtual Screening," *Expert Opin. Drug Discov.*, Jg. 11, Nr. 2, S. 137–148, 2016.

[102] M. D. Mackey und J. L. Melville, „Better Than Random? The Chemotype Enrichment Problem," *J. Chem. Inf. Model.*, Jg. 49, Nr. 5, S. 1154–1162, 2009.

[103] M. Yang, B. Tao, C. Chen, W. Jia, S. Sun, T. Zhang und X. Wang, „Machine Learning Models Based on Molecular Fingerprints and an Extreme Gradient Boosting Method Lead to the Discovery of JAK2 Inhibitors," *J. Chem. Inf. Model.*, Jg. 59, Nr. 12, S. 5002–5012, 2019.

[104] *PubChem Substructure Fingerprint*, last accessed on 02/02/2022. Adresse: https://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.txt.

[105] *Daylight Fingerprint*, last accessed on 02/02/2022. Adresse: https://www.daylight.com/dayhtml/doc/theory/theory.finger.html.

[106] *OpenEye Scientific Software, OEChem*, last accessed on 20/08/2019. Adresse: https://www.eyesopen.com.

[107] M. J. McGregor und S. M. Muskal, „Pharmacophore Fingerprinting. 1. Application to QSAR and Focused Library Design," *J. Chem. Inf. Comput. Sci.*, Jg. 39, Nr. 3, S. 569–574, 1999.

[108] Z. Deng, C. Chuaqui und J. Singh, „Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein- ligand binding interactions," *J. Med. Chem.*, Jg. 47, Nr. 2, S. 337–344, 2004.

[109] C. Da und D. Kireev, „Structural protein–ligand interaction fingerprints (SPLIF) for structure-based virtual screening: method and benchmark study," *J. Chem. Inf. Model.*, Jg. 54, Nr. 9, S. 2555–2561, 2014.

[110] D. Weininger, A. Weininger und J. L. Weininger, „SMILES. 2. Algorithm for Generation of Unique SMILES Notation," *J. Chem. Inf. Comput. Sci.*, Jg. 29, Nr. 2, S. 97–101, 1989.

[111] R. S. Cahn, C. Ingold und V. Prelog, „Specification of Molecular Chirality," *Angew. Chem., Int. Ed. Engl.*, Jg. 5, Nr. 4, S. 385–415, 1966.

[112] S. Riniker und G. A. Landrum, „Open-Source Platform to Benchmark Fingerprints for Ligand-based Virtual Screening," *J. Cheminf.*, Jg. 5, Nr. 1, S. 26, 2013.

[113] P. Jaccard, „Lois de Distribution Florale dans la Zone Alpine," *Bull Soc Vaudoise Sci Nat*, Jg. 38, S. 69–130, 1902.

[114] S. G. Rohrer und K. Baumann, „Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data," *J. Chem. Inf. Model.*, Jg. 49, Nr. 2, S. 169–184, 2009.

[115] J. J. Irwin, „Community Benchmarks for Virtual Screening," *J. Comput.Aided Mol. Des.*, Jg. 22, Nr. 3-4, S. 193–199, 2008.

[116] J.-F. Truchon und C. I. Bayly, „Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem," *J. Chem. Inf. Model.*, Jg. 47, Nr. 2, S. 488–508, 2007.

[117] T. Fawcett, „ROC Graphs: Notes and Practical Considerations for Researchers," *Mach. Learn.*, Jg. 31, Nr. 1, S. 1–38, 2004.

[118] R. Schmidt, F. Krull, A. L. Heinzke und M. Rarey, „Disconnected Maximum Common Substructures Under Constraints," *J. Chem. Inf. Model.*, Jg. 61, Nr. 1, S. 167–178, 2020.

[119] R. Schmidt, R. Klein und M. Rarey, „Maximum Common Substructure Searching in Combinatorial Make-on-Demand Compound Spaces," *J. Chem. Inf. Model.*, 2021.

[120] A. G. Dossetter, E. J. Griffen und A. G. Leach, „Matched Molecular Pair Analysis in Drug Discovery," *Drug Discov. Today*, Jg. 18, Nr. 15-16, S. 724–731, 2013.

[121] E. S. Ehmki und M. Rarey, „Exploring Structure–Activity Relationships with Three-Dimensional Matched Molecular Pairs—A Review," *ChemMedChem*, Jg. 13, Nr. 6, S. 482–489, 2018.

[122] M. Johnson, S. Basak und G. Maggiora, „A Characterization of Molecular Similarity Methods for Property Prediction," *Math. Comput. Model.*, Jg. 11, S. 630–634, 1988.

[123] Y. C. Martin, J. L. Kofron und L. M. Traphagen, „Do Structurally Similar Molecules Have Similar Biological Activity?" *J. Med. Chem.*, Jg. 45, Nr. 19, S. 4350–4358, 2002.

[124] S. L. Dixon und K. M. Merz, „One-Dimensional Molecular Representations and Similarity Calculations: Methodology and Validation," *J. Med. Chem.*, Jg. 44, Nr. 23, S. 3795–3809, 2001.

[125] M. M. Cone, R. Venkataraghavan und F. W. McLafferty, „Molecular Structure Comparison Program for the Identification of Maximal Common Substructures," *J. Am. Chem. Soc.*, Jg. 99, Nr. 23, S. 7668–7671, 1977.

[126] L. R. Dice, „Measures of the Amount of Ecologic Association Between Species," *Ecology*, Jg. 26, Nr. 3, S. 297–302, 1945.

[127] Y. Otsuka, „The Faunal Character of the Japanese Pleistocene Marine Mollusca, as Evidence of Climate Having Become Colder During the Pleistocene in Japan," *Biogeograph. Soc. Japan*, Jg. 6, Nr. 16, S. 165–170, 1936.

[128] C. Merlot, D. Domine, C. Cleva und D. J. Church, „Chemical Substructures in Drug Discovery," *Drug Discov. Today*, Jg. 8, Nr. 13, S. 594–602, 2003.

[129] M. Rarey, B. Kramer, T. Lengauer und G. Klebe, „A Fast Flexible Docking Method Using an Incremental Construction Algorithm," *J. Mol. Biol.*, Jg. 261, Nr. 3, S. 470–489, 1996.

[130] *ROCS*, last accessed on 11/02/2022. Adresse: https://www.eyesopen.com/rocs.

[131] T. S. Rush, J. A. Grant, L. Mosyak und A. Nicholls, „A Shape-Based 3-D Scaffold Hopping Method and its Application to a Bacterial Protein-Protein Interaction," *J. Med. Chem.*, Jg. 48, Nr. 5, S. 1489–1495, 2005.

[132] P. Penner, V. Martiny, A. Gohier, M. Gastreich, P. Ducrot, D. Brown und M. Rarey, „Shape-Based Descriptors for Efficient Structure-Based Fragment Growing," *J. Chem. Inf. Model.*, Jg. 60, Nr. 12, S. 6269–6281, 2020.

[133] C. Lemmen, T. Lengauer und G. Klebe, „FLEXS: A Method for Fast Flexible Ligand Superposition," *J. Med. Chem.*, Jg. 41, Nr. 23, S. 4502–4520, 1998.

[134] O. Korb, P. Monecke, G. Hessler, T. Stutzle und T. E. Exner, „pharmACOphore: Multiple Flexible Ligand Alignment Based on Ant Colony Optimization," *J. Chem. Inf. Model.*, Jg. 50, Nr. 9, S. 1669–1681, 2010.

[135] G. Wolber und T. Langer, „LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and their Use as Virtual Screening Filters," *J. Chem. Inf. Model.*, Jg. 45, Nr. 1, S. 160–169, 2005.

[136] *Moleculare Operating Environment (MOE)*, last accessed on 14/02/2022. Adresse: https://www.chemcomp.com/Products.htm.

[137] D. R. Koes und C. J. Camacho, „Pharmer: Efficient and Exact Pharmacophore Search," *J. Chem. Inf. Model.*, Jg. 51, Nr. 6, S. 1307–1314, 2011.

[138] *SmallWorld*, last accessed on 16/02/2022. Adresse: https://www.nextmovesoftware.com/smallworld.html.

[139] T. H. Cormen, C. E. Leiserson, R. L. Rivest und C. Stein, *Introduction to Algorithms*. MIT press, 2009.

[140] L. D. Pennington, B. M. Aquila, Y. Choi, R. A. Valiulin und I. Muegge, „Positional Analogue Scanning: An Effective Strategy for Multiparameter Optimization in Drug Design," *J. Med. Chem.*, Jg. 63, Nr. 17, S. 8956–8976, 2020.

[141] R. Wang, X. Fang, Y. Lu, C.-Y. Yang und S. Wang, „The PDBbind Database: Methodologies and Updates," *J. Med. Chem.*, Jg. 48, Nr. 12, S. 4111–4119, 2005.

[142] U. Lessel und C. Lemmen, „Comparison of Large Chemical Spaces," *ACS Med. Chem. Lett.*, Jg. 10, Nr. 10, S. 1504–1510, 2019.

[143] L. Ruddigkeit, R. Van Deursen, L. C. Blum und J.-L. Reymond, „Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17," *J. Chem. Inf. Model.*, Jg. 52, Nr. 11, S. 2864–2875, 2012.

[144] Q. Li, „Application of Fragment-Based Drug Discovery to Versatile Targets," *Front. Mol. Biosci*, S. 180, 2020.

[145] C. A. Lipinski, F. Lombardo, B. W. Dominy und P. J. Feeney, „Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings," *Adv. Drug Deliv. Rev.*, Jg. 23, Nr. 1-3, S. 3–25, 1997.

[146] D. F. Veber, S. R. Johnson, H.-Y. Cheng, B. R. Smith, K. W. Ward und K. D. Kopple, „Molecular Properties that Influence the Oral Bioavailability of Drug Candidates," *J. Med. Chem.*, Jg. 45, Nr. 12, S. 2615–2623, 2002.

[147] M. Congreve, R. Carr, C. Murray und H. Jhoti, „A 'Rule of Three' for Fragment-Based Lead Discovery?“ *Drug Discov. Today*, Jg. 8, Nr. 19, S. 876–877, 2003.

[148] P. G. Polishchuk, T. I. Madzhidov und A. Varnek, „Estimation of the Size of Drug-Like Chemical Space Based on GDB-17 Data,“ *J. Comput. Aided Mol. Des.*, Jg. 27, Nr. 8, S. 675–679, 2013.

[149] J. M. Sangster, *Octanol-water partition coefficients: fundamentals and physical chemistry*. John Wiley & Sons, 1997, Bd. 1.

[150] A. Leo, C. Hansch und D. Elkins, „Partition Coefficients and Their Uses,“ *Chem. Rev.*, Jg. 71, Nr. 6, S. 525–616, 1971.

[151] T. Ishida und A. Ciulli, „E3 Ligase Ligands for PROTACs: How They Were Found and How to Discover New Ones,“ *SLAS Discov.*, Jg. 26, Nr. 4, S. 484–502, 2021.

[152] J. Vagner, H. Qu und V. J. Hruby, „Peptidomimetics, a Synthetic Tool of Drug Discovery,“ *Curr. Opin. Chem. Biol.*, Jg. 12, Nr. 3, S. 292–296, 2008.

[153] J. A. Nieto-Garai, B. Glass, C. Bunn, M. Giese, G. Jennings, B. Brankatschk, S. Agarwal, K. Börner, F. X. Contreras, H.-J. Knölker u. a., „Lipidomimetic Compounds Act as HIV-1 Entry Inhibitors by Altering Viral Membrane Structure,“ *Front. Immunol.*, Jg. 9, S. 1983, 2018.

[154] I. Lukovits, „Correlation Between Components of the Wiener Index and Partition Coefficients of Hydrocarbons,“ *Int. J. Quantum Chem.*, Jg. 44, Nr. S19, S. 217–223, 1992.

[155] H. Wiener, „Structural Determination of Paraffin Boiling Points,“ *J. Am. Chem. Soc.*, Jg. 69, Nr. 1, S. 17–20, 1947.

[156] G. Klopman, J.-Y. Li, S. Wang und M. Dimayuga, „Computer Automated log P Calculations Based on an Extended Group Contribution Approach,“ *J. Chem. Inform. Comput. Sci.*, Jg. 34, Nr. 4, S. 752–781, 1994.

[157] H. Zhu, A. Sedykh, S. K. Chakravarti und G. Klopman, „A New Group Contribution Approach to the Calculation of LogP,“ *Curr Comput Aided Drug Des*, Jg. 1, Nr. 1, S. 3–9, 2005.

[158] A. K. Ghose und G. M. Crippen, „Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships I. Partition Coefficients as a Measure of Hydrophobicity,“ *J. Comput. Chem.*, Jg. 7, Nr. 4, S. 565–577, 1986.

[159] A. K. Ghose und G. M. Crippen, „Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships. 2. Modeling Dispersive and Hydrophobic Interactions," *J. Chem. Inf. Comput. Sci.*, Jg. 27, Nr. 1, S. 21–35, 1987.

[160] A. K. Ghose, A. Pritchett und G. M. Crippen, „Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships III: Modeling Hydrophobic Interactions," *J. Comput. Chem.*, Jg. 9, Nr. 1, S. 80–90, 1988.

[161] R. Mannhold, G. I. Poda, C. Ostermann und I. V. Tetko, „Calculation of Molecular Lipophilicity: State-Of-The-Art and Comparison of log P Methods on More than 96,000 Compounds," *J. Pharm. Sci.*, Jg. 98, Nr. 3, S. 861–893, 2009.

[162] C. Schärfer, T. Schulz-Gasch, H.-C. Ehrlich, W. Guba, M. Rarey und M. Stahl, „Torsion Angle Preferences in Druglike Chemical Space: a Comprehensive Guide," *J. Med. Chem.*, Jg. 56, Nr. 5, S. 2016–2028, 2013.

[163] H. Pajouhesh und G. R. Lenz, „Medicinal Chemical Properties of Successful Central Nervous System Drugs," *NeuroRx*, Jg. 2, Nr. 4, S. 541–553, 2005.

[164] P. Ertl, B. Rohde und P. Selzer, „Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and its Application to the Prediction of Drug Transport Properties," *J. Med. Chem.*, Jg. 43, Nr. 20, S. 3714–3717, 2000.

[165] W. H. Sauer und M. K. Schwarz, „Molecular Shape Diversity of Combinatorial Libraries: a Prerequisite for Broad Bioactivity," *J. Chem. Inf. Comput. Sci.*, Jg. 43, Nr. 3, S. 987–1003, 2003.

[166] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte und D. Vrgoč, „Foundations of JSON Schema," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, S. 263–273.

[167] K. Schomburg, H.-C. Ehrlich, K. Stierand und M. Rarey, „From Structure Diagrams to Visual Chemical Patterns," *J. Chem. Inf. Model.*, Jg. 50, Nr. 9, S. 1529–1535, 2010.

[168] P. Hall, „On Representatives of Subsets," *J. London Math. Soc*, Jg. 10, Nr. 1, S. 26–30, 1935.

[169] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith u. a., „Array Programming with NumPy," *Nature*, Jg. 585, Nr. 7825, S. 357–362, 2020.

Literaturverzeichnis der kummulativen Dissertation

[D1] L. Bellmann, P. Penner und M. Rarey, „Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration," *J. Chem. Inf. Model.*, Jg. 59, Nr. 11, S. 4625–4635, 2019.

[D2] L. Bellmann, P. Penner und M. Rarey, „Topological Similarity Search in Large Combinatorial Fragment Spaces," *J. Chem. Inf. Model.*, Jg. 61, Nr. 1, S. 238–251, 2021.

[D3] L. Bellmann, P. Penner, M. Gastreich und M. Rarey, „Comparison of Combinatorial Fragment Spaces and Its Application to Ultralarge Make-on-Demand Compound Catalogs," *J. Chem. Inf. Model.*, Jg. 62, Nr. 3, S. 553–566, 2022.

[D4] L. Bellmann, R. Klein und M. Rarey, „Calculating and Optimizing Physicochemical Property Distributions of Large Combinatorial Fragment Spaces," *J. Chem. Inf. Model.*, 2022, (akzeptiert).

A. Publikations- und Kongressbeiträge

A.1. Beiträge zu Artikeln der kumulativen Dissertation

Im Folgenden sind die Beiträge der Autoren zu den jeweiligen Publikationen dieser kumulativen Dissertation zusammengefasst

[D1] L. Bellmann, P. Penner und M. Rarey, „Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration," *J. Chem. Inf. Model.*, Jg. 59, Nr. 11, S. 4625–4635, 2019

L. Bellmann entwickelte die zugrundeliegenden Methoden, führte die Evaluation durch und implementierte die Verfahren in der Codebasis NAOMI sowie als Python-Modul CSFPy. P. Penner war an der Auswahl chemischer Eigenschaften für die Bildung der CSFP Varianten beteiligt und bettete die Ergebnisse der Evaluation sowie Beispiele in den chemischen Kontext ein.

[D2] L. Bellmann, P. Penner und M. Rarey, „Topological Similarity Search in Large Combinatorial Fragment Spaces," *J. Chem. Inf. Model.*, Jg. 61, Nr. 1, S. 238–251, 2021

L. Bellmann entwickelte die algorithmischen Verfahren und Konzepte der Evaluation, führte die Analyse durch und implementierte SpaceLight in die Codebasis NAOMI und als Software-Modul. P. Penner war maßgeblich an der Entwicklung der Evaluationsmethode durch Rangkorrelationskoeffizienten und der Auswahl chemischer Beispiele beteiligt.

[D3] L. Bellmann, P. Penner, M. Gastreich und M. Rarey, „Comparison of Combinatorial Fragment Spaces and Its Application to Ultralarge Make-on-Demand Compound Catalogs," *J. Chem. Inf. Model.*, Jg. 62, Nr. 3, S. 553–566, 2022

Die SpaceCompare zugrundeliegenden Methodiken wurden von L.Bellmann erarbeitet sowie die Evaluation konzeptuell entwickelt und durchgeführt. L. Bellmann

implementierte das Verfahren SpaceCompare ebenfalls in die Codebasis NAOMI und als Software-Modul. P. Penner steuerte in der Konzeptionsphase Syntheseprotokolle aus der Literatur bei, wodurch die Lösungsansätze im Bereich der kreuzenden Substrukturen und Fragmentüberdeckung ermöglicht wurden. Zusätzlich trug P. Penner maßgeblich zur Analyse der Schnittmenge der kombinatorischen Bibliotheken der drei Chemiekonzerne bei. M. Gastreich koordinierte die Abstimmung mit Projektpartnern und trug zur Niederschrift der Publikation bei.

[D4] L. Bellmann, R. Klein und M. Rarey, „Calculating and Optimizing Physicochemical Property Distributions of Large Combinatorial Fragment Spaces," *J. Chem. Inf. Model.*, 2022, (akzeptiert)

L. Bellmann erarbeitete die algorithmischen Ansätze von SpaceProp, führte die Evaluation durch und implementierte das Verfahren in die Codebasis NAOMI und als Software-Modul. R. Klein trug maßgeblich zur Analyse der Eigenschaftsverteilungen der vier kombinatorischen Bibliotheken sowie der Auswahl von Beispielen bei.

M. Rarey betreute die Arbeit an allen vier Publikationen und war an der konzeptionellen Entwicklung der Methoden und ihrer Evaluation beteiligt.

A.2. Kongressbeiträge

Die folgende Liste fasst die Beiträge des Autors dieser Dissertation auf internationalen Fachkongressen zusammen.

Vortrag **L. Bellmann** und M. Rarey. Connected Subgraph Fingerprint: From Theory to Applications. *Eigth Joint Sheffield Conference on Chemoinformatics*, 2019, Sheffield, Vereinigtes Königreich

Vortrag **L. Bellmann** und M. Rarey. SpaceLight: Topological Similarity Searching in Large Combinatorial Fragment Spaces. *16th German Conference on Cheminformatics and SAMPL Satellite Workshow - Virtual Edition*, 2020, virtuell

Vortrag **L. Bellmann** und M. Rarey. Computational Approaches for the Analysis of Ultra-Large Make-On-Demand Compound Catalogs. *ACS Spring 2022*, 2022, San Diego, USA

B. Methodische Details

B.1. Erzeugung topologischer Fragmenträume

Topologische Fragmenträume können mithilfe des Programms SpaceLight [D2] erzeugt werden. Hierfür hinterlegt der Nutzer alle relevanten Informationen in einer oder mehreren Dateien des Datenformats JSON. [166] Hierbei werden eine oder mehrere Syntheseprotokolle enkodiert. In diesen Dateien können chemische Reaktionen als reactionSMARTS [37] angegeben werden. Hierbei werden die Atome, die sowohl in einem Edukt als auch dem Produkt einer Reaktion vorkommen mit einer eindeutigen Nummer versehen. Falls im Syntheseprotokoll ein Makrozyklus durch eine Abfolge von chemischen Reaktionen entsteht, kann dies ebenfalls vom Nutzer angegeben werden. Gruppen chemischer Bausteine können entweder direkt in dieser Datei als SMILES [32] oder durch den Pfad zu einer Datei im SMILES,SDF oder MOL2 [34] Datenformat spezifiziert werden. Durch Indices kann der Nutzer zusätzlich bestimmen auf welche Gruppen chemischer Bausteine welche Reaktion angewandt wird und an welcher Position in der Reaktion die Gruppen verwendet werden.

Mithilfe dieser Informationen werden Topologiegraphen und Fragmente erzeugt. Pro Syntheseprotokoll wird ein Topologiegraph erzeugt mit einem Knoten pro Gruppe chemischer Bausteine. Für jeden reactionSMARTS Ausdruck des Syntheseprotokolls werden mehrere repräsentative Graphstrukturen gebildet. [167] Jede dieser Graphstrukturen repräsentiert die chemische Umgebung eines Edukts oder des Produkts der Reaktion. Die Graphstrukturen der Edukte werden auf die spezifizierte Gruppe chemischer Bausteine in einer SMARTS Mustererkennung angewandt. Hierbei werden diejenigen chemischen Bausteine der Gruppe identifiziert, auf die der SMARTS Ausdruck des Edukts zutrifft. Nur diese Bausteine werden zur Erzeugung von Fragmenten verwendet, die im jeweiligen Knoten des Topologiegraphen hinterlegt werden. Ein nicht nummeriertes Atom des SMARTS Ausdrucks des Edukts repräsentiert ein Atom, dass bei der Reaktion

abgespalten wird. Diese Atome werden aus Fragmenten der Gruppe entfernt. Die nummerierten Atome des SMARTS Ausdrucks des Produkts werden nach ihrer Zugehörigkeit zu einem Edukt klassifiziert. Ein nicht nummeriertes Atom im SMARTS Ausdruck des Produkts repräsentiert ein während der Reaktion neu hinzukommendes Atom. Dieses wird den Fragmenten der angrenzenden Gruppe und der Klassen der nummerierten Atome hinzugefügt. Bindungen im SMARTS Ausdruck des Produktes, die zwischen zwei Atomklassen verlaufen, stellen während der Reaktion neu gebildete Bindungen dar. Für jede dieser Bindungen wird ein Linker in den korrespondierenden Fragmentgruppen an der auf den SMARTS Ausdruck des Edukts zutreffenden Stelle hinzugefügt. Zusätzlich wird eine Kante mit diesen Linkern und dem Typ der Bindung zwischen Knoten des Topologiegraphen gebildet. Falls sich Bindungen, Ladungen oder adjazente Wasserstoffe innerhalb der Atomklassen zwischen Edukt und Produkt SMARTS Ausdruck ändern, werden die zutreffenden Bindungen in den Fragmenten an den Produkt SMARTS Ausdruck angepasst. Zuletzt werden neu gebildete Ringe in den Fragmenten geschlossen und wenn nötig mit Ringplatzhaltern aufgefüllt bzw. kontrahiert, um die Größe der Ringe an den Produkt SMARTS Ausdruck anzupassen. Falls ein Syntheseprotokoll mehrere konsekutive chemische Reaktionen enthält, werden die korrespondierenden reactionSMARTS Ausdrücke sukzessive prozessiert. Falls eine Gruppe chemischer Bausteine in mehreren Reaktionen verwendet wird, wird das beschriebene Verfahren der Fragmenterzeugung mit den im vorangegangen Schritt erzeugten Fragmenten, anstatt der chemischen Bausteine, durchgeführt. Dadurch werden an den Fragmenten weitere Änderungen vorgenommen und neue Linker angefügt, um alle Syntheseschritte einzufangen. Im Topologiegraphen werden passend dazu weitere Kanten eingefügt.

Nach der Erzeugung werden die generierten Fragmente und Topologiegraphen in einer SQLite [93] Datenbank mit relationalem Datenbankschema abgespeichert. Dazu wurden die in NAOMI bereits existieren Datenbankschemata [54] erweitert. Zusätzlich werden in der Datenbank für jedes Fragment seine Anzahl Schweratome sowie sein ECFP [98] und CSFP [D1] Fingerabdruck gespeichert. Auf molekulare Fingerabdrücke im Allgemeinen und ECFP und CSFP Fingerabdrücke im Speziellen gehen wir in Kapitel 3 ein.

B.2. Erzeugung des CSFP Identifikators eines Subgraphen

Im Folgenden ist beschrieben, wie innerhalb der in Kapitel 3.2.2 und in [D1] beschriebenen Berechnung des CSFP eines Moleküls der Identifikator eines Subgraphen erzeugt wird.

Vorher wurden bereits alle Identifikatoren jedes Atoms des Subgraphen basierend auf der CSFP Variante gebildet.

Die Atome des Subgraphen werden in eine kanonische Ordnung überführt, die mithilfe der Ideen des CANON-Algorithmus [110] generiert wird. Hierfür wird zunächst eine initiale Ordnung der Atome mithilfe ihrer Identifikatoren gebildet. Danach wird durch sukzessive Betrachtung der erweiterten Nachbarschaft eine kanonische Unterscheidung zwischen Atomen getroffen, die die Graphstruktur und die Identifikatoren der benachbarten Atome betrachtet. Falls am Ende des Verfahrens zwei Atome aufgrund von Symmetrien im Subgraphen äquivalent innerhalb der Ordnung bleiben, wird zwischen ihnen künstlich eine beliebige Ordnung (Tie-Break) eingeführt. Dies wird solange wiederholt bis keine zwei Atome mit gleichem Rang innerhalb der Ordnung mehr existieren. Nun werden die Atome des Subgraphen ausgehend vom kleinsten Atom innerhalb der kanonischen Ordnung rekursive traversiert. Der Resultat-Identfikator wird zunächst mit dem Identifikator des kleinsten Atoms initiiert. Zusätzlich wird eine Datenstruktur für bereits besuchte Atome verwendet. In jedem rekursiven Schritt werden die Nachbarn (Nachfahren) des zuletzt behandelten Atoms (Vorfahren) betrachtet. Falls ein Nachfahr schon vorher innerhalb der Traversion besucht wurde, stellt die gefundene Bindung zwischen ihnen einen Ringschluss im Molekül dar. Falls die verwendete Variante des CSFP Bindungstypen berücksichtigt, wird der Bindungstyp zwischen Vorfahr- und Nachfahratom mithilfe einer Hash-Funktion mit dem Resultat-Identifikator kombiniert. Alle verbleibenden Nachfahren werden mithilfe ihres Identifikators, Rang innerhalb der kanonischen Ordnung und Bindungstyp zum Vorfahratom verglichen. Falls zwei Nachfahren einen unterschiedlichen Identifikator aufweisen, wird der Nachfahr mit einem geringeren Rang innerhalb der kanonischen Ordnung priorisiert. Falls zwei Nachfahren den gleichen Identifikator besitzen, wird derjenige mit kleinerem Bindungstyp oder, wenn der Bindungstyp gleich ist oder die verwendete CSFP Variante Bindungstypen nicht berücksichtigt, mit geringerem Rang in der kanonischen Ordnung, bevorzugt. Der Identifikator des Nachfahren mit der höchsten Priorisierung wird mit dem Resultat-Identifikator kombiniert. Der Bindungstyp zwischen ihm und dem Vorfahratom wird mit dem Resultat-Identifikator kombiniert, falls die CSFP Variante Bindungstypen berücksichtigt. Danach wird der gewählte Nachfahr zur Datenstruktur für besuchte Atome hinzugefügt und als Vorfahr für den nächsten rekursiven Schritt innerhalb der Traversion ausgewählt. Wenn alle Atome traversiert wurden endet das Verfahren und der Resultat-Identifikator wird als Identifikator des Subgraphen zurückgegeben.

B.3. Partitions- und Paarungsschritt in SpaceLight

In diesem Abschnitt beschreiben wir das algorithmische Vorgehen von SpaceLight zur Generierung von Partitionen der Anfrage, die topologisch äquivalent zu Fragmentkombinationen des topologischen Fragmentraumes sind. Aufbauend auf der Subgraph-Fragment-Kompatibilität aus Kapitel 4.3.1 führen wir zunächst weitere Definitionen ein. Ein Subgraph und ein Knoten eines Topologiegraphen erfüllen die *Subgraph-Knoten-Kompatibilität*, wenn der Subgraph zu mindestens einem der im Knoten enthaltenen Fragmente kompatibel ist. Schließlich erfüllen eine Partition des Molekulargraphen eines Anfragemoleküls und ein Topologiegraph eines topologischen Fragmentraumes die *Partition-Topologiegraph-Kompatibilität*, wenn eine vollständige Paarung zwischen kompatiblen Subgraphen der Partition und Knoten des Topologiegraphen existiert. Somit ähnelt eine zu einem Topologiegraphen kompatible Partition in der Größe und Interkonnektivität ihrer Subgraphen einer Fragmentkombination des Topologiegraphen.

Das algorithmische Verfahren SpaceLight beginnt den Partitionierungsschritt mit der Enumeration aller Subgraphen des Molekulargraphen eines Anfragemoleküls, die die Subgraph-Fragment-Kompatibilität zu mindestens einem Fragment des topologischen Fragmentraumes erfüllen. Hierfür wird der in Kapitel 3.2.2 beschriebene CONSENS-Algorithmus verwendet. Zur Filterung wird pro Knoten eines Topologiegraphen des topologischen Fragmentraumes ein Filter erzeugt. Der Filter beinhaltet die Anzahl an erlaubten Kanten, die genau ein Atom des Subgraphen enthalten. Zusätzlich schließt der Filter eine Menge von Intervallen ein, die exakt alle Atomanzahlen ergeben, die für die Subgraph-Fragment-Kompatibilität zu einem Fragment des Knoten benötigt wird. Ein Subgraph wird im CONSENS-Algorithmus zur Ausgabe hinzugefügt, wenn er mindestens einen der Knotenfilter erfüllt. Zusätzlich wird für jeden so enumerierten Subgraphen die Menge der kompatiblen Knoten abgespeichert. Mithilfe dieser Ausgabe werden nun rekursiv Partitionen generiert, die die Partition-Topologiegraph-Kompatibilität zu mindestens einem Topologiegraphen erfüllen. Hierbei wird für die Existenz einer vollständigen Paarung der Heiratssatz verwendet. [168], [92]

Im nachfolgenden Paarungsschritt werden pro kompatiblem Paar von Partition und Topologiegraphen alle vollständigen Paarungen zwischen ihren kompatible Subgraphen und Knoten generiert. Für jede Paarung wird ein *Topologiewert* errechnet. Hierfür wird, zusätzlich zur Knotenpaarung, eine maximale Kantenpaarung erzeugt. Diese wird zwischen den Kanten des Topologiegraphen und den Bindungen des Molekulargraphen des Anfragemoleküls, die zwischen den Subgraphen der Partition verlaufen, gebildet.

Hierbei können eine Kante des Topologiegraphen und Bindung des Molekulargraphen nur dann gepaart werden, wenn sie den gleichen Bindungstyp besitzen und beide zwischen den gleichen zwei Paaren von kompatiblem Knoten und Subgraphen der Knotenpaarung verlaufen. Sei k die Größe einer so generierten maximalen Kantenpaarung und m die Anzahl der Kanten des Topologiegraphen. Ähnlich einem Tanimotokoeffizienten, [113] ergibt sich der Topologiewert dann als $\frac{k}{2m-k}$. Er liegt damit zwischen null und eins, wobei der Topologiewert nur dann eins annimmt, wenn die Komposition der Kanten des Topologiegraphen exakt den Bindungen zwischen den Subgraphen der Partition entspricht. In diesem Fall bezeichnen wir die Paarung als *topologisch gleichwertig*. Dieser Abschnitt des SpaceLight Verfahrens endet mit der Ausgabe aller Partitionen des Anfragemoleküls, die die Partition-Topologiegraph-Kompatibilität zu mindestens einem Topologiegraphen erfüllen und ihrer topologisch gleichwertigen Paarungen.

B.4. Enumeration kreuzender Substrukturen in SpaceCompare

Im Folgenden beschreiben wir das Vorgehen zur effizienten Enumeration aller kreuzender Substrukturen eines topologischen Fragmentraums. Zunächst formulieren wir allerdings einige Beobachtungen. Ihrer Definition nach existiert für jede kreuzende chemische Substruktur eine eindeutige Partitionierung in mindestens zwei Klassen. Diese ergibt sich aus dem Produkt und der zugrundeliegende Fragmentkombination. Jede Klasse der Partitionierung stellt eine stabile Teilsubstruktur dar, die in einem der Fragmente der Kombination enthalten ist. Hierbei kann ein Fragment auch mehrere der Klassen enthalten. Wir stellen fest, dass jede stabile Substruktur mindestens einen adjazenten Linker hat und sich zwei stabile Substrukturen nur dann eine adjazenten Linker teilen, wenn dieser aus einer Kontraktion entstanden ist. Die Entstehung von Linkerkontraktionen haben wir in Anhang B.1 beschrieben.

In Abbildung B.1 (b) ist eine kreuzende Substruktur markiert. In (c) sehen wir dessen eindeutige Partitionierung in stabile Substrukturen. Die Substrukturen aus dem linken Fragment ergeben vereinigt einen unzusammenhängenden Graphen, obwohl die gesamte kreuzende Substruktur zusammenhängend ist. Dies lässt sich darstellen durch einen Hilfsgraphen. Der Kanten(multi-)graph [92] $L(G)$ eines zugrundeliegenden (Multi-)Graphen G definiert für jede Kante von G einen Knoten, also $V(L(G)) = E(G)$. Zwei Knoten sind in $L(G)$ benachbart, wenn die zugrundeliegenden Kanten in G einen gemeinsamen Knoten besitzen. Falls sie zwei gemeinsame Knoten in G besitzen, dann entstehen auch zwei Kanten in $L(G)$. In (d) ist der Kantengraph des Topologiegraphen aus (a) gegeben.

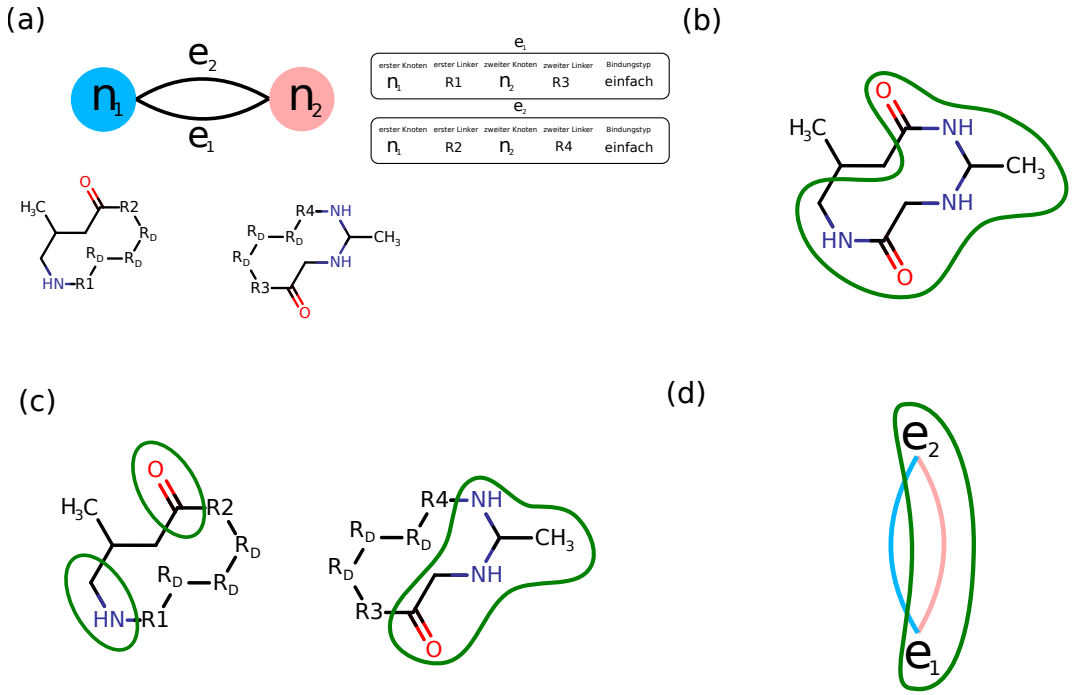


Abbildung B.1.: Partitionierung einer kreuzenden Substruktur. Substrukturen sind durch alle im grün markierten Bereich enthaltenen Atome und Bindungen definiert. In (a) ein Topologiegraph mit jeweils einem Fragment pro Knoten. In (b) das einzige Produkt des Topologiegraphen aus (a) mit einer in grün markierten kreuzenden chemischen Substruktur. In (c) die Partitionierung der kreuzenden Substruktur aus (b) in stabile Substrukturen der Fragmente. In (d) der Kantengraph des Topologiegraphen aus (a). Die Farbe der Kanten entspricht der Farbe des Knoten, in denen die Kanten gemeinsam adjazent sind. Die im grün markierten Bereich enthaltenen Knoten und Bindungen beschreiben die verwendeten Linker und Zusammenhangseigenschaften der Partitionierung aus (c).

Auf dem Kantengraphen existiert nun zu jeder kreuzenden Substruktur ein eindeutiger Teilgraphen L'. Wir stellen zunächst fest, dass jeder Knoten $e \in V(L(G)) = E(G)$ eine bei der Reaktion gebildete Bindung des Produkts repräsentiert, dessen kreuzende Substruktur wir betrachten. Nun wird ein Knoten zu L' hinzugefügt, wenn die repräsentierte Bindung auch in der kreuzenden Substruktur enthalten ist. Für zwei in $L(G)$ benachbarte Knoten $e, e' \in L'$ (bzw. Kanten in G) sei v ihr gemeinsamer Knoten im Topologiegraphen G. Wir fügen die Kante (e, e') zu L' hinzu, wenn die in e und e' enthaltenen Linker des Knoten $v \in V(G)$ beide zur gleichen stabilen Substruktur des Fragments aus v adjazent sind. Äquivalent lässt sich sagen, dass für die zwei durch e und e' repräsentierten Bindungen

in der kreuzenden Substruktur ein Pfad existiert, der ganz im Fragment aus v enthalten ist. In Abbildung B.1 (d) ist der Teilkantengraph L' zur kreuzenden Substruktur aus (b) markiert. e_1 und e_2 sind im Teilgraphen enthalten, weil die repräsentierten Bindungen in der kreuzenden Substruktur aus (b) enthalten sind. Die Linker $R3$ und $R4$ sind beide zur gleichen stabilen Substruktur adjazent. Deshalb ist die magenta gefärbte Kante in L' enthalten. Die Linker $R1$ und $R2$ wiederum sind nicht zur gleichen stabilen Substruktur adjazent. Deshalb kommt die blaue Kante nicht in L' vor. Da aber zwischen e_1 und e_2 ein Pfad in L' existiert, existiert auch insgesamt ein Pfad zwischen den zwei stabilen Substrukturen des Fragments aus n_1 in der gesamten kreuzenden Substruktur.

Generell stellen wir fest, dass der Teilkantengraph zu einer beliebigen kreuzenden Substruktur zusammenhängend sein muss. Wäre dies nicht so, dann wäre auch die kreuzende Substruktur nicht zusammenhängend. Weiterhin lässt sich aus dem Teilkantengraph eindeutig die Verteilung eines Teils der Linker von G auf adjazente stabile Substrukturen der Partitionierung ableiten. Von L' aus (d) können wr also direkt ablesen, dass drei Partitionsklassen bestehen werden. Eine Klasse mit adjazenten Linkern $R3$ und $R4$ und jeweils eine Klasse für die Linker $R1$ und $R2$, da die türkise Kante nicht in L' enthalten ist. Generell sind zwei Linker genau dann adjazent zur gleichen Klasse der Partitionierung, wenn sie aus dem gleichen Knoten $v \in V(G)$ stammen und die Linker repräsentieren Kanten $e, e' \in E(G) = V(L(G))$ eine gemeinsame durch v definierte Kante in L' haben. Mithilfe dieser Beobachtung und Definitionen können wir nun unseren Algorithmus entwickeln. Wir beschreiben das Verfahren anhand eines Topologiegraphen G. Mehrere Topologiegraphen eines topologischen Fragmentraums werden unabhängig voneinander parallel behandelt.

1. Für jeden Knoten v von G betrachten wir alle seine Fragmente und enumerieren ihre Substrukturen, die zu mindestens einem Linker adjazent sind und höchstens fünf Schweratome enthalten. Zur Enumeration verwenden wir das in Kapitel 3.2.2 beschrieben Verfahren CONSENS. Nun werden Substrukturen, die in mehreren Fragmenten aus v vorkommen mithilfe ihres fCSFP Identifikators gefunden und zusammengefasst, wenn sie die gleichen Typen adjazenten Linker aufweisen.

2. Alle enumerierten Substrukturen werden pro Knoten in einer Datenstruktur zusammengefasst. Wir speichern in einer Datenstruktur zu allen Teilmengen von Linkertypen des Knoten v alle enumerierten Substrukturen ab, die zu diesen Linkern adjazent sind. Für jede Substruktur speichern wir weiterhin die sie enthaltenden Fragmente ab.

3. Wir enumerieren rekursiv alle zusammenhängenden Teilkantengraphen L' des Kantengraphen $L(G)$. Für jeden enumerierten Teilkantengraph L' bestimmen wir die eindeutige Verteilung eines Teils der Linker von G in Klassen.

4. Für jeden enumerierten Teilkantengraph L' und jede seiner Klassen von Linkern rufen wir alle enumerierten Substrukturen aus der Datenstruktur aus Schritt 2 ab. Nun bilden wir alle durch L' repräsentierten kreuzenden Substrukturen des Graphen G, indem wir die abgerufenen Substrukturen kombinieren. Hierbei erzeugen wir nur kreuzenden Substrukturen mit höchstens sechs Atomen. Die hinterlegten Fragmente ergeben die Fragmentkombinationen und Produkte, der kreuzenden Substrukturen.

5. Wir fassen alle kreuzenden Substrukturen mit dem gleichen fCSFP Identifikator zusammen. Zu jeder kreuzenden Substruktur speichern wir den Identifikator und alle Fragmente ab, die für eine Klasse ihrer Partitionierungen in der Datenstruktur aus Schritt 1 hinterlegt wurden.

Durch das Zusammenfassen der stabilen Substrukturen für alle Fragmente eines Knoten im Schritt 1 nutzen wir die mutmaßliche Ähnlichkeit der Fragmente an der Reaktiosstelle aus. Falls hier viele stabile Substrukturen zusammengefasst werden, kann die Enumeration der kreuzenden Substrukturen in Schritt 4 deutlich effizienter sein, als die komplette Enumeration aller Produkte des topologischen Fragmentraums.

C. Bedienung der Software

In diesem Kapitel erläutern wir die Bedienungsmöglichkeiten der Software-Module in denen die in dieser Dissertation beschriebenen Verfahren implementiert sind. Alle Beschreibungen sind aus den, unter https://software.zbh.uni-hamburg.de/ verfügbaren Dokumentationen der jeweiligen Software-Module entnommen, angepasst und ins Deutsche übersetzt.

C.1. CSFPy

Das Python-Modul CSFPy ermöglicht die Berechnung und Verwendung der molekularen Fingerabdrücke der CSFP Methode und ihrer Varianten, die wir in Kapitel 3 beschrieben haben. Zunächst kann der Nutzer Moleküle direkt in SMILES Notation oder aus einer Datei im SMILES, SDF oder MOL2 Format einlesen. Hierbei können auch mehrere Moleküle aus einer Datei mithilfe von *MolFactory* Objekten gelesen werden

```
import csfpy
caffeine  = csfpy.Molecule("Cn1cnc2c1c(=O)n(c(=O)n2C)C")
caffeine.name = 'Caffeine'
caffeine.id = 0

theobromine = csfpy.Molecule("theobromine.smi")
theobromine.id = 1

molFac = csfpy.MolFactory("testMols.smi")
mols = molFac.read_all()
```

Für die so eingelesenen Moleküle kann nun der klassische CSFP und alle seine Varianten berechnet werden, wobei der Nutzer eine untere und obere Schranke für die Anzahl der in den betrachteten chemischen Substrukturen enthalten Atome angeben muss. Die

Ergebnisse werden in einem *SparseIntVec* Objekt abgespeichert. Der Tanimoto- [113] und Dice-Koeffizient [126] können mithilfe der Funktionen *tanimoto* und *dice* berechnet werden. *SparseIntVec* Objekte können zusätzlich in Listen der Python Standardbibliothek und in Bit-Arrays des Python-Moduls NumPy [169] konvertiert werden. Für Letzeres muss der Nutzer die gewünschte Länge des Arrays und damit den Teiler für die Faltung der Identifikatoren angeben.

```
cafCsfp = csfpy.csfpy(caffeine, 2, 3)
theoCsfp = csfpy.csfpy(theobromine, 2, 3)
tanScore = csfpy.tanimoto(catCsfp, theoCsfp)

cafFcsfp = csfpy.fcsfpy(caffeine, 1, 6)
theoFcsfp = csfpy.fcsfpy(theobromine, 1, 6)
diceScore = csfpy.dice(catFcsfp, theoFcsfp)

pythonList = cafCsfp.toList()
bitArray = theoFcsfp.toNumpyBitArray(nBits=64)
```

Der Nutzer kann ebenfalls eine eigene Variante des CSFP kreieren, indem er die betrachteten Atomeigenschaften anpasst. Dazu kann er ein *CSFPConfig* Objekt erzeugen und anpassen. *CSFPConfig* Objekte haben folgende Parameter. Die jeweiligen Atomeigenschaften werden betrachtet, wenn der Parameter gesetzt wird. Das Element von Atomen wird immer betrachtet.

`exoInfo` Anzahl der benachbarten Schweratome und ein Valenz-Identifikator, der alle Bindungen zu benachbarten Schweratome betrachtet.

`ringInfo` Zugehörigkeit zu mindestens einem Ring.

`stereoInfo` Stereo-Identifikator, der betrachtet ob das Atom ein Stereozentrum nach den CIP-Regeln [111] ist.

`intraInfo` Anzahl der benachbarten Schweratome innerhalb der betrachteten Substruktur und ein Valenz-Identifikator, der alle Bindungen zu benachbarten Schweratome innerhalb der Substruktur einfängt.

`aroInfo` Aromatizität des Atoms.

`aroValence` Alle aromatische Atome erhalten den gleichen eindeutigen Valenz-Identifikator für die Parameter *exoInfo* und *intraInfo*.

`hashBonds` Jede Bindung erhält einen Identifikator, der den Bindungstyp repräsentiert. Diese Identifikatoren werden in die Identifikatoren der chemischen Substrukturen eingefügt.

`torsionInfo` Nur die Atomeigenschaften der TopologicalTorsion Methode [97] werden betrachtet. Also das Element, die Anzahl benachbarter Schweratome sowie die Anzahl von π-Elektronen. Falls gesetzt, überschreibt dies alle anderen Parameter.

Ein *CSFPConfig* Objekt kann entweder aus dem CSFP oder seinen Varianten erzeugt oder direkt mit eigenen Parametern gebildet werden. Parameter können jederzeit verändert werden. Mithilfe der erzeugten Parametriesierung kann ein benutzerdefinierter CSFP mithilfe der Funktion *csfpCustom* generiert werden.

```
config = csfpy.gettCSFPConfig()
config.ringInfo = True

config = csfpy.CSFPConfig(exoInfo = True, ringInfo = False,
                          stereoInfo = True, intraInfo = True,
                          aroInfo = False, aroValence = False,
                          hashBonds = True, torsionInfo = False)

cafCustom = csfpy.csfpCustom(caffeine, 1, 4, config)
```

C.2. SpaceLight

Das Software-Modul SpaceLight umfasst neben der gleichnamigen Methode zur Ähnlichkeitssuche aus Kapitel 4.3 ebenfalls die Suche nach Fragmenten als molekulare Gerüste aus Kapitel 4.3.4 und die Erzeugung topologischer Fragmente Räume, die wir in Anhang B.1 beschrieben haben. Der Nutzer interagiert mit dem Modul auf der Kommandozeile. Hierbei gibt es drei Modi für die drei Funktionalitäten von SpaceLight. Beim ersten Aufruf muss der Nutzer eine Lizenz angeben mit

```
bin/SpaceLight --license <license string>
```

Für alle drei Modi können folgende Parameter optional vom Nutzer verwendet werden.

```
-h, -help
```

Eine Kurzbeschreibung des Modus und aller verfügbaren Parameter.

`-p, -procs`
Standardmäßig nutzt das Programm alle verfügbaren Prozesse. Mit diesem Parameter kann der Nutzer die gewünschte Anzahl verwendeter Prozesse angeben.

`-u, -userinfo`
Mit diesem Parameter kann der Nutzer die Menge an Informationen beeinflussen, die auf der Konsole ausgegeben wird. Standardmäßig oder bei Stufe zwei werden alle Informationen ausgegeben. Bei Stufe eins werden nur Fehlermeldungen und Warnungen ausgegeben. Bei Stufe null werden nur Fehlermeldungen ausgegeben.

`-license`
Mit diesem Parameter kann der Nutzer eine Lizenz angeben. Die Lizenz wird abgespeichert und muss nicht erneut angegeben werden, bis sie abgelaufen ist.

C.2.1. Topologische Ähnlichkeitssuche

Dieser Modus wird mit dem Parameter `search` ausgewählt. In seiner allgemeinsten Form sieht eine Eingabe wie folgt aus

```
bin/SpaceLight search -i my_queries.sdf -f my_fragspace.tfsdb
```

Mithilfe der folgenden Parameter lässt sich die Ähnlichkeitssuche benutzerdefiniert anpassen.

`-i, -input` (notwendig)
Pfad zu einer Datei mit einem oder mehreren Eingabemolekülen. SMILES, SDF und MOL2 Dateien werden unterstützt. Alternativ kann der Nutzer direkt ein Molekül in SMILES Notation und Anführungszeichen in der Konsole angeben.

`-f, -fragspace` (notwendig)
Pfad zu einer Datenbank eines topologischen Fragmentraums.

`-o, -output`
Wenn angegeben, wird in diesem Pfad eine CSV-Datei für die Ausgabe erzeugt. Andernfalls werden die Ergebnisse in der Konsole ausgegeben.

`-d, -descriptor`
Standardmäßig wird die fCSFP2.5 als Methode verwendet. Alternativ kann der Nutzer hier für den ECFP die Parametrisierungen 'ecfp0','ecfp2','ecfp4','ecfp6','ecfp8' oder 'ecfp10' angeben. Für den CSFP sind die Varianten fCSFP, iCSFP und tCSFP verfügbar. Diese können mit 'fcsfp$x.y$', 'icsfp$x.y$' und 'tcsfp$x.y$' spezifiziert werden, wobei x im Intervall $[1;6]$ liegen kann und y im Intervall $[x;6]$.

`-n, -nof`
Standardmäßig wird ein Resultat pro Anfragemolekül generiert. Falls der Nutzer mehr finale Produkte Anfrage wünscht, kann die Anzahl mit diesem Parameter angegeben werden.

`-s, -single`
Standardmäßig kann ein Fragment in mehreren Produkten der Ausgabe für ein Anfragemolekül vorkommen. Ist dies nicht gewünscht, kann mit diesem Parameter erzwungen werden, dass ein Fragment pro Topologiegraph nur in einem Produkt der Ausgabe vorkommt. Fragmente aus unterschiedlichen Topologiegraphen, die den gleichen chemischen Baustein repräsentierten oder aus anderen Gründen strukturell identisch sind, können trotzdem beide in der Ausgabe erscheinen.

C.2.2. Suche nach Fragmenten als molekulare Gerüste

Dieser Modus wird mit dem Parameter `scaffold` ausgewählt. In seiner allgemeinsten Form sieht eine Eingabe wie folgt aus

```
bin/SpaceLight scaffold -i my_queries.sdf -f my_fragspace.tfsdb
```

Mithilfe der folgenden Parameter lässt sich die Gerüstsuche benutzerdefiniert anpassen.

`-i, -input` (notwendig)
Pfad zu einer Datei mit einem oder mehreren Eingabemolekülen. SMILES, SDF und MOL2 Dateien werden unterstützt. Alternativ kann der Nutzer direkt ein Molekül in SMILES Notation und Anführungszeichen angeben.

`-f, -fragspace` (notwendig)
Pfad zu einer Datenbank eines topologischen Fragmentraums.

`-o, -output`
Wenn angegeben, wird in diesem Pfad eine CSV-Datei für die Ausgabe erzeugt. Andernfalls werden die Ergebnisse in der Konsole ausgegeben.

`-d, -descriptor`
Standardmäßig wird die fCSFP2.5 als Methode verwendet. Alternativ sind die CSFP Varianten fCSFP, iCSFP und tCSFP verfügbar. Diese können mit 'fcsfp$x.y$', 'icsfp$x.y$' und 'tcsfp$x.y$' spezifiziert werden, wobei x im Intervall $[1; 6]$ liegen kann und y im Intervall $[x; 6]$. Der ECFP kann hier nicht verwendet werden, da er nicht die Fingerabdruck-Teilmengeneigenschaft erfüllt.

`-n, -nof`
Standardmäßig wird ein Resultat pro Anfragemolekül generiert. Falls der Nutzer mehr finale Moleküle pro Anfrage wünscht, kann die Anzahl mit diesem Parameter angegeben werden.

C.2.3. Erzeugung topologischer Fragmenträume

Dieser Modus wird mit dem Parameter `generate` ausgewählt. In seiner allgemeinsten Form sieht eine Eingabe wie folgt aus

```
bin/SpaceLight generate -i input.json -f custom_fragspace.tfsdb
```

Die chemischen Bausteine und Reaktionen müssen in einer oder mehreren JSON-Dateien spezifiziert werden. Die genauen Angaben wurden in Anhang B.1 erläutert. Eine JSON-Datei kann dabei wie folgt aussehen

```
{"topologies" : [
{
"name": "Example",
"reactions": [
{
"components": [
1,
2
],
"reaction": "[N:1][N:2]=[N;D1:3].[C;D2:4]#[C;D1:5]>>[n:1]1:[n:2]:[n:3]:[c:4]:[c:5]:1"
},
{
"components": [
2,
3
],
"reaction": "[N;!D3:1].[C;$(C(=O)O):2]-[O;D1]>>[N:1]-[C:2]"
}
],
"reagentGroups": [
{
"groupId": 1,
"reagents": "buildingBlocksGroup1.smi"
},
{
"groupId": 2,
"reagents": "buildingBlocksGroup2.smi"
},
{
"groupId": 3,
"reagents": "buildingBlocksGroup3.smi"
}
]}
]}
```

Zusätzlich kann mithilfe des Eintrags `macrocycle` eine Auswahl von Reaktionen getroffen werden, die zusammen eine Makrozyklus erzeugen. Hierbei muss der Nutzer die Reaktion über ihren Index in der Liste `reactions` spezifizieren.

```
{"topologies" : [
{
"name": "MacrocycleExample",
"reactions": [
{
"components": [
1,
2
],
"reaction": "[N;!D3:1].[C;$(C(=O)O):2]-[O;D1]>>[N:1]-[C:2]"
},
{
"components": [
2,
3
],
"reaction": "[N;!D3:1].[C;$(C(=O)O):2]-[O;D1]>>[N:1]-[C:2]"
},
{
"components": [
3,
1
],
"reaction": "[N;!D3:1].[C;$(C(=O)O):2]-[O;D1]>>[N:1]-[C:2]"
}
],
"reagentGroups": [
{
"groupId": 1,
"reagents": "buildingBlocksGroup1.smi"
},
{
```

```
"groupId": 2,
"reagents": "buildingBlocksGroup2.smi"
},
{
"groupId": 3,
"reagents": "buildingBlocksGroup3.smi"
}
]
"macrocycle" : {
"reactions" : [
0,
1,
2
],
"minimalSize" : 9
}}]}
```

Mithilfe der folgenden Parameter lässt sich die Erzeugung benutzerdefiniert anpassen.

`-i, -input` (notwendig)
Pfad zu einer JSON-Datei mit chemischen Bausteinen und Reaktionen. Alternativ kann ein Verzeichnis angegeben werden, dass Unterverzeichnisse mit jeweils einer JSON-Datei und potenziell Dateien für chemische Bausteine enthält. Diese JSON-Dateien werden alle separat prozessiert.

`-f, -fragspace` (notwendig)
Der Pfad in dem die erzeugte Datenbank des topologischen Fragmentraums abgespeichert wird.

C.3. SpaceCompare

Das Software-Modul SpaceCompare umfasst die Implementation der gleichnamigen Schnittmengenberechnung aus Kapitel 5, sowie die Methode SpaceProp zur Bestim-

mung physikochemischer Eigenschaften aus Kapitel 6.3 und die angelehnte Funktionalität zur Eigenschaftsoptimierung aus Kapitel 6.5. Das Modul ist, wie SpaceLight, eine Kommandozeile-Applikation mit drei verschiedenen Modi. Wiederum muss der Nutzer beim ersten Aufruf eine Lizenz angeben mit

```
bin/SpaceCompare --license <license string>
```

Wie bei SpaceLight können für alle drei Modi folgende Parameter optional vom Nutzer verwendet werden.

`-h, -help`
Eine Kurzbeschreibung des Modus und aller verfügbaren Parameter.

`-p, -procs`
Standardmäßig nutzt das Programm alle verfügbaren Prozesse. Mit diesem Parameter kann der Nutzer die gewünschte Anzahl verwendeter Prozesse angeben.

`-u, -userinfo`
Mit diesem Parameter kann der Nutzer die Menge an Informationen beeinflussen, die auf der Konsole ausgegeben wird. Standardmäßig oder bei Stude zwei werden alle Informationen ausgegeben. Bei Stufe eins werden nur Fehlermeldungen und Warnungen ausgegeben. Bei Stufe null werden nur Fehlermeldungen ausgegeben.

`-license`
Mit diesem Parameter kann der Nutzer eine Lizenz angeben. Die Lizenz wird abgespeichert und muss nicht erneut angegeben werden, bis sie abgelaufen ist.

C.3.1. Schnittmengenberechnung

Dieser Modus kann mit dem Parameter `overlap` ausgewählt werden. In seiner einfachsten Form sieht die Eingabe wie folgt aus

```
./SpaceCompare overlap -f space1.tfsdb space2.tfsdb -o out.csv
```

Der Nutzer kann die Berechnung mit folgenden Parametern beeinflussen.

`-f, -fragspace` (notwendig)

Die Dateipfade zu zwei Datenbanken topologischer Fragmenträume deren Schnittmenge berechnet wird.

`-o, -output` (notwendig)
Pfad an dem eine CSV-Datei mit allen Produkten der berechneten Schnittmenge abgespeichert wird.

`-r, -reorder`
Mit diesem Parameter kann der Nutzer die Reihenfolge beeinflussen, in der die zwei Räume prozessiert werden.
0: Reihenfolge wie vom Nutzer angegeben mit –fragspace oder -f
1: Reihenfolge nach absteigender Anzahl implizit beschriebener Produkte (Standard).
Bei Laufzeiten von mehr als drei Tagen oder hohem Speicherbedarf kann eine Änderung der Reihenfolge potenziell helfen. Der Nutzer sollte hierbei beide Reihenfolgen ausprobieren, wenn unklar ist welcher topologische Fragmentraum mehr Produkte enthält.

C.3.2. Verteilungsberechnung

Dieser Modus kann mit dem Parameter `histogram` ausgewählt werden. Eine Eingabe sieht wie folgt aus

```
./SpaceCompare histogram -f my_space.tfsdb -o out_dir
```

Der Nutzer kann die Berechnung mit folgenden Eingaben parametrisieren.

`-f, -fragspace` (notwendig)
Pfad zu einer Datenbank eines topologischen Fragmentraums.

`-o, -output` (notwendig)
Ein Verzeichnis in dem die generierten Verteilungen für den aLogP, das Molekulargewicht, die Anzahl potenzieller Protonenakzeptoren und -donatoren einer Wasserstoffbrückenbindung sowie die Anzahl Schweratome in fünf CSV-Dateien abgespeichert werden.

`-a, -approx`
Mit diesem Parameter kann die approximative Variante von SpaceProp aus Kapitel 6.3.3 ausgewählt werden, um die Laufzeit des Verfahrens zu verkürzen.

0: Die exakte Berechnung wird verwendet. (Standard)
1: Die approximative Variante wird verwendet.

C.3.3. Eigenschaftsoptimierung

Dieser Modus wird mit dem Parameter `optimize` ausgewählt. Eine Eingabe hat die Form

```
./SpaceCompare optimize -f my_fragspace.tfsdb
-o optimized_space.tfsdb -t logp -r 0.3
```

Die folgenden vier Parameter müssen alle vom Nutzer spezifiziert werden, um einen optimierten topologischen Teilraum zu erzeugen.

`-f, -fragspace` (notwendig)
Pfad zu einer Datenbank eines topologischen Fragmentraums, dessen Eigenschaftsverteilung optimiert werden soll.

`-o, -output` (notwendig)
Der Pfad an dem der erzeugte optimierte Teilraum als Datenbank abgespeichert wird.

`-t, -type` (notwendig)
Die physikochemische Eigenschaft, dessen Verteilung optimiert werden soll. Der Nutzer kann eine der folgenden fünf Eingaben wählen.
`logp`: aLogP
`weight`: Molekulargewicht
`acceptor`: Anzahl potenzieller Protonenakzeptoren
`donor`: Anzahl potenzieller Protonendonatoren
`heavyatoms`: Anzahl Schweratome

`-r, -remove` (notwendig)
Anteil der Fragment mit den höchstens Eigenschaftswerten, die aus dem Raum entfernt werden sollen. Wert muss im offenen Intervall $(0; 1)$ liegen.

D. Publikationen der kumulativen Dissertation

D.1. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING
Cite This: J. Chem. Inf. Model. 2019, 59, 4625−4635
Article
pubs.acs.org/jcim

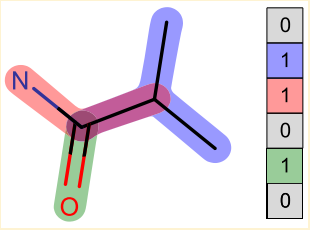
Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration

Louis Bellmann, Patrick Penner, and Matthias Rarey*

Universität Hamburg, ZBH − Center for Bioinformatics, Research Group for Computational Molecular Design, Bundesstraße 43, 20146 Hamburg, Germany

Supporting Information

ABSTRACT: Molecular fingerprints are an efficient and widely used method for similarity-driven virtual screening. Most fingerprint methods can be distinguished by the class of structural features considered. The Connected Subgraph Fingerprint (CSFP) overcomes this limitation and regards all structural features of a compound. This results in a more complete feature space and high adaptive potential to certain application scenarios. The novel descriptor surpasses widely used fingerprint methods in some cases and opens the way for topological search in combinatorial fragment spaces.



INTRODUCTION

In similarity-driven virtual screening (VS) potentially bioactive compounds are extracted from a set of molecules based on their similarity to a known active compound following the similarity principle.[1] For this process a molecular descriptor and a similarity function are required. Many other problems besides screening can be addressed in this fashion, for example clustering of compound data sets[2] or common scaffold detection.[3] For several applications, the choice of the molecular descriptor is not straightforward, and there exist a vast amount of methods.[4−6] Most molecular fingerprints capture certain chemical substructures of a compound. These substructures are then represented as bits in a bit string, counts in a count vector, or as a set of numerical identifiers. This abstract representation in a linear feature space enables efficient similarity calculation and diversity analysis[7] and its use in all kinds of machine learning applications.

In this work we will refer to two distinct definitions resembling chemical substructures, *structural atom features* and *structural bond features* related to the node and edge-based versions of the maximum common subgraph (MCS) problem.[8,9] Structural atom features consist of a set of atoms as well as all bonds between the atoms therefore representing an induced subgraph. Structural bond features consist of a set of bonds as well as all underlying atoms. Note that these two definitions yield almost the same set of chemical substructures of a compound. In cases of chemical substructures containing some parts, but not all atoms and bonds of a ring, the two definitions can lead to different structural features.

A common approach for the generation of topological fingerprints consists of three steps: enumerating a defined subset of structural atom or bond features, assigning numeric identifiers to each atom of the molecule, and finally combining atom identifiers in a unique way.[10] This results in a set of numeric identifiers that each represents a certain feature existent in the compound. Often fingerprints are distinguished by the family of structural features they consider. Some only consider pathlike[11,12] or circular[13,14] structures while others use a dictionary-based[15] or pharmacophoric[16] approach. There are also fingerprint methods in existence that capture combinations of some families of structural features.[17] Most widely used today are Extended Connectivity Fingerprints (ECFPs)[13] which consider circular structures. This restriction reduces the quantity of structural properties captured by the fingerprint.

In some cases this restriction might even lead to functional groups being neglected that are defining for the drug class of the compound. In Figure 1 sulfamerazine is shown with a sulfonamide group in its center. No features generated by the ECFP represent this group. Each feature contains a center atom together with all heavy atoms reachable from the center atom traversing at most a fixed number of bonds. As a result, all of these features either miss at least one atom of the group or contain one not belonging to it. In conclusion, all of the methods mentioned above do not describe molecules using all of their structural features which can lead to important properties of a compound being neglected. The RDKit fingerprint,[18] inspired by the Daylight fingerprint,[19] overcomes this limitation and encodes all structural bond features of a compound up to a given size.

A complete consideration of the structural feature space produces a "denser" fingerprint meaning it contains more identifiers and enables a fine-grained distinction of very similar compounds. Additionally it allows the usage of the fingerprint method in an MCS-like[9] scenario. Since each structural feature is present, the appearance and absence of a feature in a

Received: July 12, 2019
Published: October 25, 2019

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

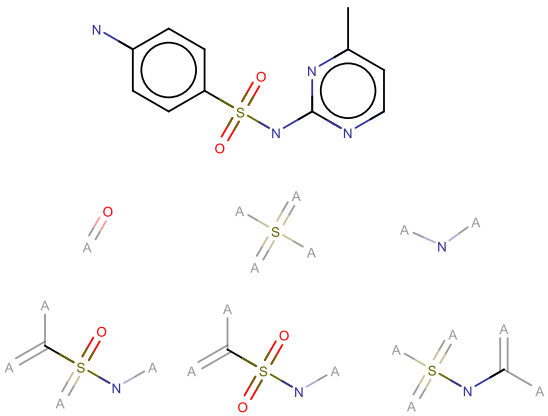


Figure 1. Sulfamerazine and the structural features around the sulfonamide group generated by the ECFP. The literal 'A' for any heavy atom together with the adjacent bonds in gray cover the information captured in the features generated by the ECFP.

compound or data set can both be fully described, resulting in a good interpretability of the fingerprint. Additionally the resulting method might be beneficial for clustering and machine learning methods due to the feature richness. To cover a bigger percentage of the chemical space upon virtual screening, large databases are a necessity. Fragment spaces can surpass noncombinatorial databases in the number of compounds contained by several orders of magnitude while requiring only a manageable amount of storage space.[20,21] However, common molecular descriptors can be inappropriate for the application to fragment spaces, and new methods tailored to the special conditions present in combinatorial libraries of small fragments are highly desirable.[22,23] We will show that the limitation to a structural feature family can be problematic for a fingerprint method when applied to fragments.

In this paper, we present a new topological fingerprint named CSFP. In contrast to existing approaches, the fingerprint precisely covers all structural atom features within a predefined size range and uniquely represents each of them with an identifier. We demonstrate that it is in fact superior to cover the feature space completely in several fingerprint use cases. Furthermore, we discuss that this fingerprint is suited for topological similarity searching in large combinatorial fragment spaces.

METHODS

In the following, we describe a novel fingerprint method as well as five different variations which are the product of different filter criteria and atom properties. The *Connected Subgraph Fingerprint* (CSFP) does not restrict itself to a certain family of substructures but instead enumerates all structural atom features with a number of heavy atoms contained in a given interval. This leads to a much denser fingerprint and some completeness properties which can have various advantages.

CSFP Algorithm. We start with a brief overview of the novel method followed by an in-depth description using concrete examples. The CSFP algorithm consists of three steps:

1. An *enumeration step* in which all connected substructures, or structural atom features, of the molecule are enumerated. The only constraint configurable by the user are a lower and an upper bound on the number of heavy atoms they can contain. Additionally any filter criteria can be applied to the features to achieve desired properties.
2. A *unification step* well-known from the creation of unique molecule identifiers where each substructure is identified with an integer that is generated in three steps:
   2.1. Identify each heavy atom contained in the subset with a numeric identifier which represents certain properties of the atom.
   2.2. Derive a strict unique ordering of the heavy atoms based on their identifiers following the CANON[24] procedure.
   2.3. Traverse the substructure applying a depth-first search (DFS), prioritizing atoms with a lower index in the ordering and combining the atom identifiers with the bond identifiers in the order that they are visited in to form a single numeric identifier representing the substructure.
3. A *duplicate identifier removal step* in which substructures with the same identifier are summarized. The existence of the identifier in the final fingerprint is equivalent to the corresponding structural feature being present in the compound. Alternatively the occurrences of an identifier can be counted instead.

In the following we describe the steps of the algorithm in detail. We want to stress that the choice of interval and filter criteria in the *enumeration step*, as well as the chosen atom properties in the *unification step*, can be chosen differently and can drastically influence the performance of the fingerprint in a given application scenario.

1. Enumeration Step. Consider a compound and an arbitrary numbering of its heavy atoms. In this step all distinct sets of numbered heavy atoms are generated that induce a connected substructure of the compound and fulfill the specified filter criteria. Note that chemical substructures, e.g., functional groups, can have multiple occurrences in a given compound. Since the underlying numbered atoms form a different set of integers, they are treated as distinct objects in this step. Apart from size constraints for the number of heavy atoms, one other filter condition is used in this paper, namely restricting the subsets to form paths resulting in a variant of the fingerprint we name pCSFP. In Figure 2 we see the result

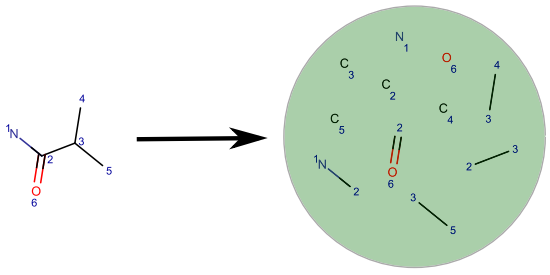


Figure 2. All numbered heavy atom sets with one or two heavy atoms of isobutyramide.

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

of the enumeration step when applied to isobutyramide with set sizes of one and two heavy atoms. For enumerating all substructures we developed a novel algorithm named CONSENS described in detail in the Supporting Information. Briefly, the atom sets are derived by repeatedly adding atom neighbors to the sets following the order given by the numbering of the heavy atoms. To avoid duplicates each enumerated set holds a set of forbidden and candidate atoms. The algorithm starts with all sets containing a single atom. For each of these singletons all atoms of lesser order are forbidden atoms, and all nonforbidden atom neighbors form the set of candidates. In each step one candidate is added to the set. Now all candidates of lesser order than the chosen one are added to the set of forbidden atoms. All atom neighbors of the chosen candidate that are not forbidden or contained in the atom set are added to the set of candidates. This procedure is applied using all singletons of the compound and all possible choices of candidates in each growing step. In Figure 3 two steps of the CONSENS algorithm are described for isobutyramide. Initially the enumerated atom set contains only the carbon atom with the number two. Then the candidate carbon atom with the number three is added, and the candidate set is updated. Next the new candidate carbon atom with the number five is added to the set, and the candidate carbon atom of lesser order is added to the set of forbidden atoms. For the pseudocode of the enumeration procedure and proof of correctness please refer to the Supporting Information.

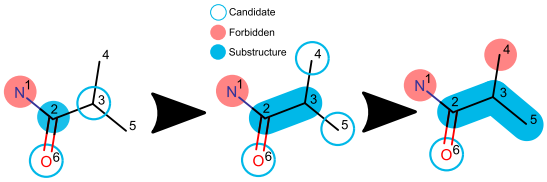


Figure 3. Two steps of the CONSENS algorithm for isobutyramide. The currently considered substructure for each step is colored in turquoise, all its candidate atoms are encircled in turquoise, and its forbidden nodes are colored in red.

2. Unification Step. The following procedure is applied to every set of heavy atoms that was enumerated in the previous step independently. To simplify the description of the algorithm only a single set S is considered for the following three substeps.

2.1. First, numeric identifiers representing atom properties are generated for each heavy atom contained in S. Here any properties that can be represented by integers and that are independent of the input order of the atoms can be used. In the versions selected for the evaluation the identifiers contain information about the atom itself and its connectivity in the molecule, as well as information about its environment. These properties are similar to the ones chosen for the ECFP, which mostly represent the Daylight invariants.[19] These properties are characterized by a string of integers that in turn is hashed to a single numeric identifier.

2.2. Second, the atoms contained in S are uniquely ordered using the numeric identifiers generated in the previous step. The problem of finding a strict unique ordering of the nodes of a labeled graph is a well-discussed topic[25,26] and closely related to the graph isomorphism problem.[27,28] Molecules can be uniquely represented by a SMILES string using the CANGEN[24] method. The procedure contains the CANON subroutine which derives symmetry classes based on a linear combination of atom invariants. Initially all atoms are ranked using the linear combination of invariants. If at least two atoms rank equivalently, each atom is assigned the product of corresponding primes of all its atom neighbors' ranks. The ranking of atoms is repeated using these products until the ranking does not change anymore. If two atoms still inhabit the same rank, a tie breaking is applied, and the previous step of corresponding prime products is repeated. The resulting ordering considers the specified invariants of the atoms as well as the underlying graph structure and possible symmetries. We follow the CANON procedure but rank the atoms by their numeric identifiers from the previous step.

2.3. Finally, a single numeric identifier representing the set S is derived. To achieve this we traverse S in a depth-first search starting with the atom of lowest rank with respect to the derived order from the previous step. The identifier of said atom functions as a seed. During the depth-first search atoms with lower rank are prioritized. If multiple atoms have the same lowest rank due to symmetry in the graph, atoms connected via a single bond are considered first. The type of traversed bond together with the identifier of the currently considered atom is combined with the seed. Additionally, bonds that were not used by the depth-first search procedure are combined with the seed when both adjacent atoms have been visited. This way the identifier accounts for ring closures.

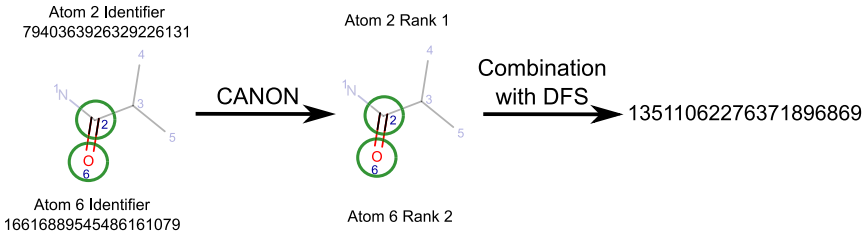


Figure 4. Unification step applied to the numbered heavy atom set of isobutyramide forming a carbonyl functional group which was derived in the enumeration step.

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

In summary, this procedure ensures that the numeric identifier of S holds information about the invariants of all heavy atoms contained in S as well as the underlying molecular graph restricted to S. In Figure 4 the unification step is described for the carbonyl group contained in isobutyramide. The identifier assigned to the carbon atom in step 2.1 is smaller than that assigned to the oxygen atom, so the carbon is ranked lower by the CANON routine in step 2.2. In step 2.3 the depth-first traversal starts at the carbon atom and combines its identifier with that of the oxygen atom.

3. Duplicate Identifier Removal Step. Up to this point, each of these atom sets is now represented by a numeric identifier encapsulating the molecular graph and the corresponding atom properties. The resulting set of identifiers can be unified to achieve an unfolded binary fingerprint. Alternatively a counted version of the fingerprint is analogously possible where the number of occurrences of each identifier is tracked. For later evaluation we restrict ourselves to the simple binary fingerprint.

Fingerprint Variations. In the following we briefly describe five CSFP variations. They differ only in the filters applied to the subgraphs generated in the enumeration step as well as the atom properties chosen in the step 2.1. The choice of atom properties for a certain application scenario is not straightforward. We included three properties in all fingerprint variations, the atomic number, the connectivity, and the valence since they capture fundamental characteristics of the atoms described. The way in which valence and connectivity are represented can differ between fingerprint variations.

CSFP. This is the standard version of the CSFP, and the only filter applied to the subgraphs is a lower and an upper bound for the number of heavy atoms x and y. To indicate the bounds chosen, we will specify them in form CSFP$x.y$, for example the CSFP3.7 considers all connected subgraphs of a compound with a number of heavy atoms between three and seven. The initial atom identifiers account for seven properties: the number of heavy atom neighbors to the atom in the currently considered subgraph, the number of heavy atoms connected to the atom in the whole compound, the valence of the atom considering all non-hydrogen bonds at the atom, and valence only considering bonds connecting the atom to another one contained in the subgraph. These two valence identifiers are derived as the sum over all identifiers of considered bonds, where single, double, and triple bonds have the bond identifier two, four, and six, respectively, and an aromatic bond has the identifier three. Additionally the atomic number, whether or not the atom is contained in a ring in the whole compound, and a stereoidentifier are included as well.

Stereoidentifier. The CSFP includes stereoinformation in the atom identifiers by using an integer following the CIP-rules.[29] The integer is zero if the atom is not a stereocenter, and one if it is a stereocenter but not specified in the input data describing the molecule. If it is a stereocenter in R or S form, the integer is two or three. If the atom is part of a bond in E or Z form, the integer is four or five. When applying the fingerprint to similarity search in fragment spaces, it is important to take into consideration that the stereochemistry of building blocks can change upon combination to a product.

iCSFP. This is the independent CSFP version specified by the prefix i. The identifiers of the iCSFP depend only on structural properties of the structural features they are associated with neglecting the environment of the feature. The subgraphs considered are again filtered by a lower and an upper bound with nomenclature of the form iCSFP$x.y$ indicating a lower bound x and an upper bound y. The atom identifiers account for three of the seven atom properties of the CSFP: the number of neighbors in the currently considered subgraph, the valence only considering bonds inside the subgraph, and the atomic number. The derived identifiers of the iCSFP can therefore be used to encode the existence of a functional group in a compound disregarding the surrounding environment.

pCSFP. This is the path-based version of the CSFP denoted by the prefix p. The subgraphs generated in the enumeration step are filtered by the aforementioned size constraints as well as the condition of not being branched and not containing any rings.We specify them in the form pCSFP$x.y$. The atom properties considered are the same as in the standard CSFP version.

gCSFP. This version of the CSFP captures less information about the structural atom features, is in turn more generic than the standard CSFP, and is specified by the prefix g. The initial atom identifiers account for three atom properties: the number of heavy atom neighbors in the whole compound, the atomic number, and the valence considering all non-hydrogen bonds at the atom. The valence identifier is derived analogously to the CSFP except when the atom is contained in an aromatic ring. Then the valence identifier is zero. This way aromatic and aliphatic atoms are differentiated, and all aromatic atoms get the same valence identifier assigned. Additionally the bond identifiers are not considered in the unification step 2.3 and are contained in the resulting substructure identifier only implicitly via the initial atom identifiers. In a similar fashion to the above version, we specify the bounds chosen with gCSFP$x.y$.

tCSFP. The last version of the CSFP is associated with the TopologicalTorsion[12] fingerprint and denoted by the prefix t. It considers the same atom properties as the TopologicalTorsion fingerprint: the number of heavy atom neighbors in the compound, the atomic number, and its number of π electron pairs. Same as for the gCSFP, the bond identifiers are not considered in step 2.3 of the CSFP algorithm. The topological version of the CSFP differs from the TopologicalTorsion fingerprint by considering all structural atom features in the given size interval which we denote again with tCSFP$x.y$ as opposed to the pathlike structures of length four considered by the TopologicalTorsion fingerprint.

In summary, the iCSFP differs from the standard CSFP by only holding information about the structural atom features independent of the surrounding environment in their identifiers. The gCSFP and tCSFP hold less information about the structural feature in general compared to the CSFP. The pCSFP reduces the number of structural atom features by topological filters applied.

RESULTS

We assessed the performance of the CSFP in different varieties using the open-source platform developed by Riniker and Landrum[30] which provides fingerprint implementations in RDKit[18] for benchmarking novel fingerprint methods. Apart from the CSFP and its variations, we consider the ECFP and RDKit fingerprint for comparison. Since the ECFP is a widely used fingerprint method, many implementations and variations exist.[17,31,32] We use the RDKit implementation of the ECFP because it was used in the paper by Riniker and Landrum.

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

Table 1. Fingerprint Representatives for Each Data Set and AUC and the Enrichment Factor at 2%[a]

	AUC			EF 2%		
	fingerprint	av score	SD	fingerprint	av score	SD
MUV	tCSFP3.5	0.671	0.102	pCSFP2.8	7.293	6.556
	gCSFP3.5	0.647	0.110	CSFP4.4	7.098	6.076
	CSFP2.3	0.647	0.116	gCSFP10.10	7.000	7.196
	pCSFP2.3	0.647	0.116	ECFP4	6.924	5.848
	RDK6	0.628	0.098	iCSFP4.8	6.866	7.117
	iCSFP4.8	0.607	0.116	tCSFP2.6	6.683	5.836
	ECFP4	0.601	0.128	RDK5	6.608	6.455
DUD	gCSFP2.5	0.900	0.090	pCSFP2.10	29.887	8.710
	tCSFP5.8	0.898	0.090	ECFP4	29.831	8.283
	iCSFP5.5	0.898	0.094	CSFP1.5	29.730	8.352
	CSFP2.3	0.890	0.105	iCSFP8.10	29.651	8.696
	pCSFP2.3	0.890	0.105	gCSFP2.7	29.476	8.180
	ECFP10	0.888	0.117	tCSFP3.10	29.083	8.287
	RDK5	0.884	0.096	RDK5	28.591	8.426
ChEMBL	tCSFP6.8	0.814	0.107	tCSFP8.10	22.173	10.087
	gCSFP5.5	0.807	0.108	CSFP9.10	21.942	10.159
	RDK6	0.804	0.117	gCSFP9.10	21.852	10.178
	CSFP3.4	0.789	0.112	pCSFP3.7	21.686	10.361
	pCSFP3.3	0.787	0.113	ECFP4	21.554	10.510
	iCSFP6.6	0.785	0.111	iCSFP9.10	21.197	10.094
	ECFP4	0.762	0.131	RDK5	19.214	9.692

[a]The representatives are sorted by decreasing average score and increasing standard deviation over all targets for that data set. The CSFP derivatives can be identified using their prefix: i stands for the independent version, p stands for the pathlike version, g stands for the generic version, and t stands for the version following the atom properties of the TopologicalTorsion fingerprint.

The platform provides four different evaluation methods. Here we use two of them: the area under the curve (AUC) of the ROC-curve as well as the enrichment factor (EF) at 2%. Riniker and Landrum showed that all "early enrichment" methods were highly correlated and therefore recommended restricting the range of methods to the enrichment factor and AUC for clarity.[30] The AUC metric evaluates the general performance of a VS method and not the early enrichment of actives. So fingerprint methods performing well when evaluated with AUC do not necessarily need to do so for the enrichment factor as well. In fact Riniker and Landrum showed that the two metrics can yield a very different ranking of fingerprint methods. There is a rich literature on multicriteria optimization which can be used to find a fair comparison of VS methods across a range of evaluation metrics.[33,34] In this work however we want to emphasize the complementary character of the two evaluation metrics at hand, in order to derive suitable fingerprint methods for different VS tasks. The performance difference of two fingerprint methods on a given data set using one of the evaluation metrics is called significant if the p-value of the corresponding Wilcoxon signed-rank[35] test is less than 0.05. The test was carried out using the scores of all repetitions per target on the data set at hand. Note that the null hypothesis of the Wilcoxon test states that the average scores per target of the two fingerprint methods come from the same normal distribution. Therefore, a low p-value does not indicate a better performance of one of the methods on the whole data set but instead implies that the distribution of average scores of the fingerprint methods are significantly different. This suggests that the two methods capture significantly different properties of the compounds contained in the data set.

Data Sets. The platform uses three different data sets for fingerprint evaluation. Riniker and Landrum found that the performance of fingerprints was highly dependent on the data set and target chosen. The maximum unbiased validation (MUV) data set[36] was designed for evaluation of fingerprint methods for ligand-based virtual screening and consists of 17 targets each with 30 actives and 15000 decoys. The directory of useful decoys (DUD)[37] was originally designed for benchmarking of structure-based virtual screening methods. Riniker and Landrum extracted a subset of 21 targets from DUD with more than 30 actives each for ligand-based virtual screening.[30] We want to note that the actives/decoys ratio varies between DUD targets which is important to be aware of when using the enrichment factor as an evaluation metric. The third data set of the benchmark platform contains 50 targets with 100 actives derived from ChEMBL[38] together with 10000 compounds extracted from ZINC[39] functioning as decoys for all of the 50 targets. Riniker and Landrum found that all fingerprint methods tested performed significantly better on the DUD data set, and the performance of fingerprints on the MUV data set was generally worse than on the other two. This might be caused by the fact that the MUV data set was designed to be hard for ligand-based VS methods, whereas the DUD data set was originally only designed for structure-based methods. Hence topological fingerprint methods can exploit potential bias in the data set.[40]

Benchmarking Process. The benchmarking process is fully automated and made available as Python scripts.[30] The activity prediction experiment is repeated 50 times for each target. For each repetition 20% of the decoys and five actives were randomly chosen and excluded beforehand by the authors of the platform. These five excluded active compounds are then used as queries. The remaining compounds for the target are ranked based on the highest Tanimoto score for the five queries following the MAX fusion method.[41] This results in a ranked list for each target,

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

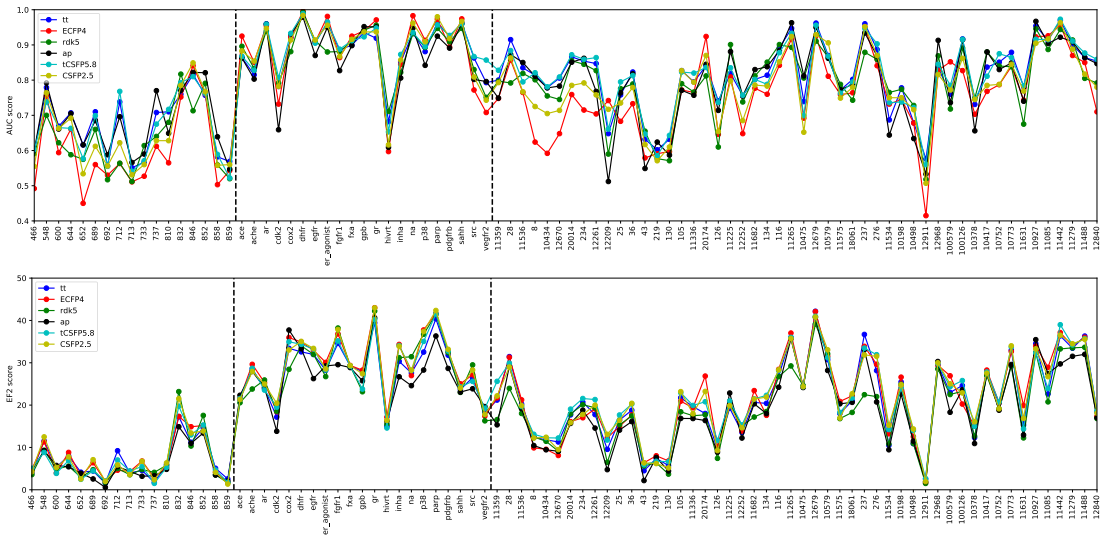


Figure 5. Performance of the five fingerprint methods TopologicalTorsion, ECFP4, RDK5, AtomPairs, tCSFP5.8, and CSFP2.5 on all targets of the benchmark platform when measured with AUC (top) and the enrichment factor at 2% (bottom). The *x*-axis accounts for the targets across the three data sets.

repetition, and fingerprint. The performance is then measured by AUC and the enrichment factor individually.

Parameter Selection. In order to find the best parameters for all fingerprint variants, the following process is done for each of the three data sets individually. To this end the benchmarking process is done separately for the ECFP, RDKit Fingerprint, CSFP, iCSFP, pCSFP, gCSFP, and tCSFP. For the ECFP the parametrized versions ECFP0, ECFP2, ECFP4, ECFP6, ECFP8, and ECFP10 are considered. For the RDKit fingerprint the upper bounds four, five, and six for the number of bonds in the substructures are considered as they are part of the benchmark platform.[30] For the CSFP all lower bounds x from one to ten and upper bounds y from x to ten where assessed. The different parameters are scored over all targets of the data set and all repetitions using the two evaluation methods AUC and EF at 2%. For each fingerprint the mean AUC and EF at 2% over all 50 repetitions for each target are derived. Ultimately the parameters with the highest mean score and lowest standard deviation for the data set are chosen as a representative for that fingerprint and data set. This process terminates with one chosen representative for each variation of the CSFP and one representative for the ECFP and RDKit fingerprint for each evaluation method and data set.

Scenario Specific Analysis. In Table 1 we see the representatives chosen for each of the two evaluation methods and each of the three data sets. We evaluate the performance of the fingerprint methods separately for each data set and evaluation metric to showcase that the superiority of a fingerprint method over another method is highly dependent on the data set and evaluation metric. The representatives are sorted by decreasing average score and standard deviation, meaning that the first representative in the list performs best on average for that data set and evaluation method. The pairwise correlation plots and Wilcoxon singed-rank test results are given in Figures S1−6 and Tables S1−6. We can see that the gCSFP and tCSFP perform well when evaluating with AUC and the pCSFP performs well when evaluating with the enrichment factor. The CSFP inhabits the second or third place in most scenarios. The ECFP is ranked last or second to last on every data set for the AUC evaluation metric but performs better when evaluating with the enrichment factor. Conversely the RDKit fingerprint inhabits the last rank on each data set for the enrichment factor and is ranked higher when evaluating with the AUC metric. Most fingerprint representatives differ significantly from one another on each data set and for both evaluation metrics. This implies that the methods describe different properties of the compounds contained in the data sets. In general, the representatives for the enrichment factor incorporate larger upper bounds than those for the AUC on the same data set and consequently contain larger structural features. Although most pairs of representatives differ significantly following the Wilcoxon test, we can infer that the average performance of most fingerprints on most data sets is highly correlated and that performance between targets and data sets generally differs much more than performance between fingerprints on one target. These findings coincide with those of Riniker and Landrum.[30] The CSFP, tCSFP, and gCSFP only differ in the atom and bond properties captured by structural features. The tCSFP and gCSFP representatives are ranked higher than the CSFP on every data set when applying the AUC evaluation metric. This indicates that fingerprint methods with more generic features can outperform more sensitive methods in certain application scenarios. The CSFP and pCSFP differ in the restriction onto pathlike structural features. This seems to be beneficial in some cases as the pCSFP outperforms the CSFP on the MUV and DUD data set when measuring with the enrichment factor. A direct comparison to the restriction onto the circular structural family considered by the ECFP cannot be made. This is due to the fact that the features considered by the ECFP for a fixed diameter, e.g., four, can contain any number of heavy atoms from three to 17.

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

General Performance. So far we have seen that the CSFP and its variations are highly adaptable due to the lower and upper bounds for the structural features considered and that some variations perform better than others depending on the data set and evaluation metric. Now we want to pick variations and parameters that perform well in a general scenario. Therefore, we seek fingerprint methods and configurations that have a high average score on all three data sets when measuring with AUC and the enrichment factor. The benchmarking done by Riniker and Landrum showed good performance of the AtomPairs and TopologicalTorsion fingerprint, so we rerun the benchmark process additionally including the ECFP4, RDK5, tCSFP5.8, and CSFP2.5 which had on average a good performance on all data sets and both evaluation methods relative to the other parametrizations for their CSFP variation.

The results for the AUC and enrichment factor at 2% per target are shown in Figure 5. The average scores over all targets are shown in Table 2. The pairwise correlations and

Table 2. Average Scores and Standard Deviation for the TopologicalTorsion, ECFP4, AtomPairs, RDK5, tCSFP5.8, and CSFP2.5 Fingerprint Evaluated with AUC and the Enrichment Factor at 2%[a]

AUC			EF 2%		
fingerprint	av score	SD	fingerprint	av score	SD
tCSFP5.8	0.804	0.129	ECFP4	20.703	11.947
TopoTorsion	0.803	0.126	CSFP2.5	20.686	11.819
AtomPairs	0.791	0.133	tCSFP5.8	20.629	11.712
RDK5	0.783	0.137	TopoTorsion	20.004	11.513
CSFP2.5	0.780	0.139	RDK5	19.017	11.402
ECFP4	0.761	0.158	AtomPairs	18.486	11.258

[a]The methods are sorted by decreasing average score.

Wilcoxon signed-rank tests for both metrics are shown in Figures S7−8 and Tables S7−8. Generally we can see that the difference in performance is mainly dependent on the target, not the fingerprint method, although most fingerprint pairs differ significantly for both metrics according to the Wilcoxon signed-rank tests. We can see that TopologicalTorsion and AtomPairs fingerprint methods perform well on average when measuring with AUC but fall behind in performance when using the enrichment factor as a metric. The reverse is true for the ECFP. This was also found by Riniker and Landrum. The ECFP inhabits the first rank for the enrichment factor metric, however not differing significantly from the CSFP and tCSFP. The RDKit fingerprint shows a similar performance to that of the AtomPairs fingerprint but differs significantly from it for both evaluation metrics following the Wilcoxon test. Since the tCSFP and the TopologicalTorsion fingerprint capture the same atom properties, it is not surprising that they perform similarly. However, the tCSFP slightly outperforms the ToplogicalTorsion fingerprint and differs significantly from it following the Wilcoxon test when measuring with the enrichment factor. This is caused by the different structural features considered. The TopologicalTorsion fingerprint only considers paths with four heavy atoms, and the tCSFP5.8 considers all structural features with five to eight heavy atoms. Here we see that the restriction to one structural feature class or number of atoms can reduce the performance of a fingerprint method. The CSFP captures more atom properties than the tCSFP and is more specific as a result. Here we can see again that more generic fingerprint methods tend to perform better when measuring with AUC but fall behind when applying the enrichment factor. The two CSFP variations were picked to perform well on average for both metrics, but we can conclude that the tCSFP should be favored when looking for a high AUC score and the CSFP when a good early enrichment is desired.

During the benchmarking process the CSFP and its variants took a computation time comparable to or lower than that of the ECFP6 around 150 μs per compound when the upper bound was not bigger than three. In general, the computation time of the CSFP grows approximately by a factor of two when increasing the upper bound for the number of heavy atoms by one. The computation of the CSFP1.10 took around 15 ms per molecule.

We have shown that the CSFP and the considered variations can compete with and in some cases surpass well-known fingerprint methods. They also describe significantly different molecule properties. Furthermore, the method is highly adaptable to specific scenarios. Generally no best method for each VS task was found which coincides with earlier findings, and the choice of the fingerprint method together with a suitable parametrization should always depend on the application scenario at hand. However, the difference in performance between targets was usually much bigger than the difference between fingerprint methods for one target. Larger structural features with more atom properties captured support an early enrichment of actives. Smaller structures considering less atom properties perform better when measuring the general VS performance with the AUC metric. Riniker and Landrum found a correlation between general VS performance and "scaffold hopping" potential on the three data sets presented here. This suggests a suitability of fingerprint methods considering smaller structural features and only a small range of atom properties for scaffold hopping tasks. This behavior is not surprising since early enrichment is dominated by highly similar molecules. Nevertheless, it demonstrates the capabilities to fine-tune the fingerprint quite well.

In the following we want to discuss useful properties of the novel fingerprint method which makes it, apart from the previously shown good VS performance, an interesting alternative to existing methods.

Fingerprint Behavior on Fragments. For the application of a fingerprint method in the context of combinatorial fragment spaces, the relationship between the fingerprint of a building block and the fingerprint of a derived product is of great importance. If the identifiers contained in the fingerprint of the building block form a subset of the identifiers in the fingerprint of the product, it is possible to approximate it by the fingerprints of all its building blocks and consequently avoid combinatorial explosion. We call this property of a fingerprint method the *fingerprint subset relation* in the following.

Structural Feature Subset Relation. In order to fulfill the fingerprint subset relation between a building block and a product, a fingerprint method must also obey the same subset relation on the underlying considered structural features of the building block and the product. We call this relation the *structural feature subset relation*. If a building block is a chemical substructure of a product in the atom-based structural feature definition, the building block and all its structural atom features are in turn also contained in the set of

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

structural atom features of the product. Since the CSFP, iCSFP, tCSFP, and gCSFP consider all structural atom features of the product, the structural feature subset relation holds true. Analogously the structural feature subset relation holds true for the RDKit fingerprint if the building block is a chemical substructure of the product in the bond-based structural feature definition. The pCSFP fulfills the structural feature subset relation as well since all pathlike structural features of a chemical substructure in the atom-based structural feature definition are in turn a pathlike structural feature of a compound containing the chemical substructure. The ECFP does not obey the structural feature subset relation and consequently also not the one for its fingerprints. In Figure 6 acetylcysteine as a product of cysteine and an acetyl

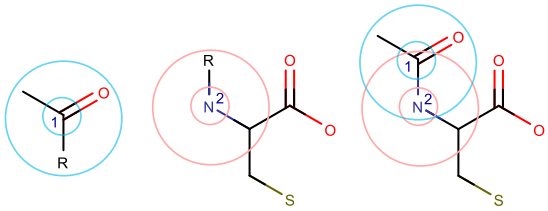


Figure 6. Some features generated by the ECFP2 for the acetyl group (left), cysteine (middle), and acetylcysteine (right). The acetyl group and the cysteine resemble building blocks of the acetylcysteine as their product, and the attachment point of the two building blocks is depicted as an R-group. The carbon and nitrogen atom at the attachment point function as centers for the structural features considered here.

group is depicted. The structural features considered by the ECFP2 at the attachment point of the two building blocks are shown before connecting the building blocks and in the derived product. The circular feature containing all heavy atoms connected via at most one bond to the labeled carbon atom differs in the acetyl group and the acetylcysteine. In the acetyl group it contains the labeled carbon itself and the connected carbon and oxygen atom together with the connecting bonds. This circular structural feature is not contained in the set considered by the ECFP2 for the acetylcysteine. The corresponding structural feature additionally contains the nitrogen atom of the cysteine. Similarly the circular structural feature containing all heavy atoms connected via at most one bond to the labeled nitrogen atom differs in the cysteine and the acetylcysteine. In the product the carbon atom of the acetyl group is contained in the structural feature, whereas it is not contained in the structural feature of the cysteine.

Fingerprint Subset Relation. We have seen that fingerprint methods which fulfill the fingerprint subset relation also obey the structural feature subset relation. The converse is not necessarily true. In Figure 7 propane, 2-methylpropane, and a

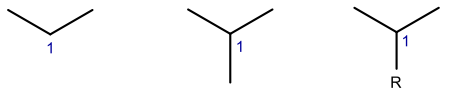


Figure 7. Propane (left), 2-methylpropane (middle), and a propyl group with an R-group attached at the central carbon atom (right). For each compound the central carbon atom is labeled with label one.

propyl group with an R-group attached to the central carbon atom are depicted. For the iCSFP and RDKit fingerprint the identifiers contained in the fingerprint of the propane form a subset of the identifiers contained in the fingerprint of the 2-methylpropane. For the CSFP, tCSFP, gCSFP, and pCSFP this is not the case. The propane forms a structural feature of the 2-methylpropane in the atom and bond definition. Since the identifiers of the iCSFP and RDKit fingerprint only capture atom properties independent of the environment of the structural feature, the fingerprint subset relation holds true for the two fingerprint methods. The CSFP, tCSFP, gCSFP, and pCSFP capture the number of heavy atom neighbors in the whole compound which is two for the labeled carbon atom in the propane and three in the 2-methylpropane. Consequently the derived identifiers for structural features containing the labeled carbon differ in the two compounds. This problem can be solved by attaching an R-group to the labeled carbon atom of the propane. The derived propyl group can now be viewed as a building block of the 2-methylpropane. If we interpret R-groups as heavy atom neighbors of the adjacent atoms, the labeled carbon atom of the propyl group now has three heavy atom neighbors, and the identifiers contained in the fingerprint of the propyl group form a subset of the identifiers present in the fingerprint of the 2-methylpropane for the CSFP, tCSFP, gCSFP, and pCSFP. By changing the representation of a building block a fingerprint method can now use the quantity of heavy atom neighbors as an atom property and still fulfill the fingerprint subset relation. Other atom properties can be more difficult to apply in the context of combinatorial fragment spaces. Ring membership and stereochemical information on an atom can both change when rings are formed or the three-dimensional configuration is adjusted during a reaction. The membership of an atom to a ring or its stereochemical properties might depend on the choice of reaction and involved reagents and not only on the building block itself.

In summary we have seen that the fingerprint subset relation is very important for the efficient applicability of a fingerprint method in the context of combinatorial fragment spaces. In order to fulfill said subset relation, the fingerprint method must in turn obey the structural feature subset relation which, in contrast to the ECFP, the CSFP, its variants, and the RDKit fingerprint obey. The iCSFP and RDKit fingerprint fulfill the fingerprint subset relation as they only consider atom properties independent of the environment surrounding a structural feature. By changing the representation of building blocks, additional atom properties can be used by fingerprint methods fulfilling the fingerprint subset relation.

Analysis of Feature Space. Since the CSFP covers all structural features of a compound following the specified size constraints by design, the number of features generated for a given library is larger than that of any fingerprint method restricted to a structural subclass like the pCSFP. The CSFP generates more features than the iCSFP, tCSFP, or gCSFP since the features of the CSFP contain more information and are therefore more specific. To quantify these observations, we use the ChEMBL24.1 data set[42] which contains 1828820 distinct compounds. For the iCSFP, pCSFP, gCSFP, tCSFP, and CSFP the quantity of features with a number of heavy atoms between one and ten is shown on the left side in Figure 8. As a general trend, we can see that number of features with a specific number of heavy atoms grows exponentially for the

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

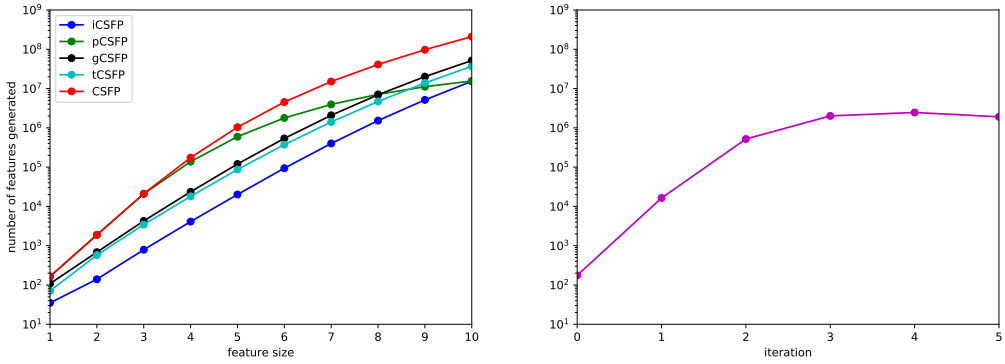


Figure 8. Number of features generated by the CSFP, its varieties on the left side, and by the ECFP on the right side. On the left side the *x*-axis accounts for the number of heavy atoms contained in the generated features. On the right side the iteration of the ECFP algorithm is depicted.

iCSFP, gCSFP, tCSFP, and CSFP. The number of features containing ten atoms generated by the CSFP exceeds that of the pCSFP by roughly 1 order of magnitude. For smaller structural features the restriction onto paths plays no role, since there are only pathlike structural features on up to three atoms except for the 3-membered ring. Moreover the iCSFP, tCSFP, and gCSFP generate fewer features compared to the CSFP since they capture less atom properties. On the right side we see the number of features newly generated in an iteration between zero and five for the ECFP. The zeroth iteration represents the initial atom identifiers. Note that the iteration of the ECFP algorithm and number of heavy atoms contained in the features generated are not directly linked. The number of features of the iCSFP, pCSFP, tCSFP, and gCSFP containing ten atoms exceed that of the ECFP generated in the fourth iteration by 1 order of magnitude and the number of features of the CSFP containing ten atoms by roughly 2 orders of magnitude.

Application Scenarios. Like for ECFP, a certain fingerprint type and parameter setting might be more appropriate for a certain application scenario. We have seen that the CSFP is highly adaptable to different data sets by the choice of a lower and an upper bound for the structural features generated.

Similarity-Driven Virtual Screening. In this scenario the distinction of active from nonactive compounds in large molecule databases is desired. In general we recommend the CSFP with small upper and lower bounds, e.g., the CSFP2.5., since this configuration showed a good performance over the whole benchmark. If the fingerprint should be highly sensitive, then a high lower and an upper bound are recommended, e.g., the CSFP9.10. This way the structural features considered are very specific, detect common scaffolds, and support an early enrichment of active compounds. If only pathlike structural features should be considered, the pCSFP should be used, e.g., the pCSFP2.7. These bounds can be adapted according to the expected size of the common features in the molecules. If the fingerprint should be able to perform "scaffold hopping", a fingerprint method with a good general VS performance should be chosen.[30] Here we recommend the tCSFP with a small distance between the lower and upper bounds, e.g., the tCSFP5.8.

Maximum Common Subgraph Problem (MCS). In this scenario the maximum common subgraph of two compounds needs to be derived. The node-based version of the MCS problem is concerned with finding the biggest common structural atom feature of two compounds, whereas the edge-based version derives the biggest common structural bond feature.[8,9,43] Typically only atom and bond types as well as the underlying graph structure of the compound are considered for the MCS problem, but there are also implementations considering other properties of the structural features such as ring membership.[18,31] Since the RDKit fingerprint considers all structural bond features and the CSFP, iCSFP, tCSFP, and gCSFP consider all structural atom features, they are suited for usage in the edge and node-based versions of the MCS problem, respectively. In Figure 9 cyclobutane and butane are

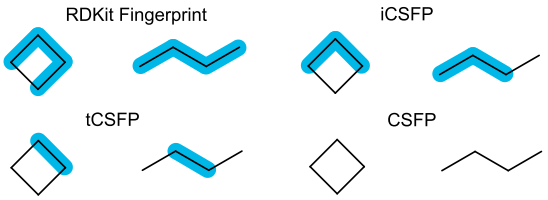


Figure 9. Biggest common structural feature with the same identifier colored in blue for cyclobutane and butane, derived using the RDKit fingerprint (top left), iCSFP (top right), tCSFP (bottom left), and CSFP (bottom right).

shown. Their fingerprints are calculated using the RDKit fingerprint, iCSFP, tCSFP, and CSFP. For each fingerprint method the largest structural feature that has the same identifier in both fingerprints is colored in blue. Since the RDKit fingerprint considers structural bond features, each set of three bonds of the cyclobutane together with the four underlying atoms is considered in its fingerprint and is assigned the same identifier as butane itself. The CSFP and its variants derive structural atom features so the set of all four atoms is only considered with all four ring bonds between the atoms. Since this structural atom feature is not contained in butane, the largest common structural atom feature for both compounds is the chain containing three carbon atoms. For the iCSFP this structural atom feature gets the same identifier assigned in both compounds, making it the largest structural feature with common identifier. In contrast to the iCSFP, the tCSFP considers the number of heavy atom neighbors in the whole compound in the identifiers of the structural features. In the cyclobutane each atom has two heavy atom neighbors,

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

whereas the two terminal carbons in the butane have only one heavy atom neighbor. So the largest structural atom features with a common identifier in both compounds in the chain contain two carbons. The CSFP includes information about ring membership resulting in the fingerprints of cyclobutane and butane sharing no common identifier. In conclusion, the CSFP, its variants, and the RDKit fingerprint are suitable for the application in an MCS-like scenario due to the complete consideration of the structural feature space of a compound. If additional atom properties such as ring membership should be included in the calculation of the maximum common subgraph, a CSFP variant considering that property should be favored.

Compound Library Analysis. In this scenario the structural diversity of a compound data set needs to be measured, or two distinct libraries should be compared for common structural features or possible subset relations. Here we recommend the iCSFP. As pointed out before, its identifiers describe structural features without considering the connectivity around the substructure. Because of this the iCSFP is a suitable method for pattern mining.[44] The derived structural features of a query compound can easily be assessed against those of a library using the identifiers contained in the fingerprint. Alternatively even two libraries can be compared. All iCSFP identifiers present in a compound data set give a complete description of its structural properties and are therefore well-suited for diversity analysis of data sets. Since they are numeric identifiers, they allow for efficient clustering of the compounds contained and comparison of data sets. We recommend using a small lower bound and a large upper bound, e.g., iCSFP1.10, to generate a high quantity of structural features. If only structural features of a certain size are of interest, the bounds can be adjusted accordingly.

We have seen that the construction of the CSFP gives it a rich feature space that covers all structural features. This leads to a high interpretability since not only the presence but also the absence of a certain structural feature in a data set can be meaningful.

Here sparse fingerprint representations were used, but folding can be applied in a straightforward way if desired. Moreover the fingerprint of a chemical building block and of a derived product satisfies the subset relation. This makes the CSFP an ideal candidate for ligand-based VS in fragment spaces. Furthermore, we gave recommendations for choice of CSFP variation and parameter selection for specific application scenarios.

CONCLUSION

A detailed description of a novel fingerprint generation method was given, and the performance of four different variants was assessed using a well-known benchmark. It was shown that the method slightly outperforms ECFP. This indicates that a restriction of radial structural features can be harmful. The application range of fingerprints is extremely large, so many other benchmarks and comparisons could be done in principle. The performance of the fingerprint is only one issue. From our perspective, its mathematical features, namely its defined completeness, is its major advantage. The richness of the CSFP feature space was showcased, and its applicability to similarity-driven VS on fragments was established. Due to its properties, the CSFP can be integrated into many similarity-driven workflows. Recommendations for three classical application scenarios were expressed. For the future, it will be interesting to see the effect of using CSFPs in machine learning applications. Furthermore, the CSFP might form a basis to model fingerprint-driven similarity searching in a more combinatorial fashion.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.9b00571.

Correlation plots and results of Wilcoxon signed-rank tests for all pairs of representative fingerprints and detailed description of substructure enumeration algorithm (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: rarey@zbh.uni-hamburg.de.

ORCID

Louis Bellmann: 0000-0002-7920-1889
Patrick Penner: 0000-0003-4988-6183
Matthias Rarey: 0000-0002-9553-6531

Notes

The authors declare the following competing financial interest(s): M.R. declares a potential financial interest in the event that the CSFPy software is licensed for a fee to nonacademic institutions in the future.

Additional supporting research data for all five fingerprints for this article may be accessed using the Python module CSFPy available for Linux, OS X, and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and are free for academic use and evaluation purposes and an open-source C++ library for enumerating connected induced subgraphs following the CONSENS algorithm may be accessed at https://github.com/rareylab/ConsensLib.

ACKNOWLEDGMENTS

The authors would like to thank Greg Landrum for several highly valuable suggestions made which substantially improved the description and validation of the Connected Subgraph Fingerprints.

REFERENCES

(1) Johnson, M.; Basak, S.; Maggiora, G. A Characterization of Molecular Similarity Methods for Property Prediction. *Math. Comput. Model.* **1988**, *11*, 630−634.

(2) Mackey, M. D.; Melville, J. L. Better Than Random? The Chemotype Enrichment Problem. *J. Chem. Inf. Model.* **2009**, *49*, 1154−1162.

(3) Clark, A. M.; Labute, P. Detection and Assignment of Common Scaffolds in Project Databases of Lead Molecules. *J. Med. Chem.* **2009**, *52*, 469−483.

(4) Sahoo, S.; Adhikari, C.; Kuanar, M.; Mishra, B. A Short Review of the Generation of Molecular Descriptors and their Applications in Quantitative Structure Property/Activity Relationships. *Curr. Comput.-Aided Drug Des.* **2016**, *12*, 181−205.

(5) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; John Wiley & Sons: Weinheim, 2008; Vol. *11*, DOI: 10.1002/9783527613106.

(6) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular Fingerprint Similarity Search in Virtual Screening. *Methods* **2015**, *71*, 58−63.

(7) Dean, P. M.; Lewis, R. A. *Molecular Diversity in Drug Design*; Springer: Dordrecht, 1999; DOI: 10.1007/0-306-46873-5.

(8) Raymond, J. W.; Gardiner, E. J.; Willett, P. Rascal: Calculation of Graph Similarity Using Maximum Common Edge Subgraphs. *Comput. J.* **2002**, *45*, 631−644.
(9) Raymond, J. W.; Willett, P. Maximum Common Subgraph Isomorphism Algorithms for the Matching of Chemical Structures. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 521−533.
(10) Morgan, H. The Generation of a Unique Machine Description for Chemical Structures - a Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.
(11) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Model.* **1985**, *25*, 64−73.
(12) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsion: a New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf. Model.* **1987**, *27*, 82−85.
(13) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(14) Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Molecular Similarity Searching Using Atom Environments, Information-based Feature Selection, and a Naive Bayesian Classifier. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 170−178.
(15) *MACCS Structural Keys*; Accelrys: San Diego, 2011.
(16) McGregor, M. J.; Muskal, S. M. Pharmacophore Fingerprinting. 1. Application to QSAR and Focused Library Design. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 569−574.
(17) *Indigo Toolkit*; GGA Software Services. https://lifescience.opensource.epam.com/indigo/ (accessed 23/06/2019).
(18) Landrum, G. *RDKit: Open-source Cheminformatics*, version 2018.09.1. https://www.rdkit.org (accessed 09/09/2018).
(19) James, C. A.; Weininger, D.; Delany, J. *Daylight Theory Manual*; Daylight Chemical Information Systems. Inc.: Irvine, CA, 1995.
(20) Lauck, F.; Rarey, M. Coping with Combinatorial Space in Molecular Design. *De novo Molecular Design* **2013**, 325−347.
(21) Hoffmann, T.; Gastreich, M. The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discovery Today* **2019**, *24*, 1148−1156.
(22) Stahl, M.; Mauser, H.; Tsui, M.; Taylor, N. R. A Robust Clustering Method for Chemical Structures. *J. Med. Chem.* **2005**, *48*, 4358−4366.
(23) Rarey, M.; Stahl, M. Similarity Searching in Large Combinatorial Chemistry Spaces. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 497−520.
(24) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Model.* **1989**, *29*, 97−101.
(25) Babai, L.; Luks, E. M. Canonical Labeling of Graphs. *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*; 1983; pp 171−183, DOI: 10.1145/800061.808746.
(26) Hsieh, S.-M.; Hsu, C.-C.; Hsu, L.-F. Efficient Method to Perform Isomorphism Testing of Labeled Graphs. *International Conference on Computational Science and Its Applications*; 2006; pp 422−431, DOI: 10.1007/11751649_46.
(27) Foggia, P.; Sansone, C.; Vento, M. A Performance Comparison of Five Algorithms for Graph Isomorphism.*Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*; 2001; pp 188−199.
(28) Hopcroft, J. E.; Wong, J.-K. Linear Time Algorithm for Isomorphism of Planar Graphs (preliminary report). *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*; 1974; pp 172−184, DOI: 10.1145/800119.803896.
(29) Cahn, R. S.; Ingold, C.; Prelog, V. Specification of Molecular Chirality. *Angew. Chem., Int. Ed. Engl.* **1966**, *5*, 385−415.
(30) Riniker, S.; Landrum, G. A. Open-Source Platform to Benchmark Fingerprints for Ligand-based Virtual Screening. *J. Cheminf.* **2013**, *5*, 26.
(31) *JChem. 282*; ChemAxon LLC. https://chemaxon.com (accessed 20/08/2019).
(32) *OpenEye Scientific Software*, OEChem. https://www.eyesopen.com (accessed 20/08/2019).
(33) Héberger, K. Sum of Ranking Differences Compares Methods or Models Fairly. *TrAC, Trends Anal. Chem.* **2010**, *29*, 101−109.
(34) Kollár-Hunek, K.; Héberger, K. Method and Model Comparison by Sum of Ranking Differences in Cases of Repeated Observations (Ties). *Chemom. Intell. Lab. Syst.* **2013**, *127*, 139−146.
(35) Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80−83.
(36) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49*, 169−184.
(37) Irwin, J. J. Community Benchmarks for Virtual Screening. *J. Comput.-Aided Mol. Des.* **2008**, *22*, 193−199.
(38) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a Large-scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.
(39) Irwin, J. J.; Shoichet, B. K. ZINC- a Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.
(40) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 947.
(41) Ginn, C. M.; Willett, P.; Bradshaw, J. Combination of molecular similarity measures using data fusion. In *Virtual Screening: An Alternative or Complement to High Throughput Screening?*; Springer: 2000; pp 1−16, DOI: 10.1007/0-306-46883-2_1.
(42) Gaulton, A.; et al. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945−D954.
(43) Sayle, R. A.; Batista, J.; Grant, J. A. Efficient Maximum Common Subgraph (MCS) Searching of Large Chemical Databases. *J. Cheminf.* **2013**, *5*, O15.
(44) Borgelt, C.; Meinl, T.; Berthold, M. Moss: A Program for Molecular Substructure Mining. *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations.* **2005**, 6−15.

DOI: 10.1021/acs.jcim.9b00571
J. Chem. Inf. Model. 2019, 59, 4625−4635

D.2. Topological Similarity Search in Large Combinatorial Fragment Spaces

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

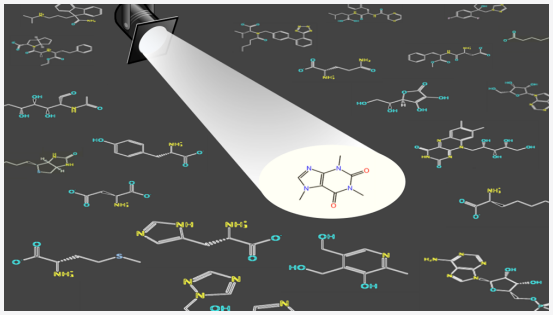
Topological Similarity Search in Large Combinatorial Fragment Spaces

Louis Bellmann, Patrick Penner, and Matthias Rarey*

Cite This: J. Chem. Inf. Model. 2021, 61, 238−251 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: In similarity-driven virtual screening, molecular fingerprints are widely used to assess the similarity of all compounds contained in a chemical library to a query compound of interest. This similarity analysis is traditionally done for each member of the library individually. When encoding chemical spaces that surpass billions of compounds in size, it becomes impractical to enumerate all their products, let alone assess their similarity, deeming this approach impossible without investing a substantial amount of resources. In this work, we present a novel search algorithm named SpaceLight for topological fingerprint similarity searching in large, practically non-enumerable combinatorial fragment spaces. In contrast to existing methods, SpaceLight is able to utilize the combinatorial character of these chemical spaces for efficiency while maintaining a high correlation of the description of molecular similarity to well-known molecular fingerprints like ECFP. The resulting software is able to search prominent spaces like EnamineREAL with more than 10 billion compounds in seconds on a standard desktop computer.



INTRODUCTION

In similarity-based virtual screening, compounds are retrieved from a chemical library by measuring the similarity to a known active compound functioning as a query. Following the similarity principle,[1] the retrieved highly similar compounds potentially exhibit a similar bioactivity profile as the query compound. To this end, molecular fingerprints[2] are often used to describe compounds by occurring chemical substructures represented in a bitstring, a set of numerical identifiers, or counts in a vector. Due to this compact representation, they enable an efficient similarity calculation while capturing important physio-chemical properties of compounds. Molecular fingerprints have also been applied in numerous other tasks such as clustering,[3] diversity analysis,[4] and common scaffold detection.[5]

Traditionally, the similarity of a chemical libraries' compounds to a given query is assessed sequentially. This characteristic of state-of-the-art similarity-driven fingerprint methods forms a bottleneck when searching in large chemical libraries exceeding billions of compounds in size. In fact, all compounds of chemical libraries of this size cannot even be practically enumerated without significant computational effort.[6]

Chemical libraries frequently used in virtual screening obviously represent only a tiny fraction of chemical space. Combinatorial chemical libraries are a possibility to overcome this limitation.[7] DNA-encoded libraries[8] are a method of combinatorial chemistry that combines fragments and reconstructs their synthetic pathway with DNA tags. In virtual screening, combinatorial fragment spaces encode buildings blocks together with chemical reactions and, through combinatorial explosion, can span a chemical space exceeding the quantity of fragments and reactions by several orders of magnitude in size. Following this approach, ChemSpace[9,10] and AllChem[11] incorporate a reaction-driven combinatorial library design together with a shape-based molecular descriptor[12] that can be used for efficient similarity searches. The Proximal Lilly Collection[13] is a large collection of virtual compounds that can be synthesized using readily available starting materials and well-known reactions. In addition, combinatorial fragment spaces recently gained further attention with several proprietary[14,15] and a publicly available space[16] being created. Most prominently, the REAL Space[17] presents the possibility to select from spaces comprising more than 13 billion compounds that can be synthesized on demand. Even highly sophisticated search engines running on parallel database servers[18] will not be able to cope with the exponential growth of these combinatorial collections in the future.

Received: July 27, 2020
Published: October 21, 2020

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

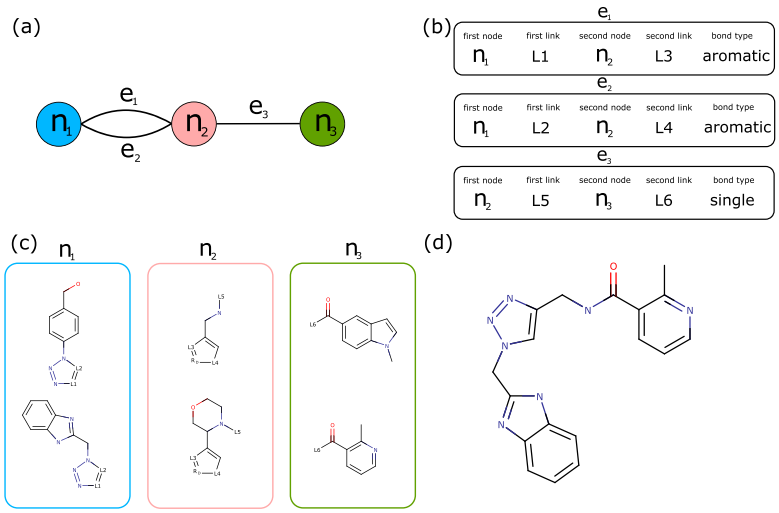


Figure 1. In (a), a topology graph containing three nodes and three edges is depicted. In (b), the information considering involved nodes, link names, and types of formed chemical bonds is given for each edge of the topology. In (c), the list of fragments for each node of the topology is specified. In (d), the compound contained in the product space of the topology graph is given by choosing the second fragment for the first node, the first fragment for the second node, and the second fragment for the third node.

One technology that is currently used for similarity-driven virtual screening in these spaces is the FTrees-FS algorithm[19] forming the core of BioSolveIT's infiniSee[20] technology. It decomposes a compound into rings and acyclic atoms and captures their size, shape, and chemical properties such as the ability to form intermolecular interactions. This pharmacophore-like molecular descriptor is therefore conceptually distinct from the substructure-driven approach of topological fingerprint methods. While randomized, heuristic search procedures following evolutionary concepts exist,[21] a combinatorial search procedure based on topological fingerprints has, to the best of our knowledge, not been developed so far.

In addition to covering large areas of chemical space, the combinatorial structure of a library can encode the synthetic pathways of its compounds. Generative models based on neural networks recently gained attention[22] since they can be combined with arbitrary objective functions and tailored to specific application scenarios. However, the synthesis routes of the generated compounds are not always straightforward. By following up with a combinatorial similarity search as presented in this paper, synthetic accessibility can be ensured while keeping the versatility of the generative model.

In this work, we describe a novel search algorithm for large combinatorial fragment spaces called SpaceLight. It utilizes the well-known ECFP[23] and the Connected Subgraph Fingerprint (CSFP)[24] to describe molecular similarity. In contrast to existing workflows using fingerprint methods, the SpaceLight approach is able to exploit the combinatorial character of fragment spaces and consequently can conduct similarity searches considering billions of compounds within seconds on a standard PC. Furthermore, we overcome the limitation to near tree-like molecular topologies known from FTrees-FS.

METHODS

In this section, we introduce the notion of topological fragment spaces and discuss the internal representation of fragments as well as the search procedure in detail.

Topological Fragment Spaces. Topological fragment spaces describe chemical spaces that are obtained by applying a set of reactions to two or more reagents to form products. Each reagent can be chosen from a set of chemical building blocks compatible with the reaction at a specified position.

A topological fragment space consists of a set of topology graphs. Each topology graph can be interpreted as a scheme of one or multiple consecutive reactions with different sets of building blocks to choose from. The topology graph contains a set of nodes that resemble the different groups of reagents. Each topology node contains a set of fragments that correspond to the different substitutions at this position derived by applying information about the involved chemical reactions to the groups of reagents. The fragments contain so-called link atoms marking the positions where new bonds are formed during the reaction. Each link atom has a unique name for identification. The number and configuration of link atom names is the same for all fragments contained in one node. Additionally, the topology graph contains a set of edges that resemble the chemical bonds that are formed between the building blocks during the reaction. Each edge contains two nodes, one link atom name from each node and the type of chemical bond that is formed. The product space of a topology graph consists of all compounds that can be obtained by selecting a fragment from each node and applying the connection rules given by the edges of the topology graph. Note that the choice of fragment for one node is independent of the fragments chosen for the other nodes as the link atom configuration is the same for each fragment of the node. In Figure 1, a topology graph with three nodes and three edges is shown. A triazole is formed between the fragments from the first and second node upon combination. The fragments of node two and three form an amide bond. The fragments in the first and second node contain a five-membered ring, which partially consists of link atoms. Apart from link atoms with names L3, L4, and L5, which are specified in the edges of the topology graph, the fragments of the second node contain a link atom with name R_D. This link atom functions as a placeholder to form a five-

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

membered aromatic ring in the fragments, such as the previously mentioned triazole. This way, the presence of an aromatic ring at this position is encoded in the fragment and will be represented in its fingerprint.

Topological Fingerprints in Fragment Spaces. The number of products in a topological fragment space can surpass the number of fragments by several orders of magnitude. For this reason, the procedure described in this work operates solely on the fragments of a topological fragment space, not the products, and exploits the combinatorial character of the space. Seeing as the method aims to measure similarity between a query compound and all products of a topological fragment space, it is important that the topological fingerprint descriptor applied to the fragments approximates the description of the product obtained by combining the fragments. In this work, the ECFP and CSFP fingerprint methods are used to describe chemical similarity in fragment spaces. The well-known ECFP describes a compound by all its circular chemical features. The CSFP considers all chemical features instead resulting in a dense fingerprint with several interesting properties described in the original publication.[24]

Connected Substructure Fingerprints (CSFPs). Among the various fingerprint types of the CSFP, we will use the iCSFP and the tCSFP in the following. The iCSFP describes all structural features of a compound by properties of the atoms that are independent of the surrounding environment of the substructure. Therefore, it is suitable when searching for products that form a chemical substructure of a given query compound and vice versa. The tCSFP captures all structural features of a compound but considers the same atom properties as the TopologicalTorsion[25] fingerprint. Moreover, we introduce a novel variant of the CSFP suitable for the application in a fragment-based approach called fCSFP. The fCSFP is similar to the standard CSFP but neglects stereochemical information of the fragment, which may change as a result of attaching other fragments. In Table 1, the characteristics of the ECFP, fCSFP, iCSFP, and tCSFP are given by the chemical features they consider as well as the atom properties they capture.

Fragment Representation. The ECFP and fCSFP both consider valence, number of heavy atom neighbors, and ring membership of atoms. All of these properties can change for atoms where new chemical bonds are formed during a reaction, resulting in identifiers that are present in the fingerprint of the fragment but not in the product fingerprint. To avoid, this we attach link atoms to the fragments and use additional ring placeholders to form the rings that are generated during a reaction. Additionally, the chemical bonds are adapted to match their configuration and order in the product. For the fingerprint generation in the CSFP and ECFP workflow, the newly added link atoms and ring placeholders are only incorporated implicitly through the connectivity, valence, and ring membership identifiers of their heavy atom neighbors. The substructures considered by the two fingerprint approaches do not contain link atoms or ring placeholders. This fragment representation can be applied to any building block and reaction. As a result, the fingerprint of a fragment, when generated by the CSFP, forms a subset for all products the fragment is a part of. This fingerprint subset relation between fragments and products enables the CSFP to describe the products of a topological fragment space using only its fragments and thus avoiding combinatorial explosion. In a later section, we will discuss an application scenario where this property of the CSFP is a necessary requirement. The ECFP considers only circular features. For this reason, it does not fulfill the fingerprint subset relation as explained in detail in an earlier work[24] and shown in the following example. However, our results in this work suggest that the ECFP is able to approximate the chemical description of the product space by only using fingerprints of fragments. In Figure 2, formaldehyde, vinylethylene, and their product dihydropyran in a Diels−Alder reaction are shown in their standard representation in (a) and our fragment representation in (b). The identifiers generated by the ECFP2 for the circular features around the oxygen atom depicted in (a) differ in the formaldehyde and the dihydropyran since a ring is formed and chemical bond types change during the reaction. In the novel fragment representation of the formaldehyde shown in (b), the oxygen atom is adjacent to a link atom. The fragment representations of formaldehyde and vinylethylene are both composed of a ring containing link atoms and resembling the chemical bond configuration in the dihydropyran. This way, all initial identifiers generated by the ECFP for the two fragments are contained in the fingerprint of their product. Therefore, the fingerprints generated by the ECFP0 for the two fragments form a subset of the product fingerprint. The same holds true for the fCSFP, tCSFP, and iCSFP as they consider all structural features of the fragments, which naturally form a subset of the structural features of their product. The second feature generated by the ECFP2 around the oxygen atom in the fragment representation of the formaldehyde in (b) is not contained in the fingerprint generated by the ECFP2 for the dihydropyran. Only one other heavy atom is adjacent to the oxygen atom in the formaldehyde, whereas the corresponding oxygen atom of the dihydropyran has two heavy atom neighbors.

Table 1. Fingerprint Types Used by SpaceLight Together with the Chemical Substructures and Atom Properties They Describe

		ECFP	fCSFP	iCSFP	tCSFP
chemical substructures	circular	x			
	all		x	x	x
	element	x	x	x	x
	connectivity	x	x		x
	connectivity in substructure		x	x	
	valence	x	x	x	
atom properties	valence in substructure		x	x	
	aromaticity				x
	π electrons				
	formal charge	x			
	weight	x			
	ring membership	x	x		

SpaceLight Algorithm. The SpaceLight algorithm operates only on the fragments of a combinatorial fragment space. This way, the potential combinatorial explosion generating large-sized product spaces is avoided. Simply comparing the query compound to the individual fragments is, however, also not an option because a compound as a whole may not be sufficiently similar to any fragment. Instead, a query compound is divided into smaller parts and these parts are then in turn compared to the fragments of the combinatorial space.[19]

First, we briefly summarize the method; a more detailed description of each step with concrete examples will be given below. The method consists of four steps:

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

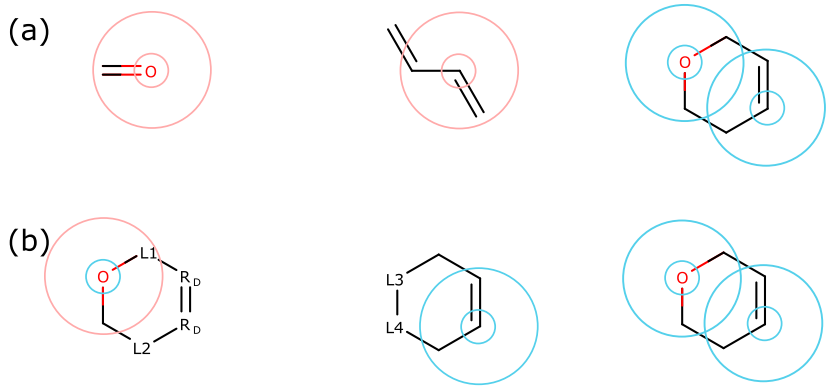


Figure 2. In (a), formaldehyde, vinylethylene, and their product dihydropyran are depicted. Additionally, the circular substructures generated by the ECFP2 around the oxygen atom and one carbon atom of the vinylethylene are shown for both fragments and the product. Features that are part of a fragment fingerprint, but not contained in the fingerprint of the product, are specified in pink. In (b), the fragment representation of formaldehyde and vinylethylene are given together with their product dihydropyran and the corresponding features generated by the ECFP2. Features that are now contained in both the fingerprint of the product and the fragment are depicted in blue.

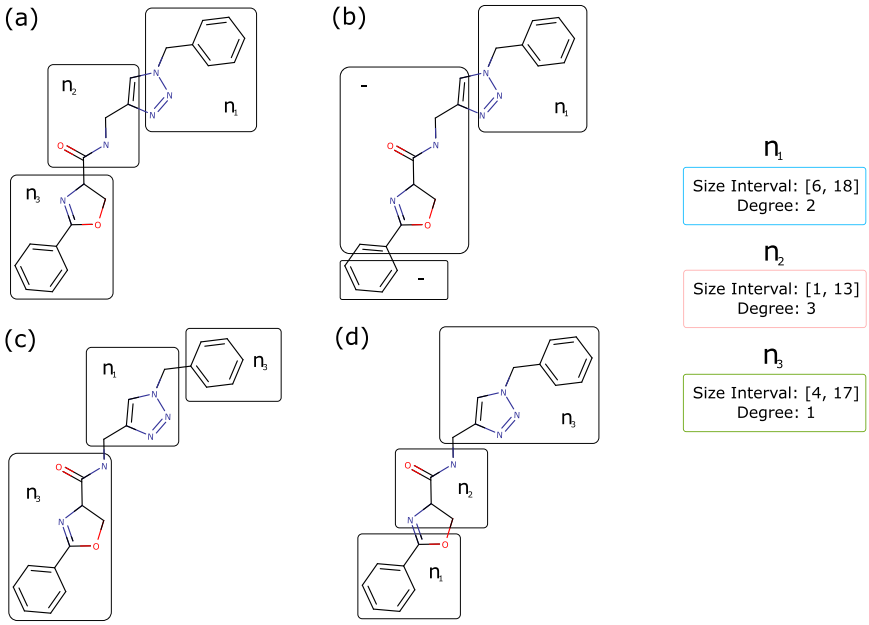


Figure 3. Four different partitions of the ChEMBL[27] compound CHEMBL1091518 are depicted together with the compatible nodes of the topology graph shown in Figure 1. For each node of the topology graph, the interval of all numbers that differ by at most five from the number of heavy atoms of at least one fragment is depicted as well as the degree of the node. A substructure is compatible to a node of the topology graph if its number of heavy atoms lays within the interval for that node and the degree of the substructure corresponds to the degree of the node.

1. A partitioning step in which all subdivisions, or partitions, of the query compound into connected substructures are determined. Topology and size filters are applied to the partitions such that they resemble topology graphs of the topological fragment space.
2. A matching step enumerating all possible matches of the partition classes of all partitions onto nodes of compatible topology graphs. The partition classes, which are connected substructures of the query compound, must be similar in size to some fragment in the node and have the same connectivity as the node they are to be matched on. A topology score is calculated that captures the similarity of the chemical bond types given by the edges of the topology graph and the types of chemical bonds connecting the partition classes of the query.
3. A comparison step in which the similarity between a connected substructure and all fragments contained in the topology node are calculated for each matched pair of partition class and node of a topology graph. The similarity is determined using topological fingerprints and the Tanimoto[26] coefficient. The fragments are then

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

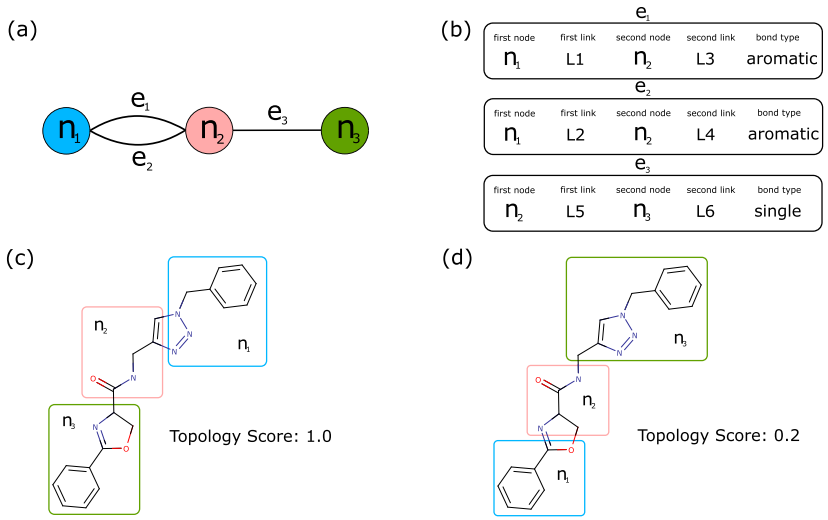


Figure 4. In (a), the topology graph shown in Figure 1 is depicted. In (b), the edges of the topology graph are shown. In (c) and (d), two partitions from Figure 3 of the ChEMBL compound CHEMBL1091518 are depicted together with their matching onto the nodes of the topology graph and their Tanimoto scores. The color of the box indicating the classes of the partition coincides with the color of the topology node the substructure is matched to.

ranked based on their similarity to the connected substructure of the query.

4. A combination step in which for each matching enumerated in step two the fragment combinations most similar to the matched partition are determined. To this end, the fragment rankings for each node of the matched topology graph derived from step three are utilized. As a fragment, combination can occur in similar results for multiple matchings; the results are then summarized across all matchings to determine the overall most similar compounds of the product space to the query compound.

In the following, we describe all steps of the algorithm in detail. The topology graph depicted in Figure 1 will be used as an example throughout this subsection.

Partitioning Step. In this step, the query compound is partitioned to mimic fragment combinations contained in the topological fragment space. We say a connected substructure of the query molecule is compatible to a fragment if the number of heavy atoms present in the substructure differs by at most five from the number of heavy atoms present in the fragment. This way, two molecules that differ by exchanging a single atom with a small ring of at most six atoms are still considered compatible. Additionally, the number of chemical bonds cut to form the substructure must coincide with the number of chemical bonds in that fragment containing one heavy atom and one link atom. We call these chemical bonds for a given partition crossing bonds in the following. The degree of a connected substructure is the number of incident crossing bonds. A connected substructure of the query molecule is compatible to a node of the topology graph if it is compatible to at least one of its fragments. If a chemical substructure is compatible to a node of the topology graph, it means that it is similar in size to at least one fragment in the node and has a degree corresponding to the number of link atoms contained in that fragment. Later, the fingerprint of that chemical substructure will only be compared to fingerprints of these similar fragments.

First, all connected substructures of the query molecule compatible to at least one node of a topology graph are enumerated using the CONSENS algorithm.[4] We call a set of connected substructures a disjoint family if each atom of the underlying compound is contained in at most one substructure. The enumerated substructures are used to recursively form disjoint families until they span the whole query compound. During the recursion, for each disjoint family, all topology graphs are tracked, for which a matching between the contained substructures and nodes of the topology graph is possible. For the matching, only pairs of compatible substructures and nodes of the graph can be used. As connected substructures are added to a disjoint family, the set of topology graphs that it can be matched to is updated. A disjoint family is rejected if it is not matchable to any topology graph or no substructure can be added to form a partition. This step terminates with a set of partitions of the query compound that are matchable to at least one topology graph under the described constraints. In Figure 3, four example partitions of the ChEMBL[27] compound CHEMBL1091518 are shown. For each connected substructure, the compatible nodes of the topology graph are given. The partitions shown in (a) and (d) both have possible matches to the topology graph using only compatible pairs of substructure and topology node. In (b), a partition is depicted for which two out of three substructures are not compatible to any node of the topology graph as they either contain too few heavy atoms or have too many incident crossing bonds. These incompatible substructures would not be enumerated by the CONSENS procedure and are therefore not used to form partitions. In (c), a partition is depicted where each substructure has a compatible node of the topology graph, but a matching of the partition onto the topology graph is not possible since no substructure is compatible to the second node of the topology graph.

Matching Step. In this step, for all partitions found in the previous step, all matchings to the topology graphs are enumerated. Since the compatible topology nodes were already determined for each connected substructure contained in some

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

partition, the calculation of the matchings can be done efficiently in a recursive manner. For each of these matchings, the crossing bonds between the connected substructures of the partition are compared to the edges running between the matched nodes of the topology graph. To this end, for each adjacent pair of topology nodes and matched substructures, the quantity and type of chemical bonds between the two nodes and the two substructures is assessed. For each type of chemical bond, the minimum quantity of bonds between the two substructures and the edges between the two topology nodes is counted. The sum of all these counts is utilized in a Tanimoto coefficient together with the total number of edges present in the topology graph. The derived coefficient is called the topology score. This step terminates with all matchings that obtained a topology score of 1.0, meaning that the topology of the partitions corresponds to the topology of the graph they are matched to. Figure 4 shows the matchings generated for the two compatible partitions appearing in Figure 3. We first derive the Tanimoto score for the matched partition depicted in (c). The chemical bonds between the substructures matched to the first and second node of the topology graph coincide in quantity and type with the first and second edge of the topology graph that runs between the first and second node of the topology. The same holds true for the chemical bond between the substructures matched to the second and third topology node and the third edge of the topology. Consequently, the topology score amounts to 1.0. Now, we consider the matched partition depicted in (d). Between the connected substructures matched to the second and third node runs a single bond that equals the bond type of the third topology edge. However, the two chemical bonds between the substructures matched to the first and second node are not contained in an aromatic ring and consequently do not coincide with the type of chemical bond specified for the first and second topology edge. As the topology graph contains three edges and the partition three crossing bonds, we arrive at one compatible pair of crossing bond and topology edge and two remaining unmatched topology edges and two unmatched crossing bonds. This results in a topology score of 0.2. This matching would not be part of the matching steps output.

Comparison Step. In this step, the similarity between connected substructures of the query and fragments of the topological fragment space is assessed using fingerprint methods. Each matching of partition and topology graph with a Tanimoto score of 1.0 from the previous step consists of matched pairs of substructures of the query compound and nodes of a topology graph. For each of these pairs, the similarity of all fragments that are compatible to the connected substructure is determined. To this end, the Tanimoto similarity scores of the substructure's fingerprint and the fragment's fingerprint are calculated and the fragments are ranked based on this score that we call the fragment score. This step terminates with a ranked list of the highest scored fragments for each matched pair of query substructure and topology node using the ECFP or CSFP fingerprint method and the Tanimoto coefficient. The length of the ranked lists corresponds with the number of results of the method specified by the user.

Combination Step. In this step, all results from the previous steps are combined to determine the products that are most similar to the query compound across the whole topological fragment space. The algorithm processes all matchings with a topology score of 1.0 of query partitions to topology graphs separately. For each of these matchings, a list of high scoring fragments was derived in the comparison step. Now, all possible combinations of fragments from these lists are enumerated. As a final similarity measure, a combination score defined as the weighted sum of fragment scores is calculated. The weight of each individual fragment score is derived as the fraction of number of heavy atoms contained in the matched substructure divided by the total number of heavy atoms of the query compound. The enumerated fragment combinations are ranked based on their combination score, and the highest ranked combinations are stored as candidates for the respective matching. Finally, all candidates together with their combination score across all matchings are summarized. If a fragment combination occurs as a candidate for multiple matchings, the one with the highest combination score is used. The fragment combinations are again ranked based on their combination score, and the list of highest ranked fragment combinations is returned. If a compound similar to the query is contained in the product spaces of two distinct topology graphs, the resulting combination scores of the underlying fragment combinations generating the compound might differ. This is caused by the different topologies that SpaceLight accounts for. If the compound is identical to the query compound, the calculated score is always 1.0 independent of the topology.

RESULTS

To assess the performance of the method, we analyze the runtime and compare the fragment combinations returned by the method to the result of a non-combinatorial similarity search directly in the product space using fingerprint methods. The evaluation process is carried out on the Enamine REAL Space[17] containing over 13 billion commercially available compounds. This vast space is generated by applying over 180 validated synthesis protocols to over 115,000 building blocks. When encoded as a topological fragment space, it contains 232 topology graphs each resembling a single or multiple reactions applied to a set of building blocks. Additionally, we evaluated the method on the publicly available KnowledgeSpace[16] containing over 10^{15} products based on building blocks from the eMolecules[28] collection. The corresponding topological fragment space contains 117 topology graphs each resembling a chemical reaction known from the literature to a set of building blocks. The analysis workflow described below is applied to the KnowledgeSpace, and the results are given in the Supporting Information. In principle, any combinatorial chemical library can be represented as a topological fragment space. However, in this work, we restrict our analysis to the prominent Enamine REAL Space and the KnowledgeSpace at it is publicly available.

Analysis Outline. As a first experiment, 10 compounds were randomly chosen from the product space of each of the 232 topology graphs. All of these 2320 queries were retrieved by the search procedure as the only fragment combination with a score of 1.0 using the fCSFP2.5 as the parameterized fingerprint method. As with fingerprint similarity measures, SpaceLight has the property where, if the molecule is contained in a fragment space, it can be identified with a score of 1.0. In theory, it is possible that other fragment combinations with a score of 1.0 resulting in a different molecule exist. For example, a rearrangement of fragments with the same chemistry at each bond would be such a case. However, this is an extremely rare event in practice. As soon as the fragment space was constructed with real synthesis protocols, we did not find a single case. This shows the potential of the fragment-based search procedure to check the occurrence or absence of a certain compound in a combinatorial fragment space.

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

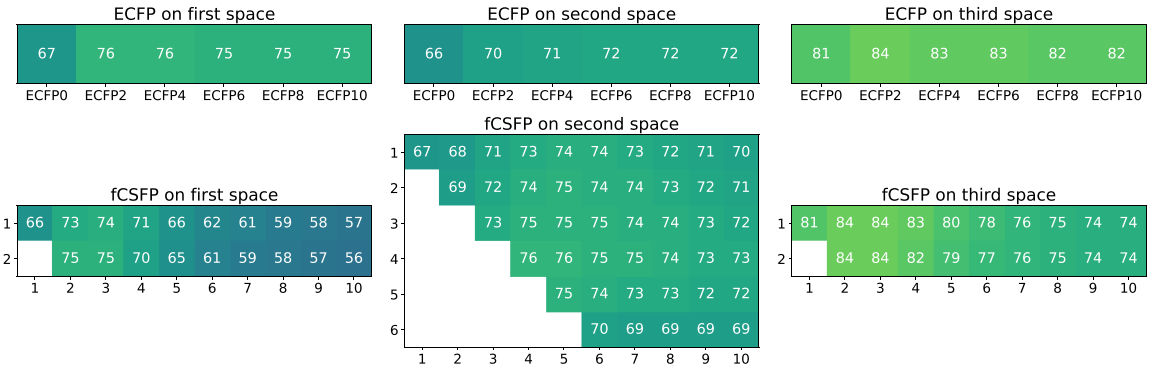


Figure 5. For three topological fragment spaces and the parameterized fingerprint methods chosen for the ECFP and fCSFP, the average Kendall's tau values across 500 randomly chosen compounds of the product space ranked by the combinatorial fragment-based and the non-combinatorial approach are shown. For the fCSFP, the lower bound is given on the y axis and the upper bound on the x axis. All values are given in percent.

In the following paragraphs, we discuss the behavior of the novel search method in combinatorial fragment spaces in correlation to a non-combinatorial similarity search in the product space utilizing fingerprint methods and the Tanimoto score. Due to the sheer size of the Enamine REAL Space, it is not possible to conduct a non-combinatorial search directly on the product space without investing a substantial amount of computing time and resources using known search methods. For this reason, we restricted our comparison of the two approaches to three randomly picked topology graphs with a product space each comprised of less than 100,000 products, which are enumerated as input for the non-combinatorial search. The three topology graphs each form a topological fragment space that is a subspace of the Enamine REAL Space. The comparison is done separately for the ECFP, fCSFP, iCSFP, and tCSFP. In principle, other fingerprint methods can be integrated into the SpaceLight approach. The fingerprints of all fragments are precalculated. As a result, only the computational time needed for the fingerprint generation of the substructures of the query compound influences the runtime of the similarity search. For the ECFP, the ECFP0, ECFP2, ECFP4, ECFP6, ECFP8, and ECFP10 were included in the analysis. Following the convention, ECFPx contains only circular chemical features of diameter at most x. Note that x has to be even. For the fCSFP$x.y$, iCSFP$x.y$, and tCSFP$x.y$, all lower bounds x from one to the smallest number of heavy atoms contained in a fragment were considered. This number amounts to two for the first and third topological fragment space and six for the second space. Additionally, all upper bounds y in the range of x and 10 are considered. We want to emphasize that the goal of this analysis is not to show the superiority of a fingerprint method in the context of combinatorial fragment spaces but instead show the applicability of the novel topological search method across different fingerprint methods.

As discussed by Riniker and Landrum,[29] a single best fingerprint method for all application scenarios does not exist and the choice of method should always be tailored to the task at hand. For a given query compound, topological fragment space, and fingerprint method, two rankings of all products contained in the product space of the topological fragment space are determined. The first ranking is based on the scores of the products achieved in the fragment-based SpaceLight procedure accounting for ties. Note that some products might not appear in the output of the combinatorial search procedure if no partition of the query compound could be found, which is topologically similar to the fragment combination of the product following the compatibility criteria formulated in the description of the method. These dissimilar products are all assigned the maximum rank. The second ranking of the products is based on their Tanimoto similarity to the query molecule considering ties using the non-combinatorial fingerprint approach. First, the two rankings are generated for a set of query compounds that is composed of 500 randomly picked products of the topological fragment spaces. We call these pairs of rankings the internal rankings. Second, a set of 100,000 randomly chosen compounds from the lead-like subset of the ZINC database were selected.[30] For each topological fragment space and parameterized fingerprint method, a molecule from this set is called comparable if it achieves a similarity Tanimoto score of at least 0.7 to at least one product of the topological fragment spaces in the non-combinatorial approach. For each space and parameterized fingerprint method, all or at most 500 comparable compounds were selected as queries for the experiment. This is done to avoid ranking the products based on a query compound that is not regarded as similar to any product by the parameterized non-combinatorial fingerprint method. We call the derived pairs of rankings the external rankings.

Correlation of Ranked Results. First, the correlation of the non-combinatorial and the fragment-based approach are evaluated using 500 internal ranking pairs. For each query and fingerprint method, the tau-b version of the Kendall's tau[31] is calculated for the ranked lists of the two approaches. In Figure 5, the average Kendall's tau values across all 500 queries for each parameterized fingerprint method are depicted for the ECFP and fCSFP. The corresponding results for the iCSFP and tCSFP are given in Figure S1 in the Supporting Information. All parameterized versions of the ECFP except the ECFP0 achieve very similar values on the same topological fragment space. On the first and third space, the parameterizations of the fCSFP, iCSFP, and tCSFP with an upper bound y of two to four are among the highest scored for their fingerprint method. On the second space, parameterizations of the fCSFP, iCSFP, and tCSFP with a lower bound x of two to four and an upper bound y from x to six perform best. The score achieved by the best parameterization for each fingerprint method is highest for the fCSFP on the second space and for the ECFP on the first space, but all four methods achieve similar results in that regard.

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

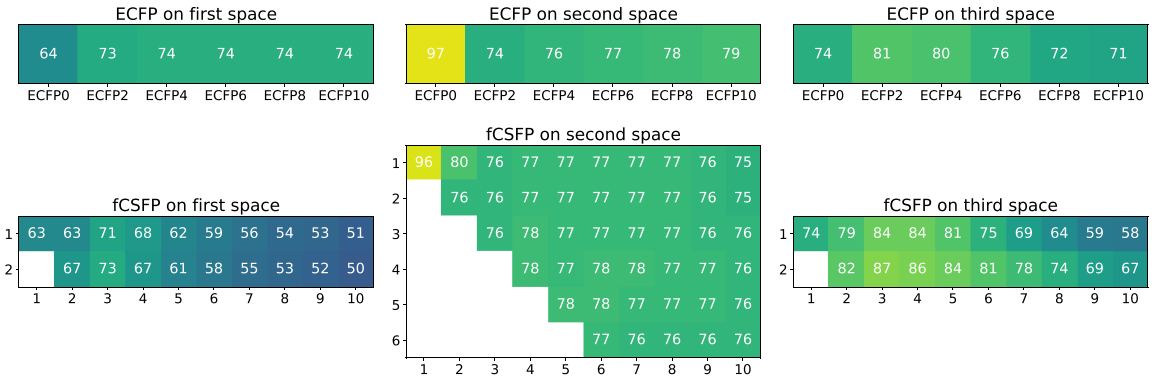


Figure 6. For three topological fragment spaces and the parameterized fingerprint methods chosen for the ECFP and fCSFP, the average Kendall's tau values across the set of comparable compounds from the ZINC database ranked by the combinatorial fragment-based and the non-combinatorial approach are shown. For the fCSFP, the lower bound is given on the *y* axis and the upper bound on the *x* axis. All values are given in percent.

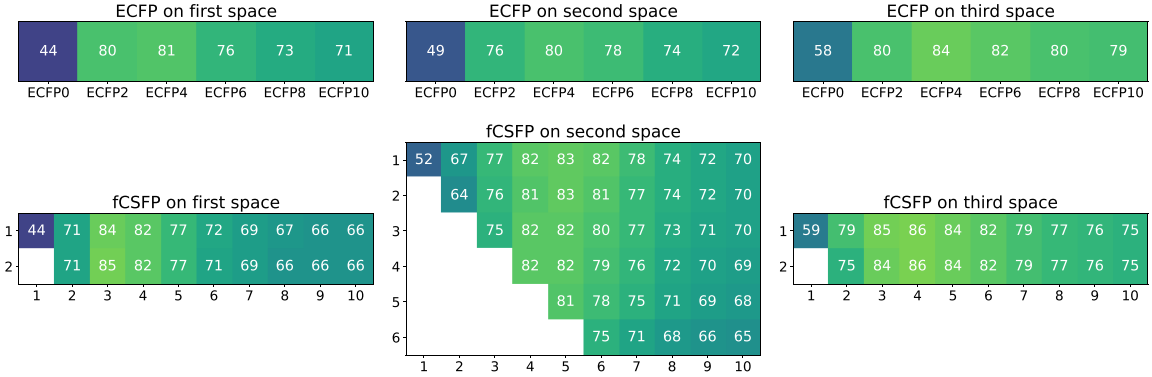


Figure 7. For three topological fragment spaces and the parameterized fingerprint methods chosen for the ECFP and fCSFP, the average relative overlap of compounds that appear in the 10 most similar results determined by the combinatorial fragment-based and the non-combinatorial approach across the internal rankings are shown. For the fCSFP, the lower bound is given on the *y* axis and the upper bound on the *x* axis. All values are given in percent.

In the second correlation experiment, the same analysis procedure is repeated for the external rankings. In Figure 6, the average Kendall's tau values across queries contained in the set of comparable compounds with respect to the parameterized fingerprint method and topological fragment space are shown for the ECFP and fCSFP. The corresponding results for the iCSFP and tCSFP are given in Figure S2 in the Supporting Information. For the ECFP, the ECFP4, ECFP6, ECFP8, and ECFP10 perform best on the first space closely followed by the ECFP2. On the second space, the ECFP0 achieves by far the highest score of 0.96 followed by the ECFP10 with a score of 0.79. On the third space, the ECFP2 with a score of 0.81 followed by the ECFP4 with a score of 0.8 achieves the best results. Similar to the results of the first correlation experiment, again the parameterized versions of the fCSFP, iCSFP, and tCSFP with an upper bound *y* of two to four perform best for their fingerprint method on the first topological fragment space. On the second space, the fCSFP1.1, iCSFP1.3, and tCSFP1.1 achieve the highest average Kendall's tau score for their CSFP variant. On the third space, the fCSFP2.3 performs best, and for the iCSFP and tCSFP, parameterizations with an upper bound of four to six achieve the highest scores for their fingerprint method. In total, the ECFP achieves the highest scores for some parameterization on the first space, the iCSFP on the second, and the fCSFP on the third space. However, like in the first correlation experiment, the highest scores are closely related.

In conclusion, we have seen that the average Kendall's tau values of the novel combinatorial fragment-based similarity search approach and the non-combinatorial Tanimoto-based similarity search procedure are in the range of 0.65−0.9 for at least one parameterization of each fingerprint method. Parameterized non-combinatorial fingerprint methods like the ECFP4, ECFP6, fCSFP2.4, and fCSFP2.5 showed good performance using an open-source benchmark platform.[24,29] These parameterizations inhibit high average Kendall's tau values on all three topological fragment spaces using random products of the space itself as well as compounds from the ZINC database as queries. The average Kendall's tau scores across the internal and external query set for all three topological fragments spaces together with the standard deviation is given in the Supporting Information in Tables S4−S9.

Agreement of Most Similar Compounds. In the preceding paragraphs, we discussed the correlation of the full rankings retrieved by the novel combinatorial topological search approach and the non-combinatorial topological fingerprint search method using the Tanimoto coefficient. In the correlation

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

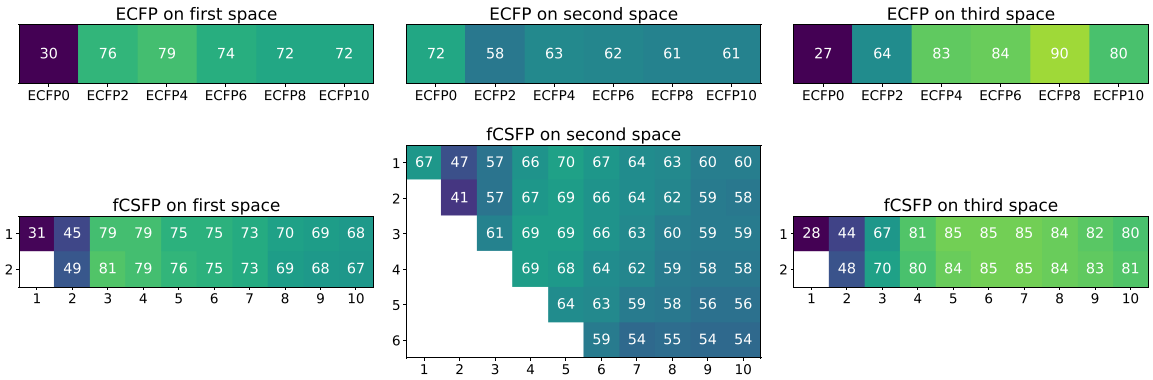


Figure 8. For three topological fragment spaces and the parameterized fingerprint methods chosen for ECFP and fCSFP, the average relative overlap of compounds that appear in the 10 most similar results determined by the combinatorial fragment-based and the non-combinatorial approach across the external rankings are shown. For the fCSFP, the lower bound is given on the *y* axis and the upper bound on the *x* axis. All values are given in percent.

analysis, all products of the topological fragments spaces were included. In the following paragraphs, we analyze the overlap of the 10 compounds that achieved the highest similarity scores by the two approaches. In Figure 7, the average accordance of the 10 highest ranked compounds across all internal ranking pairs is depicted for the ECFP and fCSFP. The corresponding results for the iCSFP and tCSFP are given in Figure S3 in the Supporting Information. For the ECFP, the ECFP4 has the highest average score on all three topological fragment spaces as opposed to the ECFP0, which achieves by far the lowest average score on all three spaces. On the first and third space, the parameterized versions of the fCSFP and tCSFP with an upper bound of three to five are among the best performing parameterizations and for the iCSFP the versions with an upper bound five to six. On the second space, the versions of the fCSFP, iCSFP, and tCSFP with a lower bound from one to five and an upper bound from four to six inhibit the highest average scores. The highest average score across all fingerprint methods and parameterizations is achieved by the fCSFP on all three spaces, tied with the iCSFP on the second space. Again, the highest scores for each fingerprint method are comparable. In Figure 8, the results of the same procedure applied to the external rankings are shown for the ECFP and fCSFP. The corresponding results for the iCSFP and tCSFP are given in Figure S4 in the Supporting information. For the ECFP, the ECFP4 performs best on the first topological fragment space with an average relative overlap of 0.74, and the ECFP0 takes the last place with an overlap of 0.14. On the second space, the ECFP0 achieves by far the highest average relative overlap of 0.72 for the ECFP fingerprint method, and on the third space, the ECFP8 performs best across all parameterized ECFP methods. Regarding the fCSFP, the parameterizations with an upper bound of three and four heavy atoms perform best on the first space and inhibit the highest average relative overlap across all fingerprint methods. On the second space, the fCSFP versions with an upper bound of four and five are among the parameterizations with the highest relative overlap for the fCSFP of 0.66−0.69. On the third space, parameterizations with an upper bound of five to eight heavy atoms achieve the highest average overlap across all fingerprint methods. The iCSFP achieves lower relative overlaps compared to the fCSFP on the first topological fragment space, and configurations with an upper bound of seven heavy atoms perform best for the iCSFP. On the second space, parameterizations of the iCSFP with upper bounds of one to three heavy atoms achieve the best results across all fingerprint methods. On the third space, configurations with upper bounds of 7 to 10 heavy atoms inhibit the best average relative overlaps for the iCSFP. For the tCSFP, versions with an upper bound of five to six heavy atoms achieve the highest scores for the tCSFP on the first topological fragment space and with three to five atoms on the second space. On the third space, the results are best for parameterizations with an upper bound of 5 to 10 heavy atoms. A parameterized version of the fCSFP inhibits the highest score on the first space. On the second space, configurations of the iCSFP perform best and on the third space the ECFP8. The average overlap across the internal and external query set for all three topological fragments spaces together with the standard deviation is given in the Supporting Information in Tables S10−S15.

To conclude all four experiments, we showed that the results of the novel search method are generally highly correlated to the non-combinatorial fingerprint similarity search approach across different fingerprint methods and the highest ranked compounds overlap to a large extend for some parameterizations of each fingerprint method. With increasing upper limit for the number of heavy atoms in a chemical feature, the correlation of the combinatorial and non-combinatorial approach decreased for the fCSFP, iCSFP, and tCSFP. The same holds true for ECFP parameterizations with increasing diameter. As fingerprints parameterized this way consider large, very specific chemical features, they are very sensitive to small local structural changes drastically influencing the fingerprint of the whole compound. In the SpaceLight approach, the fingerprint of each fragment is handled independently. As a result, a small local change in one fragment has only a local impact on the fingerprint of the product. This behavior is described in more detail in the next paragraphs. Following these findings, we recommend the fCSFP2.4 and fCSFP2.5 and additionally the ECFP4 and ECFP6 for the fragment-based search approach since they showed good results across all four experiments discussed in the preceding paragraphs as well as benchmark procedures in earlier studies.[24,29]

Distinction in Molecular Description. Similarity searching is per se not an exact problem definition. Besides assigning a score of 1.0 to identical molecules, the quality of a ranking by similarity remains subjective. While we were aiming at getting as

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

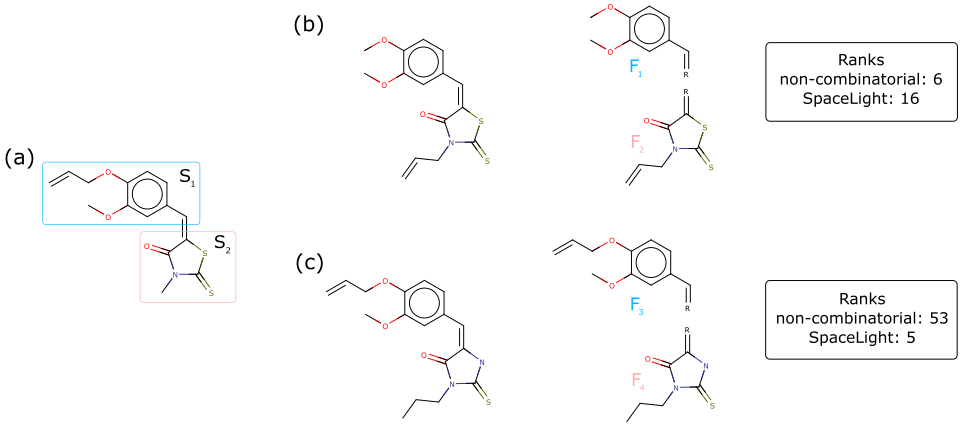


Figure 9. In (a), the ZINC[30] compound with the identifier 2932278 is shown together with the only partition that is topologically similar to the second topological fragment space. The connected substructures of the compound are annotated with S_1 and S_2. In (b) and (c), two fragment combinations contained in the second topological fragment space are depicted together with the fragments forming the combination annotated with F_1, F_2, F_3, and F_4. The colors indicate the assignment of connected substructure and fragment that result in the similarity score derived by the SpaceLight approach. Additionally, the ranks assigned by the non-combinatorial and SpaceLight approach during the ranking experiments when applying the ECFP4 and using the ZINC compound as a query are given for both fragment combinations.

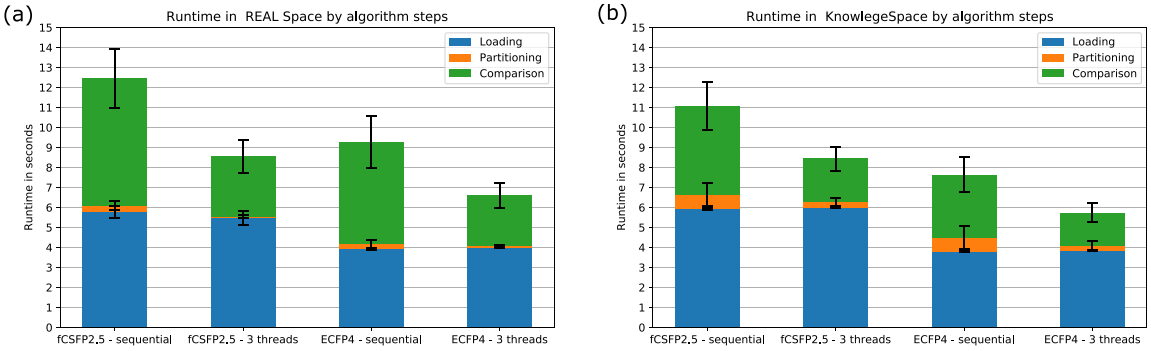


Figure 10. The average run times of the method on the Enamine REAL Space in (a) and the KnowledgeSpace in (b) across 500 randomly chosen compounds from the lead-like subset of the ZINC database are shown for the fCSFP2.5 and the ECFP4 with sequential and parallel computation using three threads. The average run times together with the standard deviation are subdivided into the time needed to retrieve the topological fragment space from disk as well as the partitioning and comparison step of the search procedure.

close as possible to the non-combinatorial search results, we noticed that this is not necessarily advantageous. The SpaceLight approach considers the locality of features in molecules while fingerprints on whole molecules do not. In other words, the SpaceLight approach recognizes whether two functional groups are in the same fragment and therefore topologically close rather than in different fragments and therefore topologically distant. Overall, we see this as a substantial strength of SpaceLight since it, to the best of our experience, reflects the general thinking of chemists.

To further describe this behavior of the SpaceLight approach, we selected the compound ZINC02932278. It is contained in the ranking analysis experiment for the ECFP4 on the second topological fragment space as a comparable compound and shown in Figure 9. The compound ZINC02932278 shown in (a) differs from the fragment combination depicted in (b) only by the position of the terminal alkene group. As non-combinatorial fingerprints only consider the presence or absence of chemical features in the whole compound, this change in the molecule does not result in a big difference of the fingerprint of the ZINC compound and the fragment combination. The SpaceLight approach considers the fingerprints of the fragments and connected substructures individually, and therefore this change in locality of the alkene group has a bigger impact on the corresponding fingerprints. The fingerprint of the substructure S_1 contains an identifier for the alkene group, but the fingerprint of the assigned fragment F_1 does not. Vice versa, the fingerprint of the substructure S_2 lacks this identifier, but the fingerprint of the fragment F_2 contains it. This behavior results in a higher rank assigned to the fragment combination by the non-combinatorial compared to the SpaceLight approach. The fragment combination shown in (c) differs from the ZINC compound shown in (a) by prolonging a carbon chain and transforming the thiazolidine to a diazole. As the five-membered ring is situated in a central position in the ZINC compound as well as the fragment combination, a large number of identifiers generated by the non-combinatorial ECFP4 differ in the two fingerprints resulting in a low rank of 53 of the fragment combination. For

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

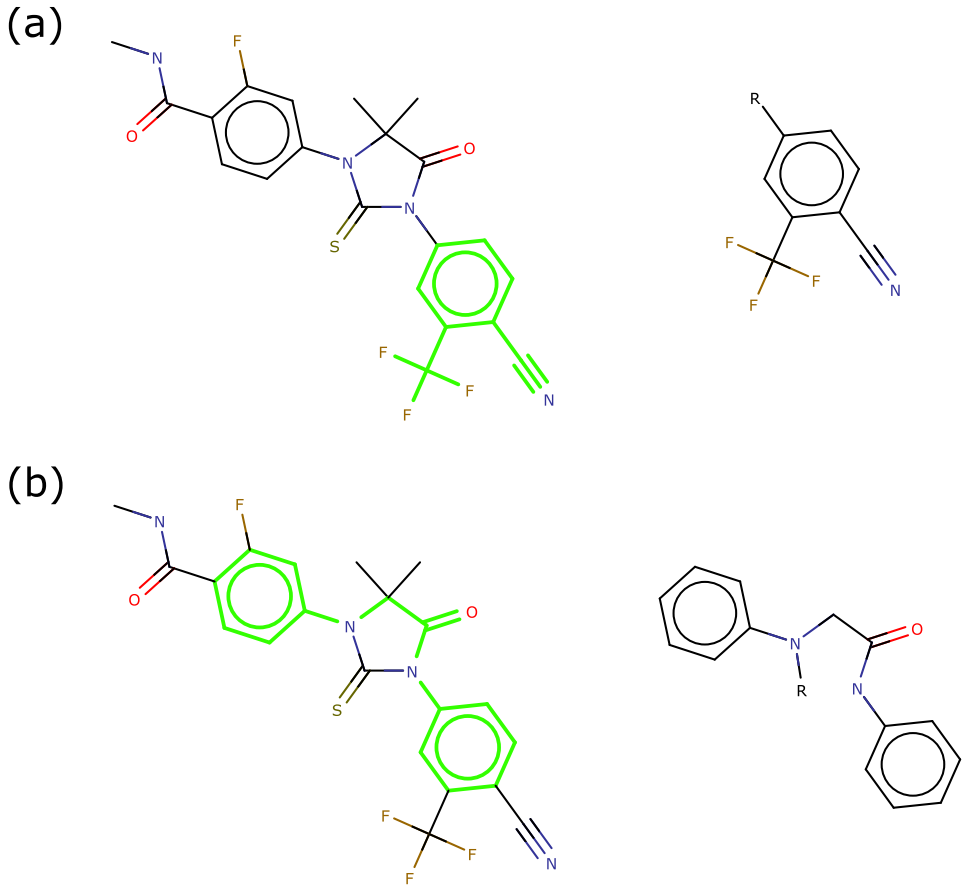


Figure 11. In (a), enzalutamide is shown together with the fragment forming a substructure and containing the largest quantity of heavy atoms retrieved when searching with the fCSFP2.5 in the Enamine REAL Space. In (b), the same is shown when searching with the iCSFP2.5. In both (a) and (b), the corresponding substructure is marked green in enzalutamide.

the SpaceLight approach, the substructure S_1 and fragment F_3 are identical and their fingerprints are not influenced by the transformation of the thiazolidine to a diazole and the altered carbon chain. This results in a higher rank of 5 compared to the non-combinatorial approach. In summary, these examples reflect the consideration of locality of the SpaceLight approach distinguishing it from non-combinatorial fingerprint methods.

Run Times. To measure the time efficiency of SpaceLight, we randomly selected 500 compounds from the lead-like subset of the ZINC database.[30] These compounds were each individually used as queries to search in the whole Enamine REAL Space and the KnowledgeSpace using the fCSFP2.5 and ECFP4 parameterized fingerprint methods. The search is conducted sequentially as well as in parallel using three threads for each of the two fingerprint methods. The experiment is conducted with openSUSE Leap 15 on a 64-bit machine with 16 GB memory and Intel Core i5-6500 CPUs architecture (3.2 GHz). In Figure 10, the average computation time in seconds is shown for the two fingerprint methods and the sequential as well as the parallel SpaceLight run including the time needed to retrieve all necessary data from disk. The average run time is given for the retrieval of the data needed, the time used for the partitioning step and for the comparison step individually. Note that the matching and combination step each took on average only single to double digit milliseconds and are therefore not shown in this figure. In (a), the computation times for searching in the Enamine REAL Space are shown. The time required by the novel search algorithm does on average not exceed 13 s. Since the data retrieval is not conducted in parallel, the average time needed does not differ between the sequential and parallel approaches. The fingerprints generated by the fCSFP2.5 are generally denser than derived by the ECFP4 for the same compound. Consequently, more data has to be retrieved from disk for the search using the fCSFP2.5 in comparison to the ECFP4, resulting in longer time periods needed for loading the topological fragment space together with the precalculated fingerprints of the fragments. For the same reason, the comparison step takes more time using the fCSFP2.5 since the calculation of the Tanimoto coefficient requires more operations. The time required for the partitioning and comparison step is independent of the fingerprint method used and is roughly halved in the parallel approach in comparison to the sequential search. In this experiment, the data was loaded separately for each query compound. However, when multiple compounds are specified as queries by the user, the required data has to be loaded only once. When searching in

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

parallel, only an additional 3 s is required on average by the actual search algorithm for each compound using the fCSFP2.5 or ECFP4. In (b), the computation times when searching in the KnowledgeSpace are depicted. On this space containing 10^{15} products, the SpaceLight approach requires at most 12 s of run time on average. Again, the loading and comparison step require longer time periods when applying the fCSFP2.5 in comparison to the ECFP4 due to the density of the generated fingerprints. In the parallel approach, the time needed during the partitioning and comparison step is roughly halved compared to the sequential approach for the fCSFP2.5 and ECFP4. The KnowledgeSpace contains chemical reactions using three and even four components forming complex ring systems such as the Grieco three-component condensation.[32] The resulting topology graphs are highly connected, and generating topologically similar partitions of the query compound takes longer periods of time compared to topology graphs containing two nodes and one edge, which occur in high quantities in the topological fragment space representation of the Enamine REAL Space. Consequently, the partitioning step of the search algorithm requires on average more time when searching in the KnowledgeSpace compared to the Enamine REAL Space. These highly connected topology graphs are also very selective in the partitions that are topologically similar to it, reducing the number of comparisons between fingerprints of connected substructures of the query compound and fingerprints of fragments that have to be considered in the comparison step of the algorithm. This results in shorter time periods required on average in the comparison step when searching in the KnowledgeSpace. Notably, while the KnowledgeSpace exceeds the Enamine REAL Space by five orders of magnitude in size, the actual search algorithm requires again only roughly 3 s on average when searching in parallel using three threads. This shows the strength of this combinatorial approach when coping with very large chemical spaces since the required run time scales with the number of fragments not the number of products of a combinatorial chemical space. More details on computing times including the average run times and standard deviations required by each step of the algorithm is given in the Supporting Information in Tables S3 and S4.

Employing SpaceLight for Scaffold Detection. For the identification of reaction routes or other virtual screening applications such as scaffold detection, it can be interesting to determine synthesizable and commercially accessible building blocks of large combinatorial libraries that form a substructure of a given compound. As discussed in the Methods section, the fCSFP, iCSFP, and tCSFP fulfill the subset relation using the described fragment representation. Consequently, if a fragment forms a substructure of a query compound, the fragment fingerprint generated by the fCSFP, iCSFP, and tCSFP forms a subset of the fingerprint generated for the query compound. Additionally, through the novel fragment representation, the bond configuration of the building block within its product is captured. Therefore, SpaceLight can be used to identify large substructures in a fragment space by only slight modifications of the scoring functions. In this application, the partitioning of the query compound can be neglected. Instead, the fingerprint of the whole query compound is calculated. Then, for each fragment and all the identifiers forming its fingerprint, the presence in the fingerprint of the query compound is assessed. If the fingerprint of the fragment is fully contained, it forms a substructure of the query compound. This subset assessment can be done efficiently as the fingerprints of all fragments are precalculated and can be retrieved from disk.

In Figure 11, the results of the substructure search are shown for enzalutamide when searching in the Enamine REAL Space. Since no product of the Enamine REAL Space could be retrieved with a score of 1.0 for enzalutamide, it is not contained in the product space. In (a) and (b), the fragment with the largest number of heavy atoms is shown, which has a fingerprint forming a subset of enzalutamide's fingerprint. In (a), the fCSFP2.5 is used for the fingerprint generation. The fCSFP captures valence, aromaticity, number of heavy atom neighbors, and ring membership of atoms. For this reason, the atom and bond configuration of the fragment coincides with the corresponding substructure in enzalutamide. The link atom in the fragment representation marks the position of a reaction, in this case forming a peptide bond to another fragment. This information about potential reaction sites together with commercially available substructure building blocks might aid chemists in the search for synthesis protocols. In (b), the iCSFP2.5 is used to calculate the fingerprints. As the iCSFP only captures valence and connectivity of atoms within each substructure, the retrieved fragment resembles the result of an MCS substructure search.[33] Since the matched nitrogen atoms are part of a ring in enzalutamide but not in the fragment, its fingerprint would not form a subset of enzalutamide's fingerprint when generated with the fCSFP. The results of the substructure search for fluconazole and paroxetine are given in Figures S5 and S6 in the Supporting Information.

CONCLUSIONS

In this work, we presented a novel algorithm named SpaceLight for searching in large combinatorial chemical spaces using fingerprint methods. The results of the search method are in good agreement with classical non-combinatorial fingerprint methods for the ECFP and CSFP. If retrieved compounds diverge, one has to keep in mind that the algorithm captures the locality of features due to the consideration of the compound fragmentation. Although this depends on the viewpoint of the user, we found that this notion of similarity is mostly beneficial.

SpaceLight enables the application of well-known fingerprint methods to combinatorial fragment spaces that go far beyond the size of enumerable chemical libraries for the first time. Even when searching in chemical spaces surpassing billions of compounds, the method can operate within seconds on a standard PC. As it is the case for the FTrees-FS approach published 20 years ago, SpaceLight scales only with the number of fragments involved rather than with the number of products. Therefore, chemical spaces going even beyond 10^{20} molecules can be searched. Notably, SpaceLight is exact in the sense that a molecule identical to the query reaches the maximum score of 1.0, which is extremely seldom for non-identical molecules, at least in non-polymeric chemical spaces. Additionally, the mathematical completeness of the CSFP was used to formulate an efficient approach to detect scaffolds together with potential reaction routes for arbitrary query compounds. Theoretically, other fingerprint methods could be used for the novel fragment-based search method and it would be interesting to incorporate the approach into similarity-driven virtual screening and machine-learning workflows.

So far, only sequential screening was available for similarity searching with a topological fingerprint. The problem can be easily coarse-grained parallelized, and the best software systems available today achieve a throughput of 450 million molecules

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

per second using 16 threads.[18] Even under the assumption that such a method scales without loss, searching the REAL space with 13 billion compounds in about 10 s would require at least 46 threads. To search the KnowledgeSpace with 10^{15} compounds in about 10 s, roughly 3.5 million threads would be needed. Note that this approach does not scale very well since data storage and transfer becomes a bottleneck. Today, spaces with up to 10^{20} compounds already exist.[7] Obviously, similarity searching can only be addressed by a combinatorial approach as SpaceLight today and certainly in the near future.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00850.

Average Kendall's Tau scores and average overlap of highest ranked compounds for the novel search approach and the non-combinatorial fingerprint approach across two different sets of query compounds and three topological fragment spaces; the average computation time and standard deviation of the described method per algorithm step in a sequential and parallel computation on the Enamine REAL and KnowledgeSpace; the results of the analysis workflow applied to the KnowledgeSpace together with the three randomly chosen topological fragment spaces, the enumerated products, and randomly chosen queries; and results of the scaffold detection workflow for fluconazole and paroxetine (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *ZBH-Center for Bioinformatics, Research Group for Computational Molecular Design, Universität Hamburg, Hamburg 20146, Germany;* orcid.org/0000-0002-9553-6531; Email: rarey@zbh.uni- hamburg.de

Authors

Louis Bellmann − *ZBH-Center for Bioinformatics, Research Group for Computational Molecular Design, Universität Hamburg, Hamburg 20146, Germany;* orcid.org/0000-0002-7920-1889

Patrick Penner − *ZBH-Center for Bioinformatics, Research Group for Computational Molecular Design, Universität Hamburg, Hamburg 20146, Germany;* orcid.org/0000-0003-4988-6183

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00850

Notes

The authors declare the following competing financial interest(s): M.R. declares the following competing financial interest: In addition to software availability from Universitt Hamburg, the SpaceLight technology will be licensed to commercial customers by BioSolveIT GmbH of which M.R. is a shareholder.

The SpaceLight tool is available for Linux, MacOS, and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and is free for academic use and evaluation purposes. Furthermore, SpaceLight may be integrated into BioSolveIT's InfiniSee platform available at https://www.biosolveit.de/infiniSee/. The KnowledgeSpace in its topological fragment space representation can be accessed at https://www.zbh.uni-hamburg.de/forschung/amd/datasets.html. For a detailed description of the KnowledgeSpace, see https://www.biosolveit.de/CoLibri/spaces.html#knowledgespace.

■ REFERENCES

(1) Johnson, M.; Basak, S.; Maggiora, G. A Characterization of Molecular Similarity Methods for Property Prediction. *Math. Comput. Model.* **1988**, *11*, 630−634.

(2) Cereto-Massagué, A.; Ojeda, M. J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular Fingerprint Similarity Search in Virtual Screening. *Methods* **2015**, *71*, 58−63.

(3) Mackey, M. D.; Melville, J. L. Better Than Random? The Chemotype Enrichment Problem. *J. Chem. Inf. Model.* **2009**, *49*, 1154−1162.

(4) Dean, P. M.; Lewis, R. A. *Molecular Diversity in Drug Design*; Springer: Dordrecht, 1999.

(5) Clark, A. M.; Labute, P. Detection and Assignment of Common Scaffolds in Project Databases of Lead Molecules. *J. Med. Chem.* **2008**, *52*, 469−483.

(6) *OpenEye Large Scale Virtual Screening*; https://www.eyesopen.com/large-scale-virtual-screening, last accessed on 22/06/2020.

(7) Homann, T.; Gastreich, M. The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discovery Today* **2019**.

(8) Brenner, S.; Lerner, R. A. Encoded Combinatorial Chemistry. *Proc. Natl. Acad. Sci.* **1992**, *89*, 5381−5383.

(9) Cramer, R. D.; Patterson, D. E.; Clark, R. D.; Soltanshahi, F.; Lawless, M. S. Virtual Compound Libraries: A New Approach to Decision Making in Molecular Discovery Research. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1010−1023.

(10) Andrews, K. M.; Cramer, R. D. Toward General Methods of Targeted Library Design: Topomer Shape Similarity Searching with Diverse Structures as Queries. *J. Med. Chem.* **2000**, *43*, 1723−1740.

(11) Cramer, R. D.; Soltanshahi, F.; Jilek, R.; Campbell, B. AllChem: Generating and Searching 1020 Synthetically Accessible Structures. *J. Comput.Aided Mol. Des.* **2007**, *21*, 341−350.

(12) Jilek, R. J.; Cramer, R. D. Topomers: A Validated Protocol for Their Self-Consistent Generation. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1221−1227.

(13) Nicolaou, C. A.; Watson, I. A.; Hu, H.; Wang, J. The Proximal Lilly Collection: Mapping, Exploring and Exploiting Feasible Chemical Space. *J. Chem. Inf. Model.* **2016**, *56*, 1253−1266.

(14) Boehm, M.; Wu, T.-Y.; Claussen, H.; Lemmen, C. Similarity Searching and Scaffold Hopping in Synthetically Accessible Combinatorial Chemistry Spaces. *J. Med. Chem.* **2008**, *51*, 2468−2480.

(15) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with Feature Trees. *J. Chem. Inf. Model.* **2009**, *49*, 270−279.

(16) Detering, C.; Claussen, H.; Gastreich, M.; Lemmen, C. KnowledgeSpace-a Publicly Available Virtual Chemistry Space. *Aust. J. Chem.* **2010**, *2*, 11.

(17) *Enamine REAL Space*; https://enamine.net/library-synthesis/real-compounds/real-space-navigator, last accessed on 20/04/2020.

(18) *NextMove Arthor*; https://www.nextmovesoftware.com/arthor.html, last accessed on 10/07/2020.

(19) Rarey, M.; Stahl, M. Similarity Searching in Large Combinatorial Chemistry Spaces. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 497−520.

(20) *BioSolveIt InniSee*; https://www.biosolveit.de/infiniSee/, last accessed on 10/07/2020.

(21) Schneider, G.; Lee, M.-L.; Stahl, M.; Schneider, P. Novo Design of Molecular Architectures by Evolutionary Assembly of Drug-derived Building Blocks. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 487−494.

(22) Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J. Chem. Inf. Model.* **2018**, *58*, 1194−1204.

(23) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(24) Bellmann, L.; Penner, P.; Rarey, M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* **2019**, *59*, 4625−4635.
(25) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsion: a New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82−85.
(26) Jaccard, P. Lois de Distribution Florale dans la Zone Alpine. *Bull. Soc. Vaudoise Sci. Nat.* **1902**, *38*, 69−130.
(27) Mendez, D.; Gaulton, A.; Bento, A P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M. P.; Mosquera, J. F; Mutowo, P.; Nowotka, M.; Gordillo-Marañón, M.; Hunter, F.; Junco, L.; Mugumbate, G.; Rodriguez-Lopez, M.; Atkinson, F.; Bosc, N.; Radoux, C. J; Segura-Cabrera, A.; Hersey, A.; Leach, A. R ChEMBL: Towards Direct Deposition of Bioassay Data. *Nucleic Acids Res.* **2019**, *47*, D930−D940.
(28) *eMolecules*; https://www.emolecules.com/, last accessed on 17/07/2020.
(29) Riniker, S.; Landrum, G. A. Open-Source Platform to Benchmark Fingerprints for Ligand-based Virtual Screening. *Aust. J. Chem.* **2013**, *5*, 26.
(30) Irwin, J. J.; Shoichet, B. K. ZINC- a Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.
(31) Kendall, M. G. The Treatment of Ties in Ranking Problems. *Biometrika* **1945**, *33*, 239251.
(32) Larsen, S. D.; Grieco, P. A. Aza Diels-Alder Reactions in Aqueous Solution: Cyclocondensation of dienes with Simple Iminium Salts Generated under Mannich Conditions. *J. Am. Chem. Soc.* **1985**, *107*, 1768−1769.
(33) Raymond, J. W.; Willett, P. Maximum Common Subgraph Isomorphism Algorithms for the Matching of Chemical Structures. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 521−533.

https://dx.doi.org/10.1021/acs.jcim.0c00850
J. Chem. Inf. Model. 2021, 61, 238−251

D.3. Comparison of Combinatorial Fragment Spaces and Its Application to Ultralarge Make-on-Demand Compound Catalogs

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

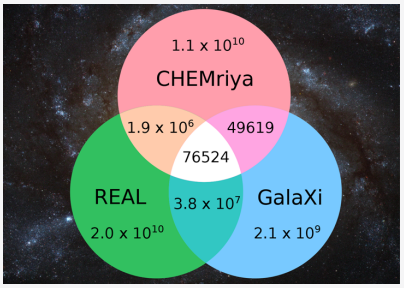
Comparison of Combinatorial Fragment Spaces and Its Application to Ultralarge Make-on-Demand Compound Catalogs

Louis Bellmann, Patrick Penner, Marcus Gastreich, and Matthias Rarey*

Cite This: J. Chem. Inf. Model. 2022, 62, 553−566 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The set of chemical compounds shared by two or more chemical libraries is assessed routinely as means of comparing these libraries for various applications. Traditionally this is achieved by comparing the members of the chemical libraries individually for identity. This approach becomes impractical when operating on chemical libraries exceeding billions or even trillions of compounds in size. As a result, no such analysis exists for ultralarge chemical spaces like the Enamine REAL Space containing over 20 billion compounds. In this work, we present a novel tool called SpaceCompare for the overlap calculation of large, nonenumerable combinatorial fragment spaces. In contrast to existing methods, SpaceCompare utilizes topological fingerprints and the combinatorial character of these chemical spaces. The tool is able to determine the exact overlap of prominent spaces like Enamine's REAL Space, WuXi's GalaXi Space, and Otava's CHEMriya for the first time.



INTRODUCTION

Chemical libraries in physical or virtual form describe the search space in early phase drug discovery. While they came traditionally as enumerated libraries like vendor catalogs, they could also be represented as chemical fragment spaces or chemical knowledge in general. With a growing number of chemical libraries at hand, comparing them is a logical first step for drug discovery projects. A natural analysis in this regard is the size and character of the overlap, e.g., finding all chemical compounds existing in multiple chemical libraries.[1,2] Using this analysis, compounds can be identified that are part of a focused chemical library and are commercially available through a specific vendor or can be obtained from different vendors. In addition, the overlap of chemical knowledge domains can be used to retrieve different synthetic pathways for compounds. Furthermore, the overlap of chemical libraries in general can give insights into their character and relative location in the chemical universe. Various other applications, like patent space analysis or polypharmacology detection, come to mind. For chemical libraries containing up to several million compounds, this analysis can be achieved efficiently by deriving a unique representation of all compounds, e.g., the unique SMILES,[3] and comparing these unique representations for two or more different chemical libraries.

Traditionally, virtual chemical libraries consist of a set of enumerated chemical compounds. This way the resource requirements needed to store and access a chemical library grow roughly linear with the number of chemical compounds present in the library. This characteristic forms a bottleneck for the growth of enumerated chemical libraries, requiring an extensive amount of computational resources for chemical libraries containing up to single digit billions of compounds.[4] A different approach is needed to represent chemical libraries exceeding billions of compounds in size.[5]

Combinatorial chemical libraries, e.g., DNA-encoded libraries,[6] using building blocks and chemical reactions, form a possibly vast product space through combinatorial explosion. Following this design, a virtual combinatorial chemical space can encompass a trillion chemical products or more while only using a limited number of building blocks and chemical reactions. Of these ultralarge virtual combinatorial chemical spaces, three are compound vendor spaces with make-on-demand products, the Enamine REAL Space,[7] the GalaXi Space,[8] and the CHEMriya Space.[9] In addition, several proprietary virtual combinatorial chemical spaces exist.[5,10−13] Beyond these commercial and proprietary chemical spaces, a publicly available[14] space exists that has been created using reactions known from the literature and purchasable building blocks. To date, the largest reported size of any combinatorial chemical space is 10^{26} compounds, which are contained in the GSK Space.[12] This showcases the vast size increase of combinatorial chemical libraries compared to enumerated ones.

Received: November 11, 2021
Published: January 20, 2022

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

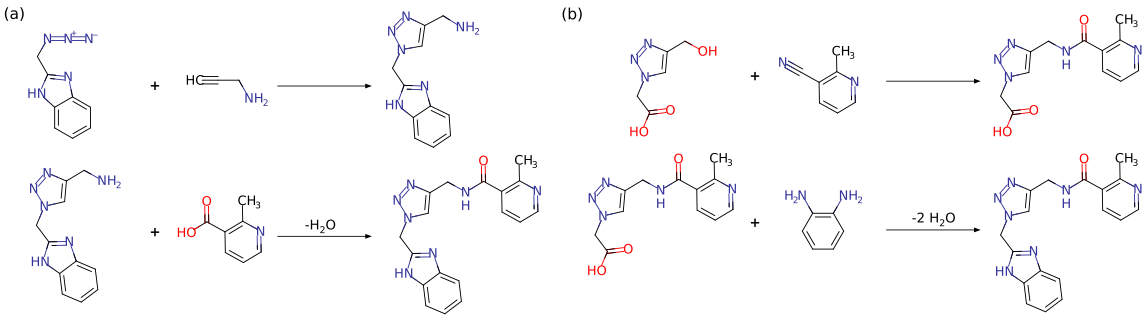


Figure 1. In (a), a reaction scheme is depicted. The product is generated by using a Diels−Alder reaction followed by an amide bond formation. In (b), a reaction scheme generating the same product is shown. A Ritter reaction is applied followed by a condensation reaction.

For these ultralarge chemical spaces, no comprehensive analysis of their overlap exists. It is not known how many compounds are shared between the three vendor spaces. For two or more combinatorial chemical spaces, it is not clear how large their overlap is or if the set of products of one space might even be fully contained in another combinatorial chemical space. As these chemical spaces are practically not enumerable, the strategy for determining the products contained in two or more spaces, using a unique representation for each product individually, is not feasible. In fact, the set of products contained in two nonenumerable combinatorial chemical spaces might itself be too large to be practically enumerated. Obviously, comparing only the building blocks of the chemical spaces is insufficient, since different synthetic routes or reactant combinations might result in the same product. So far, existing methods compare nonenumerable chemical libraries by sampling them around query compounds to obtain enumerable subspaces for comparison.[15]

In this work, we present a novel algorithmic approach which resulted in a tool called SpaceCompare. SpaceCompare utilizes a combinatorial approach based on topological fingerprints and is able to calculate the exact set of products contained in two nonenumerable, ultralarge combinatorial chemical libraries for the first time. We will apply SpaceCompare on three commercial vendor spaces and the freely available KnowledgeSpace. We will show that the overlap between these spaces is surprisingly low such that a further increase in purchasable chemical space can be expected in the future.

METHODS

To process ultralarge chemical libraries while using only a limited amount of computing resources, the novel SpaceCompare method avoids complete enumeration of all products of a combinatorial fragment space. Instead, it operates on the fragments of two fragment spaces and identifies those fragments that cannot be part of the products contained in both spaces. In this context, we describe fragments and products by their chemical substructures. We employ the fragment-based Connected Subgraph Fingerprint (fCSFP),[16,17] as it captures all chemical substructures of a compound and gives a fine-grained molecular description considering element, connectivity, valence, aromaticity, and ring membership of atoms. In addition, if a compound forms a chemical substructure of another compound, the identifiers present in its fCSFP fingerprint form a subset of the identifiers present in the fCSFP fingerprint of the other compound. This way we can efficiently detect fragments that cannot be used for products present in both combinatorial fragment spaces. We distinguish chemical substructures by all properties captured by the fCSFP. Note that stereochemical properties of fragments and products are not considered in this work but could in principal be integrated in the future.

In Figure 1, two synthesis routes are shown generating the same product each using three reactants and two consecutive reactions. Since the generated products are identical, their chemical substructures are identical as well. However, this is not true for the reactants involved in the reaction schemes. In fact, no two reactants share the same set of chemical substructures or even contain the set of the other reactant. So when conducting a reactant-wise comparison, one might falsely conclude that the two reaction schemes do not generate the same product. Following this argument, we have to change the set of chemical substructures we consider. Generally, chemical substructures fall into three categories in this context: *Stable substructures* are contained in both the product and at least one reactant, as they do not change during the reaction. *Changing substructures* are contained in some reactant but not in the product, as they change during the applied reaction scheme. *Crossing substructures* are newly formed during the reaction and contain atoms from different reactants. As a result, they are contained in the product but not in the individual reactants. In this context, the latter two types of chemical substructures need to be addressed. For example, in (a), one reactant contains an alkyne group which is a changing substructure and is not contained in the product. The same holds true for the carboxyl group in (b). In (a), the triazole is formed in the Diels−Alder reaction and is a crossing substructure, as it is part of the product but not contained in any reagent. Note that one reactant of the reaction scheme in (b) does contain the triazole so it is not a crossing substructure for the reaction scheme shown in (b). Vice versa, benzimidazole is a crossing substructure in (b), as it is formed during the condensation reaction. Again, the benzimidazole is part of one reactant shown in (a). This example shows the complexity of finding the overlap between products when the fragment space can only be analyzed in its condensed combinatorial description.

Topological Fragment Spaces. Topological fragment spaces represent a chemical space in a compact form, enabling efficient search in its products avoiding enumeration.[17] They describe the topology of products resulting from the reaction schemes employed in the space. This topology is captured in a topology graph which nodes represent each pool of reactants that can be used interchangeably at a specific position within

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

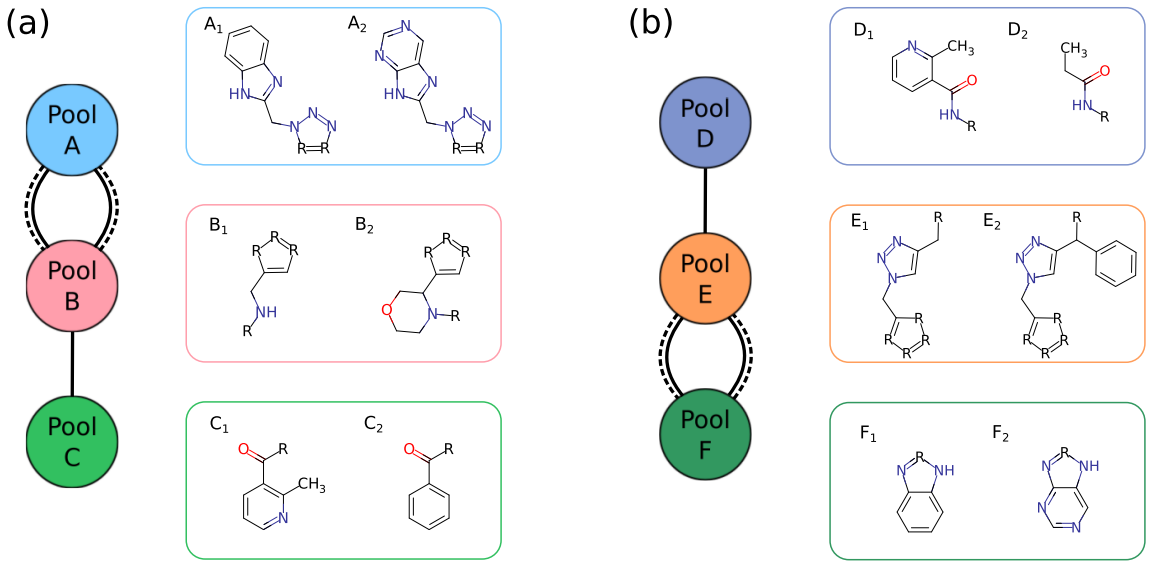


Figure 2. In (a), a topological fragment space is depicted that was generated with the reactions of Figure 1(a). Similarly in (b), a topological fragment space is depicted that was generated with the reactions of Figure 1(b). The edges between pool A and pool B and between pool E and pool F represent chemical aromatic bonds in the generated products. The edges between pool B and pool C and between pool D and pool E represent chemical single bonds. The fragment representations in the colored boxes represent the reactants for the pools with the same color. All link atoms are marked with the label 'R'.

the reaction scheme. The edges represent the newly formed chemical bonds between the reactants in the reaction scheme. Additionally, the reactants are represented as fragments containing link atoms describing their chemical environment once they are part of a product. Each product of the topological fragment space is generated by selecting a topology graph and exactly one fragment from each of its nodes. The link atoms are removed from the selected fragments. Finally, the information from the edges of the topology graph is retrieved to generate the newly formed bonds between the fragments during the represented reactions. As the link atoms act as placeholders for the applied chemical reactions, the chemical bond configuration and valence of each fragment's heavy atoms remain unchanged in this process. A building block's fragment representation contains information about the building block itself as well as the reaction the building block is applied to. As a result, a building block can have multiple differing fragment representations if the building block is used in different reactions.

In Figure 2, two topological fragment spaces are shown that represent the reaction schemes depicted in Figure 1. Note that fragments A_1, B_1, and C_1 represent the reactants of the reaction scheme (a) in Figure 1, and fragments D_1, E_1, and F_1 represent the reactants of the reaction scheme (b) in Figure 1. For each position in both reactions, an alternative reactant was added. The configuration of link atoms and their heavy atom neighbors in the fragments resembles the chemical environment of building blocks once they are part of a product. The fragment B_1 represents the building block 2-propynamine of the reaction scheme (a) in Figure 1. The 5-membered ring resembles the triazole formed during the Diels–Alder reaction and is marked as aromatic. It contains the two carbon atoms from the alkyne group of 2-propynamine together with three link atoms to fill up the ring and act as placeholders for the remaining heavy atoms of the triazole. The link atom at the amine acts as a placeholder for the heavy atom that will be attached via a single bond in the amide bond formation. If B_1 is selected to form a product, its link atoms will be removed, and the triazole and amide bond will be generated using B_1 together with the other two selected fragments and the edges of the topology graph shown in (a). Due to the pre-existing link atoms, the chemical bond configuration, aromaticity, and valence do not change in the process for the amine and the two carbon atoms representing the alkyne group.

The fragment representation uses link atoms and an adjusted chemical bond configuration to capture the chemical environment of the reactant within a product. As a result, they do not contain changing chemical substructures anymore. However, crossing chemical substructures like the benzimidazole are still missing in all fragments of the topological fragment space shown in (b) of Figure 2.

Crossing Substructures. We now address crossing chemical substructures that are formed during the reaction and are contained in a final product but no fragment representation of a reactant generating it. In Figure 3, the product of Figure 1 is shown together with the fragments A_1, B_1, and C_1 of Figure 2(a) generating it. The highlighted amide group in the product is a crossing substructure, as it is not contained in any of the fragments. However, notice that the smaller chemical substructures of the carbonyl group and nitrogen atom are each contained in one fragment adjacent to a link atom. In general, due to the design of the fragment representation, a subdivision into two or more parts exists for each crossing substructure, such that each part forms a chemical substructure contained in a fragment adjacent to a link atom.

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553–566

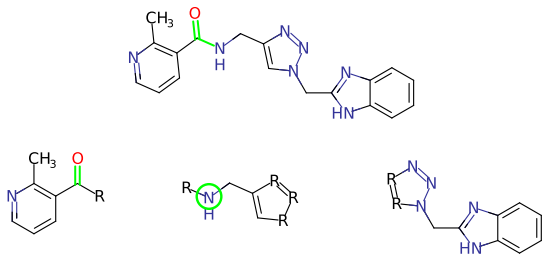


Figure 3. A fragment combination (bottom) together with its derived product (top) is shown. Additionally, the amide group forming a crossing substructure and the corresponding carbonyl group and nitrogen within the fragments are highlighted in green. All atoms incident to the highlighted bonds are considered as part of the substructure.

We can use this property to efficiently enumerate all crossing substructures over all fragment combinations of a topological fragment space even if the crossing substructures are distributed over more than two fragments. First, we traverse all fragments and enumerate all chemical substructures having at least one link atom as a direct neighbor. We then summarize these substructures across fragments from the same pool using the fCSFP identifier[16] for each chemical substructure. For example, the carbonyl group adjacent to a link atom is present in both fragments C_1 and C_2. The two fragments can be used interchangeably in the topological fragment space. So each crossing substructure containing this carbonyl group for a fragment combination using C_1 can be generated the same way by exchanging C_1 with C_2 in the fragment combination and using the carbonyl group contained in C_2. In the second step, we combine the summarized chemical substructures at link atoms to form crossing substructures. This way considering individual fragment combinations can be avoided. Fragments A_1 and A_2 contain the same part of the triazole with the same configuration of heavy atom neighbors. The same holds true for fragments B_1 and B_2. So we can deduce that all fragment combinations of the topological fragment space of Figure 2(a) contain the triazole with the same configuration of heavy atom neighbors without considering its eight fragment combinations individually.

Extended Chemical Substructure Set. We define the *extended chemical substructure set* of a fragment as the set of all chemical substructures present in the fragment together with all crossing substructures of a fragment combination containing that fragment and intersecting it. The extended chemical substructure set of a fragment is represented by its extended fCSFP fingerprint containing an identifier for each of its chemical substructures. The chemical substructures of a product form a subset of the union of extended chemical substructure sets of the fragment combination generating the product. This property will allow SpaceCompare to describe all chemical substructures of products while only operating on the fragment level. In Figure 4, the fragment C_1 from the topological fragment space depicted in Figure 2(a) is shown together with two crossing substructures of its extended chemical substructure set. Both of them are amide groups. In one case, the nitrogen atom of the amide group has one attached hydrogen. In the other case, it is part of a ring and has in total three heavy atom neighbors. The information about the types and counts of adjacent chemical bonds to heavy atom neighbors of the chemical substructures is contained in its identifier within the extended fCSFP fingerprint. Note that no product of the topological fragment space contains both chemical substructures. So the union of extended chemical substructure sets of the fragments A_1, B_1, and C_1 forms a proper superset of the chemical substructures of the product shown in Figure 1. The chemical substructures of the product form a superset of all chemical substructures contained in the individual fragments A_1, B_1, and C_1. The same holds true for fragments D_1, E_1, and F_1. As both fragment combinations generate the same product, the union of extended chemical substructure sets of the fragments A_1, B_1, and C_1 forms a superset of all chemical substructures contained in the fragments D_1, E_1, and F_1 and vice versa. This property of extended chemical substructure sets will play a key role in the SpaceCompare algorithm. A fragment is *covered* by a fragment combination if all its chemical substructures are contained in the union of extended chemical substructure sets of the fragment combination.

SpaceCompare Algorithm. The SpaceCompare algorithm detects all products that are contained in both of the topological fragment spaces. To compute this overlap efficiently, SpaceCompare operates only on fragments that are covered by fragment combinations of the other topological fragment space. This way the potential combinatorial explosion resulting from large sized product spaces is avoided. The resulting set of products being part of both spaces will appear

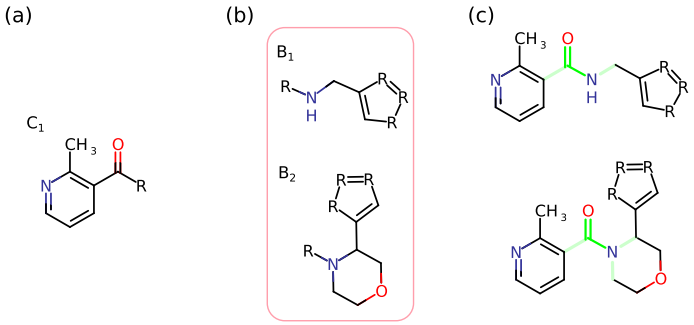


Figure 4. In (a), the fragment C_1 from Figure 2(a) is shown. In (b), the fragments for pool B from Figure 2(a) are depicted. In (c), the combinations of fragments C_1 and B_1 and fragments C_1 and B_2 are shown. Two crossing substructures contained in the extended chemical substructure set of fragment C_1 are highlighted in green. Their adjacent chemical bonds in the fragment combinations are highlighted in light green.

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

in its enumerated form and be validated via their unique SMILES representation. Therefore, the required computing resources are output sensitive, i.e., the algorithm can only succeed as long as the resulting intersection is tractably enumerable.

We briefly summarize the method and give a more detailed description of each step with concrete examples below. The method consists of two steps:

1. A *covering step* in which initially all fragments of the first topological fragment space are determined that can be covered by a fragment combination of the second space. This process consists of three intermediate steps.
   1.1. The extended fCSFP fingerprint is calculated for all fragments of the second topological fragment space.
   1.2. For each fragment of the first topological fragment space, all fragment combinations of the second topological fragment space covering it are determined recursively. Fragments with at least one covering fragment combination are stored together with their covering combinations.
   1.3. All topology graphs of the first topological fragment space are assessed. If a topology graph contains no covered fragments for at least one of its pools, all covered fragments of that topology graph are removed.

   The process is repeated in an inverse manner such that fragments of the second space are determined that can be covered by a fragment combination of the first space. Here, only fragments from the first space are used that were covered in the initial part of the covering step.

2. A *combination step* in which initially all combinations of fragments from the first space are determined that can be covered by at least one shared fragment combination of the second space in the initial part of the covering step. Products are generated and stored using a unique SMILES.[3] Similarly the process is repeated in an inverse fashion with the roles of the first and second space flipped. Here, only fragments from the first space are used for covering that are part of some product generated in the initial part of the combination step.

In the following, we describe the two steps of the algorithm in detail.

Covering Step. As discussed earlier, the extended fCSFP fragment fingerprints will be used to operate on a fragment level and avoid product enumeration as much as possible. Consider a product contained in the overlap of the two topological fragment spaces and two fragment combinations from the different spaces generating that product. Each individual fragment is covered by the fragment combination of the other topological fragment space. Conversely, we can neglect fragments from the overlap calculation for which no covering fragment combination from the other topological fragment space exists. The goal of the covering step is to determine all fragments from one space that have a fragment combination from the other space covering it. We first give a simplified example of the covering step before we describe all involved substeps extensively. In Figure 5, the general concept of the covering step is depicted. The fCSFP fingerprints of all fragments contained in the first topological fragment space are retrieved from its database. Afterward, the extended fCSFP

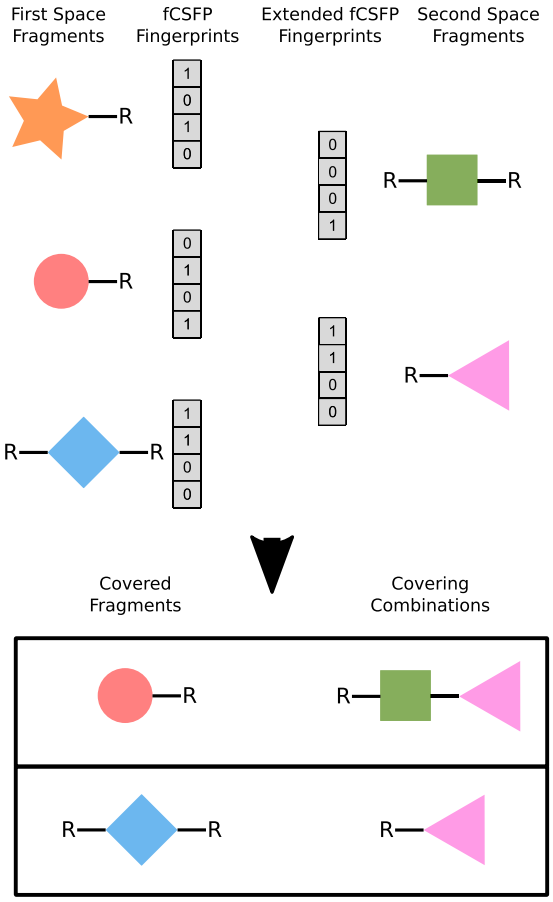


Figure 5. An overview of the covering step of SpaceCompare. The fragments are depicted by the different shapes together with link atoms indicated by the 'R' group. The fCSFP and extended fCSFP fingerprints are shown as bit vectors. In the lower part of the picture, the resulting data structure showing all covered fragments together with their covering fragment combinations is shown.

fingerprints of all fragments contained in the second topological fragment space are calculated. Now we determine all potential coverings of the fragments of the first space using the retrieved and calculated fingerprints. The star-shaped fragment has an fCSFP fingerprint with a set bit at the third position. No extended fCSFP fingerprint from the second space has this bit set. As a result, the star-shaped fragment cannot be covered. The circle-shaped fragment can be covered by combining both fragments from the second space. No single extended fingerprint contains all of its bits individually. Lastly, the blue rectangular-shaped fragment has exactly the same bits set in its fCSFP fingerprint as the triangle-shaped fragment from the second space. As a result, it can be covered by just this fragment. The resulting data structure of covered fragments and covering fragment combinations is the final result of the covering step of SpaceCompare.

Now we will give a more detailed and technical description of the individual substeps. We first calculate the extended fCSFP1.6 fingerprint for each fragment of the second topological fragment space. This can be done efficiently

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

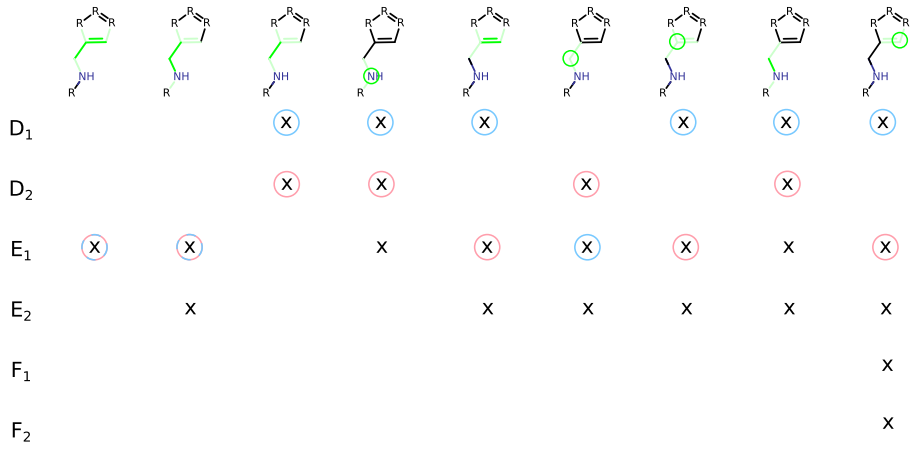


Figure 6. At the top, all crossing substructures represented by the fCSFP1.3 fingerprint of fragment B_1 from Figure 2(a) are shown. The substructure is highlighted in green, and the adjacent chemical bonds to heavy atom neighbors are highlighted in light green. An 'x' marker appears under the crossing substructure for a specific fragment from Figure 2(b), if the corresponding identifier is contained in its extended fCSFP1.3 fingerprint. In the covering step, two coverings of B_1 are found indicated by the colors magenta and turquoise. The 'x' markers are encircled in these colors if the corresponding candidate fragment was used to update or keep one of the fragment combinations during the traversal.

following the strategy described in subsection 'Crossing Substructures'. The bounds one and six were chosen for the fCSFP to balance the compactness and number of chemical substructures captured by a fragment's fingerprint. The fragments of the second topological fragment space containing a given identifier in their extended fCSFP1.6 fingerprint *cover* the identifier and are called the *candidate fragments* for that identifier. These candidate fragments are stored for each identifier for later efficient retrieval.

We can now use the generated extended fCSFP fingerprints to detect fragments from the first topological fragment space that can be covered. As a first filter, all fragments of the first topological fragment space that contain an identifier without any candidate fragments can be neglected. This identifier is not covered by any fragment from the second space, and as a result, the fragment itself cannot be covered. We process each remaining fragment individually and traverse the identifiers of its fingerprint to find all fragment combinations of the second space covering it.

Identifier Traversal. For the description of the identifier traversal, we use a single example fragment of the first topological fragment space. All other fragments are processed individually in the same way. We sort the identifiers of its fingerprint by the ascending number of candidate fragments and traverse them in this order. The goal of the procedure is to determine all combinations of fragments such that at least one of them is a candidate for each traversed identifier. These fragment combinations precisely cover the fragment of the first topological fragment space. During the traversal, the combinations of candidate fragments which contain all of the already traversed identifiers in the union of their extended fCSFP1.6 fingerprints are stored. When considering a new identifier during the traversal, we check all stored fragment combinations if they already cover this identifier or can be combined with a candidate fragment for the current identifier. This way we extend the set of covered identifiers until all identifiers were considered. For a given stored fragment combination, three cases occur.

1. One of its fragments is a candidate for this identifier, meaning it contains the identifier in its extended fCSFP1.6 fingerprint. The stored fragment combination stays unchanged.
2. At least one candidate fragment for this identifier is compatible with the stored fragment combination. An updated version of the stored fragment combination is generated for each compatible candidate fragment by adding it to the existing combination.
3. No candidate fragment is compatible to the stored fragment combination. The combination is removed.

A candidate fragment for this identifier is compatible with the stored combination of candidate fragments, if it belongs to the same graph but not the same fragment pool for each fragment of the combination. This is because products can only be generated by picking a single fragment from each pool of a topology graph. Additionally, we check if the candidate fragment could be added earlier to the combination as a substitution for one of its fragments. This way we avoid duplicates or nonminimal covering fragment combinations.

For this purpose, we determine the set of all already traversed identifiers for which the current candidate fragment was a candidate as well. We establish the corresponding set for all fragments of the combination. If this set, derived for the current candidate fragment, forms a superset of one of the sets derived for the fragments of the combination, it could have been added earlier to the combination. In this case, the candidate fragment covers all identifiers covered by the fragment of the combination and additionally the currently considered identifier. The candidate fragment is incompatible and will not be added to the combination.

The procedure terminates when all identifiers of the fragment's fingerprint were traversed. The stored combinations of candidate fragments give precisely the fragment combinations covering the considered fragment. This process is repeated for each fragment of the first topological fragment space.

In Figure 6, the traversal procedure is shown for the fragment B_1 from Figure 2(a). For simplicity, the fCSFP1.3 is

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

used instead of the fCSFP1.6. Only the fragment E_1 from Figure 2(b) contains the 3-membered carbon chain with two aromatic carbons, and this configuration of adjacent chemical bonds to heavy atom neighbors. Consequently, each covering fragment combination must contain E_1. The second identifier during the traversal represents an aliphatic nitrogen with two adjacent single bonds to heavy atom neighbors. The identifier is covered by E_1 and E_2. Since the only currently stored fragment combination contains E_1, this combination is kept and not updated. The third traversed identifier is covered by D_1 and D_2. As they are both compatible to the stored fragment combination, two updated combinations $\{E_1, D_1\}$ and $\{E_1, D_2\}$ are generated. For the rest of the traversal, these two fragment combinations are not updated. Both fragment combinations are minimal, meaning that the removal of a fragment from the combination would result in the combination not covering B_1 anymore. However, since both fragment combinations do not contain a fragment from pool F, F_1 or F_2 can be freely added to both combinations resulting in a nonminimal covering combination for B_1.

Cleaning and Inversion. At the end of the initial covering step, we perform a cleaning procedure. Since a fragment from each pool of a topology graph is needed to form a product, all of its nodes need to contain at least one covered fragment to contribute to the overlap. Hence, we remove all covered fragments of the first topological fragment space that belong to a topology graph not fulfilling this property. The remaining covered fragments together with their covering fragment combinations are stored in a single data structure.

For the inverse covering step, the same process is repeated with the roles of the first and second topological fragment space switched. Now, coverings of all fragments of the second topological fragment space are determined. Only covered fragments can contribute to products in the overlap of the two spaces. Following this reasoning, fragments of the first topological fragment space are used as candidate fragments during the traversal only if they were covered themselves in the initial covering step. In the end of the inverse covering step, the same cleaning procedure is performed.

Combination Step. In the previous step, we calculated all covered fragments in both spaces together with their covering fragment combinations. In this step, we determine all products for which the underlying fragments all share a covering fragment combination. These products form the candidates for the overlap of the two spaces. Similar to the covering step an initial and inverse version of the combination step is performed. We calculate the number of combinations of covered fragments for both topological fragment spaces. We then select the space where this number is smaller as the first space in the initial combination step. This way computation run time is reduced.

In Figure 7, a simplified example of the combination step is shown. We use the data structure of covered fragments and covering combinations from the covering step and try to find a fragment combination that covers the circle-shaped as well as the star-shaped fragment. They both have a covering fragment combination, but these combinations are incompatible because they contain a different fragment from pool D. As a result, they cannot be combined to a common covering combination, and the two covered fragments from the first space do not share a covering combination. The circle-shaped and rectangular-shaped fragments have covering fragment combinations that are compatible. The covering combinations both contain the same fragment for pool D. The common covering fragment combination is generated and shown at the bottom right. We combine the circle-shaped and rectangular-shaped fragments to a product and generate its uSMILES.[3] The product is a candidate for the overlap, and if we find a candidate with the same uSMILES in the inverse covering step, it is in fact a product contained in both spaces.

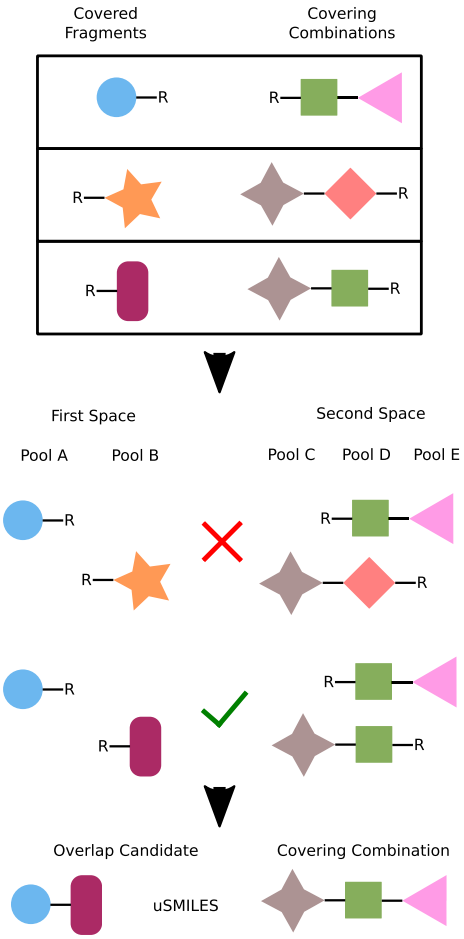


Figure 7. An overview of the combination step of SpaceCompare. The fragments are depicted by the different shapes together with link atoms indicated by the 'R' group. Final products have no link atoms attached. In the upper part of the picture, the data structure showing all covered fragments together with their covering fragment combinations is shown.

We now describe the combination step in more detail. Covered fragment combinations from the first topological fragment space are recursively enumerated. We store all fragment combinations from the second topological fragment space that cover all fragments of the currently considered enumerated covered combination from the first space. When considering a covered fragment from the first topological fragment space for addition to the currently enumerated covered combination, we retrieve its covering fragment combinations derived in the covering step of the algorithm.

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

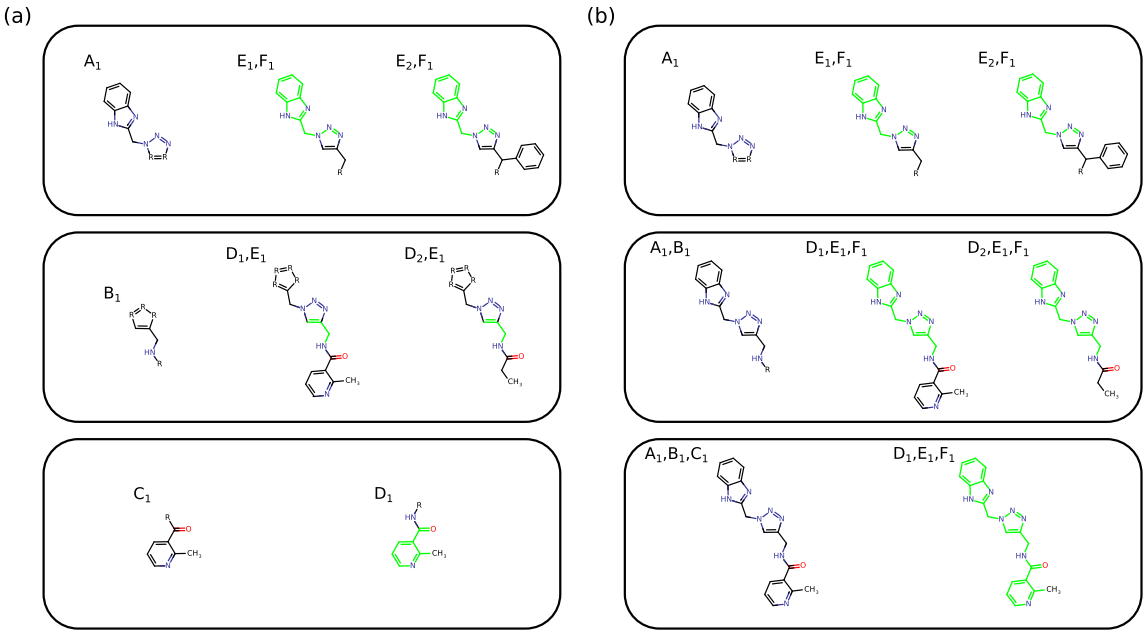


Figure 8. In (a), the fragments A_1, B_1, and C_1 from Figure 2(a) are shown together with their covering combinations of fragments from Figure 2(b) in one box. The chemical substructure corresponding to the covered fragment is highlighted in green within the covering fragment combinations. In (b), the enumeration of the covered fragment combination with fragments A_1, B_1, and C_1 is depicted. Each box indicates one recursive call with the enumerated covered fragment combination on the left and the stored covering fragment combinations on the right. The chemical substructure corresponding to the enumerated covered fragment combination is highlighted in green within the stored fragment combinations.

We call these combinations *candidate fragment combinations* for the considered fragment of the first space.

Fragment Combination Compatibility. Our goal is to update the covering fragment combinations of the second space such that they cover the newly considered fragment of the first space as well. In this context, we want to merge fragment combinations to larger ones that ultimately form products. Two fragment combinations are *incompatible* if they do not belong to the same topology graph or contain a different fragment for some pool of their shared topology graph. Two incompatible combinations cannot be merged to form a product. We define four types of relations between two compatible fragment combinations x and y. First of all, x can be *identical*, *smaller*, or *larger* than y. This is the case if the set of fragments contained in x is identical, a proper subset, or a proper superset of the set of fragments contained in y. If x is larger than or identical to y, it contains all of its fragments and covers all combinations that y covers. If x contains a fragment not contained in y and vice versa, the fragment combinations x and y are *additive*. In this case, x and y can both cover some combination that the other does not cover.

We will now use these definitions to update the stored covering fragment combinations of the second space using the candidate combinations covering the currently considered fragment of the first space. If a stored covering fragment combination has no compatible candidate fragment combination, it is removed, as it cannot be updated to cover the currently considered fragment of the first space. If it is identical or larger than at least one candidate fragment combination, it is kept, and the next stored fragment combination is considered. Otherwise, updated versions of the stored fragment combination are generated by merging it with all additive and bigger candidate fragment combinations.

If no stored covering fragment combinations remain, the recursion stops for this covered fragment combination. If during the enumeration a covered fragment combination is full, meaning it contains a fragment for each pool of its topology graph, the corresponding product is generated. Its unique SMILES[3] is derived and stored for later overlap calculation. Afterward, the recursion stops for this covered fragment combination. In addition, we mark fragment combinations in the initial combination step to reduce the computing time for the inverse covering step. For a fully covered fragment combination we mark all covering fragment combinations in the inverse covering data structure, that are smaller or equal to it. Only a combination that was marked this way can lead to a product in the final overlap of the two topological fragment spaces.

Once the initial covering step is finished, the roles of the first and second topological fragment space are switched for the inverse covering step. Here, only covering fragment combinations of the first topological fragment space are considered as candidate combinations if they were marked in the initial covering step. Products are only stored if their unique SMILES coincides with that of at least one product stored in the initial combination step. The unique SMILES that were stored in both the initial and inverse combination steps give precisely the set of products contained in both topological fragment spaces.

In Figure 8, the recursive call for the fragment combination containing fragments A_1, B_1, and C_1 from Figure 2(a) is shown. In the first step, the candidate fragment combinations $\{E_1, F_1\}$

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

and $\{E_2, F_1\}$ for fragment A_1 are stored. In the second step, these two stored fragment combinations are checked for compatibility with the candidate fragment combinations for the fragment B_1. Both candidate fragment combinations are additive to the combination $\{E_1, F_1\}$, and the two updated fragment combinations are stored. The combination $\{E_2, F_1\}$ is incompatible to both candidate fragment combinations, as it contains a different fragment for pool E. The combination is removed. B_1 is added to the covered fragment combination as at least one compatible pair of fragment combinations was found. In the next step, the two newly stored covering fragment combinations are compared with the candidate fragment combination $\{D_1\}$ for the fragment C_1. $\{D_1\}$ is smaller than the stored fragment combination $\{D_1, E_1, F_1\}$, and therefore, this stored fragment combination is kept. The other fragment combination is incompatible to $\{D_1\}$, as it contains a differed fragment for pool D. Again, C_1 is added to the covered fragment combination as at least one compatible pair of stored and candidate fragment combination was found. The resulting covered fragment combination is full; so the corresponding product is generated, and its unique SMILES is calculated and stored.

RESULTS

We analyze the results of SpaceCompare using four large chemical spaces and assess its computing time and memory requirements. In addition, we generate chemical subspaces that are small enough for a classic one-by-one overlap calculation using unique SMILES or fingerprints and compare the results to the output of SpaceCompare.

Chemical Spaces. In this work, three chemical vendor spaces together with a public chemical knowledge domain space are considered. The Enamine REAL Space[7] contains more than 20 billion commercially available compounds by applying 156 synthesis protocols to over 104 000 building blocks. The compounds are generated by combining two or three building blocks. Otava's CHEMriya[9] on-demand chemical space is based on 30 000 building blocks and 44 in-house reactions. In the reactions, two to four building blocks are combined to products. WuXi designed the GalaXi[8] virtual chemical space by using 30 different reaction types and more than 155 000 building blocks. The space is comprised of 2.3 billion compounds which consist of two or three building blocks. In addition, the publicly available KnowledgeSpace[14] contains over 10^{14} compounds. The space is generated using building blocks from the eMolecules[18] collection together with 117 reactions known from the literature. Its products contain two to five building blocks.

Validation. We validate SpaceCompare by showing that its output is in accordance with the overlap generated by a classic pairwise one-by-one product comparison of two enumerated compound lists. As the four chemical spaces are too large to be practically enumerated, we generated smaller subspaces for the validation procedure.

A single reaction together with the compatible building blocks forms a chemical subspace of the full chemical space. For each space, the reactions are identified for which the chemical subspace contains at most 100 000 products. All of these reactions are used collectively to form a single larger chemical subspace for each of the four chemical spaces. The resulting larger chemical subspace contains all products of the chemical subspaces defined by the identified reactions. The combined chemical subspace for the Enamine REAL Space contains 1 036 275 compounds, the subspace for the GalaXi Space contains 55 672 compounds, the subspace for the CHEMriya Space contains 279 700 compounds, and the subspace for the KnowledgeSpace contains 787 042 compounds. For all four chemical subspaces, we enumerated the products, and the six pairwise overlaps are used for validation.

We will later see that a large portion of the overlap of the Enamine REAL Space, GalaXi Space, and CHEMriya Space is part of the overlap of the chemical subspaces generated by the reactions m_22bba from the Enamine REAL Space, WXVL001 from the GalaXi Space, and S_R001 from the CHEMriya Space. For each of the three chemical subspaces, we randomly selected 1000 building blocks from each pool and generated three smaller subspaces each containing 1 million compounds that we enumerated. The three pairwise overlaps are used for validation.

All nine pairwise overlaps can be calculated in a straightforward manner using the unique SMILES representation of all products. In addition, we calculated the ECFP20[19] fingerprint of all products and generated the overlap as the set of products sharing their fingerprint with a product from the other enumerated chemical subspace. We chose 20 as the diameter of the ECFP to avoid false-positives. Some products contained in the chemical subspaces used for validation contain carbon chains of 19 or 20 carbon atoms. The ECFP18 fingerprint of a 19-membered carbon chain is identical to the ECFP18 fingerprint of a 20-membered carbon chain. This leads to a false-positive in the enumerated overlap calculation using the ECFP fingerprint. The results of SpaceCompare for all nine pairwise comparisons are identical with both enumerated overlap calculation approaches. For reproducibility, the same experiment was repeated for the only two reactions used in the combined chemical subspace of the KnowledgeSpace that have a nonempty overlap of their generated products. The results together with the two topological fragment spaces and their enumerated products are given in the Supporting Information.

Overlap Analysis. We assess the overlap calculated with SpaceCompare for the full Enamine REAL Space and GalaXi Space together with the KnowledgeSpace and CHEMriya Space. In Figure 9(a), the pairwise and overall overlap is shown for the Enamine REAL Space, GalaXi Space, and Knowledge-

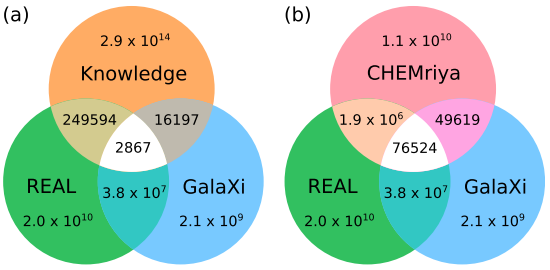


Figure 9. Overlap of the Enamine REAL Space (green) and the GalaXi Space (turquoise) with the KnowledgeSpace (orange) in (a) and with the CHEMriya Space (magenta) in (b) is shown. The numbers in the spheres indicate the number of products contained in the chemical spaces. The numbers in the overlap of the spheres indicate the overlap of the corresponding two or three chemical spaces. E.g., the pink overlap in the right of (b) indicates the set of compounds contained in the CHEMriya Space and GalaXi Space and not in the Enamine REAL Space.

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

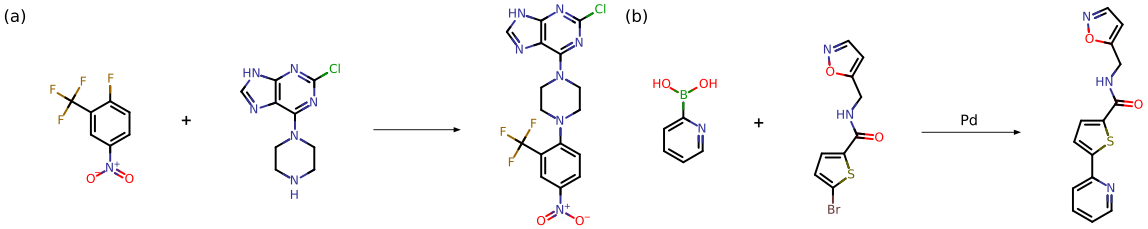


Figure 10. Two products contained in the overlap of the Enamine REAL Space, GalaXi Space, and KnowledgeSpace. For both products, one reaction scheme retrieved from the KnowledgeSpace is depicted. In (a), the nucleophilic aromatic substitution reaction scheme and in (b) the Suzuki coupling reaction scheme.

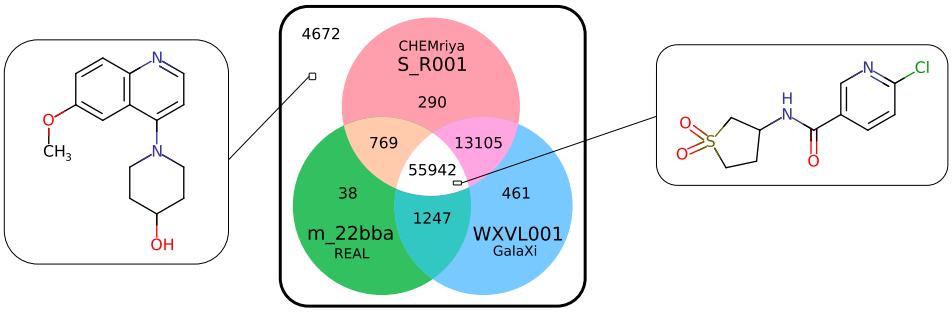


Figure 11. Composition of products contained in the overlap of the Enamine REAL Space, GalaXi Space, and CHEMriya Space is shown in the central Venn diagram. The three spheres indicate products of this overlap that are contained in the three chemical subspaces generated by the reactions m_22bba for the Enamine REAL Space, WXVL001 for the GalaXi Space, and S_R001 for the CHEMriya Space. The overlap of the spheres indicates the corresponding overlap of these chemical subspaces, e.g., the orange overlap indicates that 769 compounds are contained in the m_22bba and S_R001 subspaces but not in the WXVL001 subspace. To the right, one product contained in all three of the chemical subspaces is depicted. The white area within the central box around the spheres represents the products contained in the overlap of the Enamine REAL Space, GalaXi Space, and CHEMriya Space that are not part of any of the three chemical subspaces. To the left, one of these products is shown.

Space. Although the KnowledgeSpace exceeds the Enamine REAL Space and the GalaXi Space by several orders of magnitude in size, its overlap with the two spaces individually and collectively contains less than 0.002% of each chemical space and is orders of magnitude smaller than the overlap of the Enamine REAL Space and the GalaXi Space. Although different reaction schemes can result in the same product, this small overlap might be caused by the difference in the chemical reactions employed in the KnowledgeSpace compared to the Enamine REAL Space and GalaXi Space. Many products of the KnowledgeSpace consist of four or five building blocks. The Enamine REAL Space and GalaXi Space use at most three building blocks in their reactions. The overlap of the GalaXi Space and Enamine REAL Space consists of over 38 million compounds that are commercially available through WuXi as well as Enamine. Although the overlap appears to be large in size, it contains less than 2% of the compounds of the GalaXi Space and less than 0.2% of the compounds of the Enamine REAL Space showcasing the ultralarge character of combinatorial chemical spaces. In (b), the results are shown for the pairwise and overall overlap of the Enamine REAL Space, the GalaXi Space, and the CHEMriya Space. The overlap of the Enamine REAL Space and CHEMriya Space with over 1.9 million compounds contains less than 0.01% of the Enamine REAL Space and less than 0.02% of the CHEMriya Space. The CHEMriya Space and GalaXi Space share only 126 143 products, and the overlap of all three chemical vendor spaces contains only 76 524 compounds.

In Figure 10, two products contained in the overlap of the Enamine REAL Space, GalaXi Space, and KnowledgeSpace are shown. As they are contained in the Enamine REAL Space and GalaXi Space, these compounds are commercially accessible through Enamine and WuXi. In addition, the reactions and building blocks incorporated in the KnowledgeSpace can be used to retrieve synthetic pathways known from the literature for both compounds using commercially available building blocks. Apart from the Suzuki coupling reaction scheme shown in (b), the same product can be synthesized using the Negishi, Stille, or Ullmann coupling reaction. The corresponding building blocks and reaction schemes are given in Figure S1 in the Supporting Information.

In Figure 11, an analysis of the 76 524 compounds contained in the 3-fold overlap of the Enamine REAL Space, GalaXi Space, and CHEMriya Space from Figure 9 is shown. The reaction schemes m_22bba for the Enamine REAL Space, WXVL001 for the GalaXi Space, and S_R001 for the CHEMriya Space all form an amide bond. With 55 942 compounds, over 70% of the overlap of the full Enamine REAL Space, GalaXi Space, and CHEMriya Space is part of the overlap of the chemical subspaces generated by these three reaction schemes. As all three chemical subspaces use the same chemical reaction, a shared carboxylic acid building block and amine building block lead to a product shared by all the chemical spaces. The chemical subspace of the Enamine REAL Space contains 18 873 carboxylic acid building blocks and 22 070 amine building blocks. The chemical subspace of the GalaXi Space contains 5107 carboxylic acid building blocks

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

and 1891 amine building blocks. The chemical subspace of the CHEMriya Space contains 2887 carboxylic acid building blocks and 4086 amine building blocks. All three chemical subspaces share only 387 carboxylic acid building blocks and 144 amine building blocks, resulting in 55 728 compounds in the overlap where they use the same building blocks and the same chemical reaction. An example compound is shown to the right in Figure 11. The remaining 214 compounds in their overlap contain more than one amide bond, and the three chemical subspaces use different building blocks to generate the same product by forming different amide bonds. The overlap analysis shows that the three chemical vendor spaces agree on the choice of building blocks and chemical reaction for most of the products contained in their overlap. However, 4672 compounds of this 3-fold overlap are not contained in any of the three chemical subspaces and are synthesized differently. The compound to the left in Figure 11 is generated by a condensation reaction at the quinoline in the CHEMriya Space. The same compound is synthesized using a coupling reaction at the amine in the Enamine REAL Space and GalaXi Space. For both synthesis routes, the exact type of reaction, reaction conditions, or intermediate products are not directly encoded in the chemical spaces by the vendors. Overall, the results show that these three very large chemical spaces containing commercially available compounds share only a very small fraction of compounds for which they mostly agree on the synthetic pathways. As the three chemical spaces share less than 2% of their compounds with any other space, they each cover their own area of the chemical universe. SpaceCompare is able to calculate the exact overlap of these vast chemical spaces for the first time. In addition, the method can retrieve the chemical information encoded in the topological fragment spaces. This way the synthesis routes of shared products can be compared between different chemical spaces. Even if classical enumeration-based methods were capable of processing chemical spaces of this size, they would not be able to detect differences in synthetic routes of products.

Computational Resource Analysis. We analyze the resource requirements of SpaceCompare for the three pairwise overlap calculations of the Enamine REAL Space, the KnowledgeSpace, and the GalaXi Space. The experiment is conducted on a 64-bit machine with openSUSE Leap 15, 250 GB RAM, and Intel Xeon Prozessor E5-4620 CPU architecture (2.2 GHz). For the SpaceCompare calculations, 12 parallel threads are used. In Table 1, the computing time and memory requirements are given for the three SpaceCompare calculations. The overlap calculation of the GalaXi Space and KnowledgeSpace takes less than 30 min and less than 20 GB of memory. In contrast, the overlap calculations including the Enamine REAL Space take 23 h and 58 GB of memory using the GalaXi Space and 63 h and 182 GB of memory using the KnowledgeSpace. Despite the KnowledgeSpace being 5 orders of magnitude larger in size compared to the GalaXi Space, the requirements increase only by roughly a factor of 3. This showcases the combinatorial approach behind SpaceCompare making the overall number of products less relevant for estimating the required resources.

Table 1. Computing Time Together with the Memory Requirements Are Shown for the Three Pairwise Comparisons of the Enamine REAL Space, GalaXi Space, and KnowledgeSpace[a]

pairwise comparison	covering step time (h)	combination step time (h)	total time (h)	memory (GB)
REAL vs Knowledge	18.735	44.084	62.819	181.8
REAL vs GalaXi	2.394	20.200	22.594	57.4
GalaXi vs Knowledge	0.426	0.020	0.446	17.3

[a]In addition to the total computing time of SpaceCompare, the computing time for each algorithm step is given as well.

In the following, we take a look at factors influencing the computational resource requirements of SpaceCompare. In the covering step, the algorithm detects fragments that cannot be used to generate products contained in the overlap. If many such fragments are detected and can be neglected afterward, the number of products that we have to consider for the overlap calculation of two spaces can be drastically reduced. As a result, the quantity of remaining fragments after the covering step plays a far greater role for the amount of resources required by SpaceCompare than the actual size of the involved chemical spaces. In Figure 12, the number of products that remain as candidates for the overlap of the two topological fragment spaces is shown for each of the three SpaceCompare calculations and each step of the algorithm. Only products generated by combinations of covered fragments remain as candidates for the overlap after the covering step. Notice how this number is smallest for the SpaceCompare calculation for the GalaXi Space and the KnowledgeSpace and largest for the calculations for the Enamine REAL Space and KnowledgeSpace reflecting the relation of computing time and memory requirements of the three calculations. The number of products remaining after the combination step is very similar for both topological fragment spaces across all three calculations. This indicates that the number of covered fragment combinations found in the initial part of the combination step already gives a good estimate of the final overlap of the two topological fragment spaces. For the Enamine REAL Space, the number of products remaining after the covering step is roughly an order of magnitude smaller than its total number of products. This difference amounts to at least 4 orders of magnitude for the KnowledgeSpace showcasing the strengths of the fragment-based approach in this scenario. The overall reduction of candidate products is smallest for the GalaXi Space in the overlap calculation with the Enamine REAL Space, but even in this case, the method is able to reduce the number of products that remain as candidates for the overlap by more than 1 order of magnitude.

However, this reduction of the number of candidate products for the overlap in the covering step is dependent on the two topological fragment spaces given as input and may not always be large enough. E.g., when comparing a topological fragment space to itself, no reduction can be achieved in the covering or combination step. If the overlap itself is not enumerable with available computational resources, the reduction will not be sufficient for SpaceCompare to operate. If the Enamine REAL Space shared 50% of its products with a hypothetical chemical space of similar size, SpaceCompare would not be able to calculate the overlap of the two spaces with reasonable computational resources since the overlap would be too large to be practically enumerable. However, SpaceCompare is able to calculate the overlap of the Enamine REAL Space and KnowledgeSpace, although the Knowledge-

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

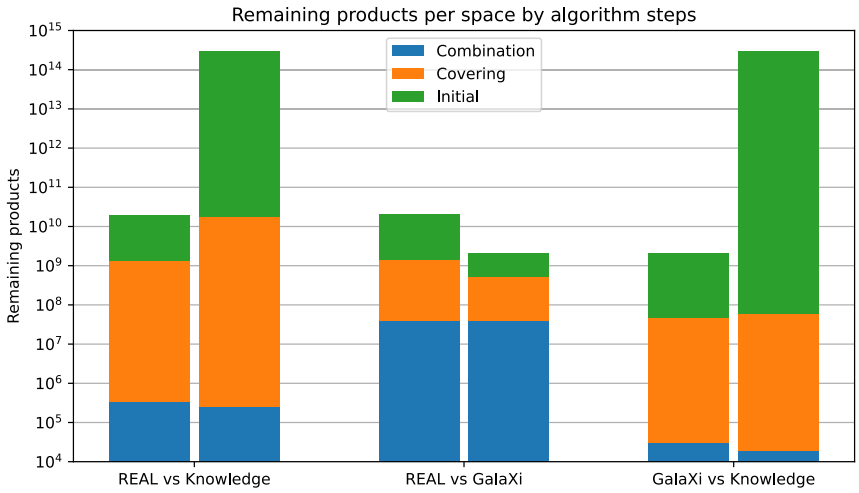


Figure 12. Remaining number of products that are potentially part of the overlap is shown for the three pairwise comparisons of the Enamine REAL Space, GalaXi Space, and KnowledgeSpace. The full bars indicate the total number of products in the topological fragment spaces. The orange and blue bars together indicate the number of products that can be generated from covered fragments after the finished covering step. The blue bar indicates the number of products remaining after the combination step with a stored unique SMILES. All numbers are shown in logarithmic units.

Space exceeds the Enamine REAL Space in size by 4 orders of magnitude. The results show that the critical factor for the resource requirements of SpaceCompare is the chemical similarity between the fragments of two chemical spaces instead of the number of contained products.

CONCLUSION

In this work, we described a novel tool called SpaceCompare for the calculation of the overlap of the products of two large combinatorial chemical spaces. Since SpaceCompare employs a combinatorial algorithmic approach, the overlap can be calculated in cases where the chemical spaces are too large for enumeration. Even if they are enumerable with available computational resources, SpaceCompare reduces the runtime requirements by several orders of magnitude.

With this new tool at hand, medicinal chemists can compare their own in-house ultralarge chemical libraries with other chemical spaces to identify compounds that are available through vendors, retrieve synthetic pathways from different chemical knowledge domains, or avoid patent space. SpaceCompare results can also be used to expand an in-house chemical library by incorporating chemical reactions and building blocks from other chemical spaces that do not contribute to the calculated overlap.

The method was validated, and the results of SpaceCompare were discussed using four well-known combinatorial chemical spaces, three of which are make-on-demand vendor catalogs. While being limited to the calculation of enumerable overlaps, SpaceCompare is able to operate on ultralarge chemical spaces exceeding trillions of products in size for the first time. For further analysis, it would be interesting to assess the performance of SpaceCompare on even larger chemical spaces like the GSK Space[12] containing 10^{26} compounds.

The pairwise and overall overlap of the Enamine REAL Space, WuXi's GalaXi Space, and Otava's CHEMriya was calculated. Each of these three ultralarge make-on-demand vendor catalogs shares less than 2% of its products with any of the other two. The results suggest that, where the three vendors apply the same chemical reactions, the overlap in building blocks is actually quite small, leading to an overall small overlap in products. This opens the way for further growth in commercially available chemical space by the addition of more building blocks and reaction chemistries. For the future, it would also be interesting to compare pricing by the three vendors for the products contained in the overlap of their chemical spaces.

At this point, it should be noted that mere *in silico* availability should not be the only guideline when comparing chemical spaces. Here, we used the two extremes of (a) the 10^{14}-sized KnowledgeSpace and (b) purchasable chemical spaces. The KnowledgeSpace has been created with a focus on size, public exposure of reactions, and known building blocks. It can be used to explore new areas of chemical spaces but neglects synthetic success rates, actual purchasability of products, or price-driven decision making. On the other side, there are smaller purchasable chemical spaces that are expected to provide physical samples for testing after synthesis at the suppliers' laboratories. An extremely low overlap between them can be a consequence of different synthesis strategies used by the suppliers which may also affect synthesis success rate, delivery time, and compound prices. To date, only data related to the Enamine REAL Space has been published.[5,20]

So far, only approaches relying on the sequential comparison of enumerated products of chemical libraries existed. From a computer scientist's point of view, what would be the computing time and memory requirements to calculate the overlap of the Enamine REAL Space and the KnowledgeSpace using a sequential approach? Identity comparison of each product individually in a classical, sequential approach would result in over 5×10^{24} such identity checks. To process this many identity comparisons in less than 63 h like SpaceCompare does, over 1 million comparisons would have to be conducted in every CPU cycle. Even with 1000 10 GHz CPU

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

cores that are used in parallel this is not possible with the hardware that exists today.

To reduce the number of required identity checks, unique identifiers could be generated and hashed for all products in both spaces. This way we would have to conduct only 2×10^{10} identity checks. Yet, we would require sufficient memory resources to store and access the 2.9×10^{14} products of the KnowledgeSpace. This by itself is currently beyond storage possibilities. Assuming one wanted to achieve this with 182 GB of memory like SpaceCompare does, over 199 products would have to be stored in a single bit, an unsolvable task.

Therefore, both approaches are, if not technically infeasible, definitely impractical with enumeration-based methods. Summarizing, we think that a combinatorially driven approach such as SpaceCompare is a fast and amenable way forward in a world of ever growing chemical libraries.

DATA AND SOFTWARE AVAILABILITY

The SpaceCompare tool will be available for Linux, MacOS, and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and is free for academic use and evaluation purposes. The tool SpaceLight is available at the same URL and can be used to generate custom topological fragment spaces. The KnowledgeSpace in its topological fragment space representation can be accessed at https://www.zbh.uni-hamburg.de/forschung/amd/datasets.html. For a detailed description of the chemical spaces used in this publication, see https://www.biosolveit.de/infiniSee#chemical_spaces.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01378.

Two chemical subspaces of KnowledgeSpace for reactions SnAP1 and SnAP2 as topological fragment spaces and as list of enumerated products, generated overlap using product lists and their unique SMILES and output of SpaceCompare for two topological fragment spaces, and two reaction schemes for product in overlap of Enamine REAL Space, GalaXi Space, and Knowledge Space (ZIP)

AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *Universität Hamburg, ZBH - Center for Bioinformatics, Research Group for Computational Molecular Design, 20146 Hamburg, Germany;* orcid.org/0000-0002-9553-6531; Email: rarey@zbh.uni-hamburg.de

Authors

Louis Bellmann − *Universität Hamburg, ZBH - Center for Bioinformatics, Research Group for Computational Molecular Design, 20146 Hamburg, Germany;* orcid.org/0000-0002-7920-1889

Patrick Penner − *Universität Hamburg, ZBH - Center for Bioinformatics, Research Group for Computational Molecular Design, 20146 Hamburg, Germany;* orcid.org/0000-0003-4988-6183

Marcus Gastreich − *BioSolveIT GmbH, 53757 Sankt Augustin, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c01378

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors would like to thank Vladimir Ivanov from Enamine Ltd., Xu Yan from WuXi Pharmatech Co. Ltd., and Yaroslav Bilokin from Otava Ltd. for access to their make-on-demand catalogs. Additionally, we would like to thank Yurii Moroz for highly valuable suggestions which improved the overlap analysis of the three vendor catalogs.

REFERENCES

(1) Petrova, T.; Chuprina, A.; Parkesh, R.; Pushechnikov, A. Structural Enrichment of HTS Compounds from Available Commercial Libraries. *MedChemComm* **2012**, *3*, 571−579.

(2) Engels, M. F.; Gibbs, A. C.; Jaeger, E. P.; Verbinnen, D.; Lobanov, V. S.; Agrafiotis, D. K. A Cluster-Based Strategy for Assessing the Overlap Between Large Chemical Libraries and its Application to a Recent Acquisition. *J. Chem. Inf. Model.* **2006**, *46*, 2651−2660.

(3) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.

(4) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20 - A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *J. Chem. Inf. Model.* **2020**, *60*, 6065−6073.

(5) Hoffmann, T.; Gastreich, M. The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discovery Today* **2019**, *24*, 1148.

(6) Brenner, S.; Lerner, R. A. Encoded Combinatorial Chemistry. *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89*, 5381−5383.

(7) Enamine REAL Space. https://enamine.net/library-synthesis/real-compounds/real-space-navigator (accessed 2021-08-05).

(8) GalaXi Space. https://www.labnetwork.com/frontend-app/p/#!/library/virtual (accessed 2021-08-05).

(9) CHEMriya Space. https://www.otavachemicals.com/products/chemriya (accessed 2021-08-23).

(10) Boehm, M.; Wu, T.-Y.; Claussen, H.; Lemmen, C. Similarity Searching and Scaffold Hopping in Synthetically Accessible Combinatorial Chemistry Spaces. *J. Med. Chem.* **2008**, *51*, 2468−2480.

(11) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with Feature Trees. *J. Chem. Inf. Model.* **2009**, *49*, 270−279.

(12) Warr, W. A. *NIH Meeting Ultra-Large Databases in Chemistry*. 2020. https://chemrxiv.org/articles/preprint/Report_on_an_NIH_Workshop_on_Ultralarge_Chemistry_Databases/14554803 (accessed 2022-01-15).

(13) Grebner, C. *Webinar: Exploration and Mining of Large Virtual Chemical Spaces*. 2018. https://youtu.be/fMrI11SXwpU (accessed 2022-01-15).

(14) Detering, C.; Claussen, H.; Gastreich, M.; Lemmen, C. KnowledgeSpace - A Publicly Available Virtual Chemistry Space. *J. Cheminformatics* **2010**, *2*, O9.

(15) Lessel, U.; Lemmen, C. Comparison of Large Chemical Spaces. *ACS Med. Chem. Lett.* **2019**, *10*, 1504−1510.

(16) Bellmann, L.; Penner, P.; Rarey, M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* **2019**, *59*, 4625−4635.

(17) Bellmann, L.; Penner, P.; Rarey, M. Topological Similarity Search in Large Combinatorial Fragment Spaces. *J. Chem. Inf. Model.* **2021**, *61*, 238.

(18) eMolecules. https://www.emolecules.com/ (accessed 2021-08-05).

(19) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(20) Klingler, F.-M.; Gastreich, M.; Grygorenko, O. O.; Savych, O.; Borysko, P.; Griniukova, A.; Gubina, K. E.; Lemmen, C.; Moroz, Y. S. SAR by Space: Enriching Hit Sets from the Chemical Space. *Molecules* **2019**, *24*, 3096.

https://doi.org/10.1021/acs.jcim.1c01378
J. Chem. Inf. Model. 2022, 62, 553−566

D.4. Calculating and Optimizing Physicochemical Property Distributions of Large Combinatorial Fragment Spaces

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

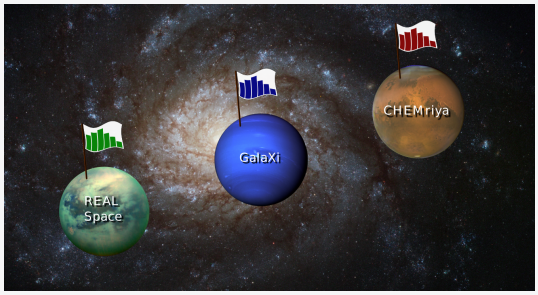
Calculating and Optimizing Physicochemical Property Distributions of Large Combinatorial Fragment Spaces

Louis Bellmann, Raphael Klein, and Matthias Rarey*

Cite This: https://doi.org/10.1021/acs.jcim.2c00334 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The distributions of physicochemical property values, like the octanol−water partition coefficient, are routinely calculated to describe and compare virtual chemical libraries. Traditionally, these distributions are derived by processing each member of a library individually and summarizing all values in a distribution. This process becomes impractical when operating on chemical spaces which surpass billions of compounds in size. In this work, we present a novel algorithmic method called SpaceProp for the property distribution calculation of large nonenumerable combinatorial fragment spaces. The novel method follows a combinatorial approach and is able to calculate physicochemical property distributions of prominent spaces like Enamine's REAL Space, WuXi's GalaXi Space, and OTAVA's CHEMriya Space for the first time. Furthermore, we present a first approach of optimizing property distributions directly in combinatorial fragment spaces.



INTRODUCTION

In the first phases of drug discovery, compounds of interest are retrieved from physical or virtual chemical libraries. Often times, the quality of a compound is related to certain physicochemical text properties like water solubility,[1,2] and many methods for the efficient calculation exist.[3−6] In this context, the applicability of a given chemical library can be assessed by calculating property distributions of the contained compounds. For chemical libraries with up to several million compounds, these distributions are generated routinely by calculating the property value of each compound one-by-one and then concluding them in a single distribution.

Traditionally, virtual chemical libraries represent a set of compounds as an enumerated list. As a result, the memory needed to store and the computational time needed to access the chemical library scales roughly linear with the number of compounds contained in the library. This leads to a large amount of computational resources required for chemical libraries of single digit billion compounds.[7] To represent chemical libraries containing double digit billion or more compounds, without investing an impractical amount of computational resources, a different representation is needed.[8]

Combinatorial chemical libraries, like large make-on-demand catalogues or DNA-encoded libraries,[9] use reactions together with compatible building blocks to generate products in a combinatorial approach. Following the same rational virtually, combinatorial fragment spaces do not enumerate each compound individually. Instead, reactions and building blocks are encoded to implicitly describe a chemical space of products. Due to combinatorial explosion, the number of products represented this way can surpass the number of building blocks and reactions by several orders of magnitude. A chemical space exceeding trillions of compounds in size can be generated by using a limited number of encoded building blocks and reactions. With methods for virtual screening[10−12] and comparison[13] being developed, the interest in these ultralarge combinatorial chemical spaces increased. Several proprietary,[8,14−17] a publicly available,[18] and three compound vendor[19−21] spaces exist. Currently, the largest number of 10^{26} contained compounds is reported in the GSK space,[16] showcasing the vast difference in size between combinatorial and enumerated chemical libraries.

For these ultralarge chemical spaces, no distributions of physicochemical properties of the represented products exist. The strategy of calculating each product's property value one-by-one is not feasible as the products of ultralarge chemical spaces cannot be practically enumerated. In this work, we present a novel algorithm called SpaceProp. Like other algorithmic methods[10−13] for ultralarge chemical spaces, SpaceProp utilizes a combinatorial approach to avoid product enumeration. In contrast to these methods, the algorithm described in this work is not concerned with virtual screening or

Received: March 22, 2022

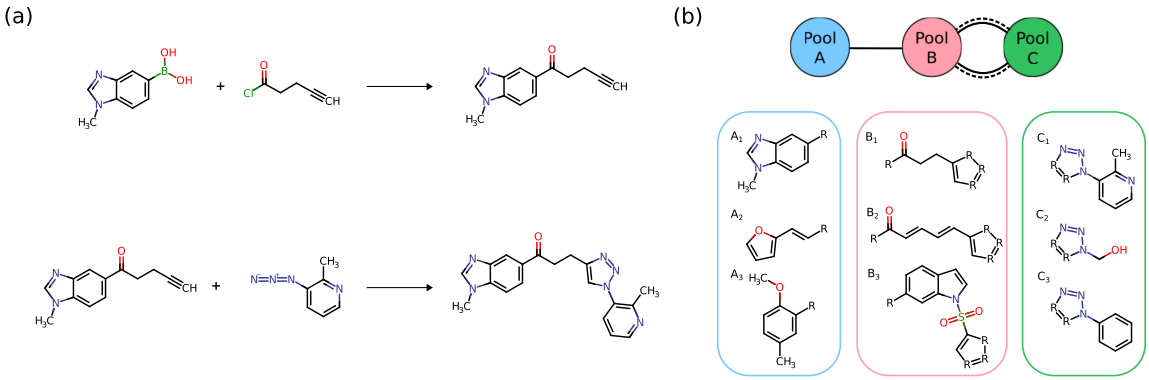


Figure 1. In (a), we show a reaction involving a Suzuki coupling reaction, followed by an azide−alkyne cycloaddition. In (b), the topology graph representing the two combined reactions is depicted. Fragments belonging to a pool are surrounded by a box with the corresponding color. Pool A contains fragments representing boronic acids. The fragments of Pool B represent organohalides. Pool C contains fragments representing hydrazoic acids. The edge between pool A and B represents the chemical single bond formed between the reactants during the Suzuki coupling reaction. The edges between pools B and C represent chemical aromatic bonds of the 1,2,3-triazole formed in the azide−alkyne cycloaddition.

overlap calculation. Instead, SpaceProp determines the exact distributions of product property values of combinatorial chemical spaces for five well-known physicochemical properties for the first time. The algorithm scales with the number of building blocks and the property value range, rather than with the number of products. SpaceProp is therefore applicable to ultralarge combinatorial spaces. We will employ the algorithm to assess the product property distributions of three commercial vendor spaces and the publicly available KnowledgeSpace. In addition, we describe a workflow to generate custom optimized chemical subspaces containing an increased percentage of products with desired properties defined by the user.

METHODS

SpaceProp is able to generate physicochemical property distributions of large-size combinatorial chemical spaces, without enumerating all its products. Instead, the physicochemical properties of the building blocks are calculated, summarized, and combined to avoid combinatorial explosion. This enables the exact calculation of property distributions across chemical spaces containing trillions of compounds. Here, we describe the workflow for atom-based properties, but the procedure can in principle be applied to bond- or fragment-based properties as well.

Topological Fragment Spaces. Topological fragment spaces represent combinatorial chemical spaces without specifically enumerating the contained products and found their application in previously developed combinatorial algorithmic approaches.[12,13] Topological fragment spaces extract information from reactants and chemical reactions and implicitly describe the combinatorial space of derived products. This results in a compact form, enabling efficient algorithmic approaches even in very large-sized chemical spaces. Products that originate from the same chemical reaction using different reactants share the same topology induced by the chemical reaction. This behavior is captured in the topology graph. The nodes of the topology graph represent a pool of reactants that can be used interchangeably at the same position in the chemical reaction. The edges of the topology graph represent the chemical bonds that are formed during the reaction between the reactants. For the reactants themselves, we use a fragment representation. These fragments contain link atoms, and the atom and bond configuration around them is adjusted to resemble the chemical environment of the reactant within the product. This way, the fragment property values can be used to derive the property values of generated products. One or more topology graphs together with the involved fragments make up a topological fragment space. A product can be generated by selecting one fragment from each pool of a topology graph. Link atoms are removed from the fragments. The edges of the topology graph are used to form chemical bonds between the heavy atoms of different fragments. In this process, the chemical bond configuration of all heavy atoms stays unchanged as the fragment representation captures the chemical environment of a reactant once it is part of a product.

In Figure 1, we show a topological fragment space capturing a Suzuki coupling reaction,[22] using an arylboronic acid group and an acyl chloride group.[23,24] This is followed by an azide−alkyne cycloaddition.[25] The reactants in Figure 1(a) are represented by fragments A_1, B_1, and C_1 in Figure 1(b). For all three pools, we added two additional reactants represented by fragments. We exchange the boronic acid group with a linker for all fragments in pool A. Similarly, we exchange the halogen atom with a linker for all fragments in pool B. Both groups are not contained in the resulting products and are therefore removed. Additionally, the alkyne group in the reactants for pool B are adjusted and combined with link atoms to form their configuration in the final products. The same procedure is applied to the hydrazoic acids in the reactants of pool C. The newly formed single bond and aromatic bonds of the product are captured by the edges of the topology graph.

This fragment representation will help us to calculate the property value of products while operating on the fragment level. The sum of heavy atoms contained in fragments A_1, B_1, and C_1 is identical to the number of heavy atoms contained in their product. The same does not hold for the represented reactants in Figure 1(a), as the leaving boronic acid group and the halogen atom are still attached.

Physicochemical Properties. In this work, we consider the four physicochemical properties of compounds that are part of Lipinski's "rule of five",[2] which are the number of hydrogen bond acceptors, the number of hydrogen bond donors, the molecular weight, and the octanol−water partition coefficient

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(log P). In this context, we use the definition for hydrogen bond acceptors and donors introduced by Lipinski et al.[2] For log P calculation, we apply the widely used atom-based strategy of Wildman and Crippen[4] which is an updated version of earlier methods[26−28] and performed well in a benchmark experiment comparing 30 log P calculation methods.[29] We denote this calculation strategy with aLogP throughout this work. We want to mention that all log P calculation methods assessed in the benchmark,[29] including aLogP, showed the best performance for small, drug-like molecules and decreasing accuracy for larger, more lipophilic compounds. For aLogP calculation, each atom is assigned one of 68 atom types based on their element and chemical environment. Each atom type corresponds to a fixed contribution to the overall aLogP of the compound which is derived as the sum of these contributions. In addition to these properties, we include the number of heavy atoms contained in a compound, which is not part of Lipinski's "rule of five" but is correlated to the molecular weight of a compound.[30]

SpaceProp calculates these five properties for all products of a topological fragment space while only working with its fragments and combinations of property values. In the topological fragment space, we represent the employed building blocks by fragments mimicking the configuration of the building block after the reaction within a product. As a result, the combined property values of fragments give a good estimate of the property value of a product. In this context, we call a physicochemical property *fragment-additive*, if the property value of a product always equals the sum of property values of the fragments generating the product. Note that the number of hydrogen bond acceptors and heavy atoms are fragment-additive properties. The atom types used in the aLogP depend on the chemical environment of the atom. As a result, we cannot calculate these atom types for each fragment individually. The number of hydrogen bond donors and the molecular weight are not fragment-additive properties due to delocalized representations of reaction schemes with heteroaromatic ring closures. This special case is further described in the Supporting Information.

Fragment-Additive Physicochemical Property Distribution Algorithm. For a fragment-additive property, its distribution across a topological fragment space can be calculated without enumerating its products with the following approach in two steps.

1. Calculate the property value for all fragments of the topological fragment space. For each pool of some topology graph, generate a distribution by binning contained fragments with the same value.
2. For each topology graph, generate a distribution by adding all combinations of values from the distributions of its pools and multiplying the corresponding counts. Add the multiplied counts, if two of these value combinations result in the same sum of values. Summarize these distributions across all topology graphs of the topological fragment space.

Nonfragment-Additive Physicochemical Property Distribution Algorithm. All properties we consider in this work are atom-based, meaning that the property value of a compound can be derived as the sum of values of its atoms. In this subsection, we establish an algorithm applicable to atom-based nonfragment-additive properties. This procedure will be applied to the aLogP, molecular weight, and number of hydrogen bond donor properties, but it could in principle be used for other properties as well. If a property is not based on atom values, but, e.g., on chemical substructures or rotatable bonds, the general idea of the algorithm can be applied with adjusted definitions. For a nonfragment-additive property, the value of a product does not necessarily equal the sum of values of fragments generating that product. This is caused by atoms of the product, for which the values are influenced by multiple fragments. The goal of the algorithm we introduce is to retrieve all relevant information in order to derive the correct property value of these atoms, without actually enumerating all products of a topological fragment space.

In this context, we call the atom of a fragment an *inner atom* if its property value does not change once the fragment is part of a product. Note that, for a fragment-additive property, all atoms are inner atoms. The *internal property component* of a fragment is defined as the property value sum of its inner atoms. The *external property component* of a product is defined as the property value sum of atoms that are noninner atoms in their respective fragments. The property value of a product can thus be derived by adding the external property component to the sum of internal property components of the fragments generating that product. We aim to detect those products sharing the same external property component and avoid enumerating all of them in the process. The *boundary information* of a fragment contains all its noninner atoms as well as all relevant information to derive the external property components of all products that can be generated using the fragment. In the case of the aLogP calculation, this will include information about the chemical environment of link atoms. In the case of the molecular weight and hydrogen bond donor property, this will include information about heteroaromatic ring closures.

If we exchange one fragment with another fragment from the same pool sharing its boundary information, the external property component of the corresponding product stays unchanged. We only need to exchange one internal property component to derive the correct value of the product. If additionally the internal property components of the exchanged fragments stay unchanged, the resulting product values are also identical. Using this fact, we can define a procedure calculating the nonfragment-additive property distribution of a topological fragment space without enumerating all of its products.

1. Calculate the internal property component and boundary information for all fragments of the topological fragment space.
2. For each pool of some topology graph, group all fragments sharing the same boundary information and generate distributions of their internal property components.
3. For each topology graph, enumerate all combinations of distributions from different pools. For each of these combinations, generate a summarized distribution by performing the following two substeps.
   - 3.1. Resolve the boundary information to calculate the external property component.
   - 3.2. Sum all combinations of internal property components, add the external property component, and multiply their counts. Add the counts if two combinations result in the same sum.
4. Summarize all these distributions across all combinations and topology graphs of the topological fragment space to generate its overall property distribution.

By grouping fragments with the same boundary information in a distribution, we avoid calculating the external property

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

component for each product individually. Instead of considering all fragments individually in the second step of the algorithm, we can combine distributions representing multiple if not all fragments of a pool. In the following, we will apply this strategy to the aLogP. The application of this procedure to the number of hydrogen bond donors and the molecular weight is described in the Supporting Information.

aLogP Distribution Calculation. The atom types used by Wildman and Crippen are derived with information about the element of the atom, its attached chemical bonds, and its direct atom neighbors. The only exceptions are oxygen atoms that are contained in a carbonyl group or a hydroxy group and hydrogens that are contained in a hydroxy group. For these *exceptional atoms*, the atoms that are attached to its neighbors are also relevant to determine their atom type. The atom type definition of Wildman and Crippen of an inner atom must not depend on the chemical environment of the fragment within a product. Thus, a fragment atom is an inner atom, if it has no link atom attached and additionally is not exceptional or none of its neighbors have a link atom attached. We call an atom a *boundary atom* if it has a link atom attached. Depending on its chemical environment, boundary atoms and their exceptional atom neighbors might also have an atom type that is not influenced by atoms from other fragments. In this case, these atoms are also inner atoms.

We now summarize all relevant properties for the fragment boundary information. For this, we consider all boundary atoms. We call a boundary atom *resolved*, if it is an inner atom and all its exceptional atom neighbors are as well. For resolved boundary atoms, we only need to store information that is needed to resolve boundary atoms from other fragments. To achieve this, we store the atomic element of a boundary atom as well as its aromaticity and attached link atoms.

For all unresolved boundary atoms, we need to store more information to resolve its atom type and the atom type of unresolved exceptional atom neighbors. We store the same information as for resolved boundary atoms but add information about its valence state and additionally count occurrences of certain atom neighbors. In the publication of Wildman and Crippen, an atom is considered as a heteroatom if it is a nitrogen, oxygen, phosphorus, sulfur, fluorine, chlorine, or iodine atom. We count the number of attached hydrogens and the number of attached carbons. Additionally, we include the number of attached heavy atoms that are not carbon or heteroatoms. Following the definition of Wildman and Crippen, we call these atoms *unusual*. Moreover, we count the number of attached aromatic atoms and include attached aromatic link atoms as they resemble aromatic atoms once the fragment is part of a product. Information on unresolved exceptional atom neighbors is included. All of this information combined for all boundary atoms forms the boundary information on the fragment. An fCSFP[12,31] identifier is added to the boundary information to account for heteroaromatic ring closures containing nitrogen atoms. The strategy is further described in the Supporting Information. In the following examples, this identifier is neglected for simplicity.

In Figure 2, the boundary information and atom types of all inner atoms are shown for the fragment B_1 from Figure 1(b). The atom type definition of Wildman and Crippen is independent of other attached fragments for all atoms, except the oxygen atom of the carbonyl group. The aLogP values corresponding to the resolved atom types can be added to form the internal aLogP value of the fragment B_1. Note that we

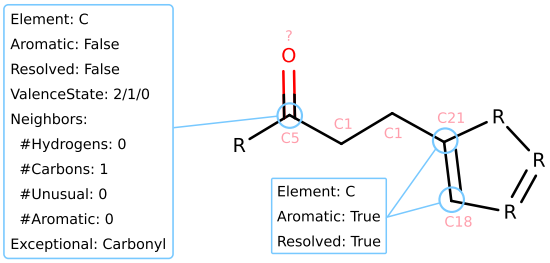


Figure 2. Fragment B_1 from Figure 1(b) is shown together with its boundary information about the boundary atoms encircled in blue. Additionally, the Wildman and Crippen atom type is depicted for each inner heavy atom. For clarity, hydrogens are omitted. The oxygen atom is exceptional and not an inner atom, which is indicated by the question mark for its unresolved atom type. The entry 'ValenceState' indicates the two single and one double chemical bond of the carbon atom.

omitted the hydrogen atoms in Figure 2. Their aLogP contribution must be added to the inner aLogP value. For all boundary carbon atoms, we store all relevant information to determine the unresolved atom types of other attached fragments. In addition, the relevant information to later resolve the atom type of the exceptional oxygen atom of the carbonyl group is stored as well.

We can now derive the aLogP distribution over all products of a topological fragment space, by applying the procedure described in the previous section. For the first step of the algorithm, we have to calculate the boundary information and inner aLogP value for all fragments. In the next step, we group fragments sharing the same boundary information.

In Figure 3, the result of the first step of the algorithm is shown for the topological fragment space from Figure 1(b). Instead of nine fragments and 27 products, we now have five concluded groups of fragments with the same boundary information and four combinations thereof. As a result, only four external aLogP values have to be calculated. As no two fragments of a group share the same internal aLogP value, we have to calculate 27 sums of internal aLogP values.

In Figure 4, the two remaining unresolved atom types are resolved to generate the external aLogP value for one combination of fragment groups. The unresolved atom of fragments A_1 and A_3 has an additional carbon atom neighbor contained in fragments B_1 and B_2 in the corresponding product. Thus, we can resolve its atom type to C21 following the definition of Wildman and Crippen. Similarly, the carbonyl group of fragments B_1 and B_2 has an additional aromatic neighbor in fragments A_1 and A_3, and we assign the atom type O10 to the oxygen atom of the carbonyl group in both fragments. We can now derive the aLogP values of all products that are generated using fragments from this group combination. The product displayed in Figure 4 generated by fragments A_1, B_1, and C_1 has an aLogP value of 3.2167. Similarly the aLogP value of other fragment combinations can be calculated the same way by just exchanging the corresponding internal aLogP values in the sum. All product aLogP values are rounded to one decimal place, and all product molecular weights are rounded to integers.

Approximate Approach. In this section, we present an approximate approach to speed up the distribution calculation. Internal property components of fragments have to be summed in order to derive the property values of products. If no two fragments share the same internal property component, no

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

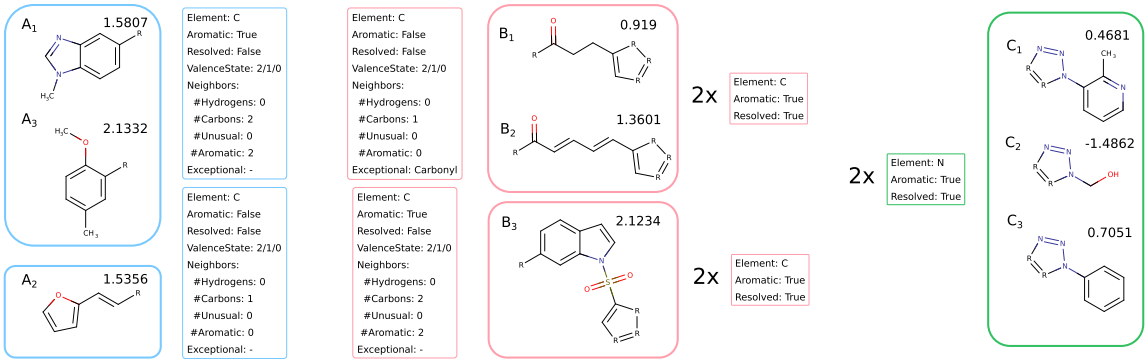


Figure 3. Boundary information and internal aLogP value for each fragment of the topological fragment space from Figure 1(b). Indicated by the boxes, the fragments of a pool are grouped by shared boundary information. The corresponding boundary information is next to the group and shares the same color. The decimal numbers give the internal aLogP value of the fragments.

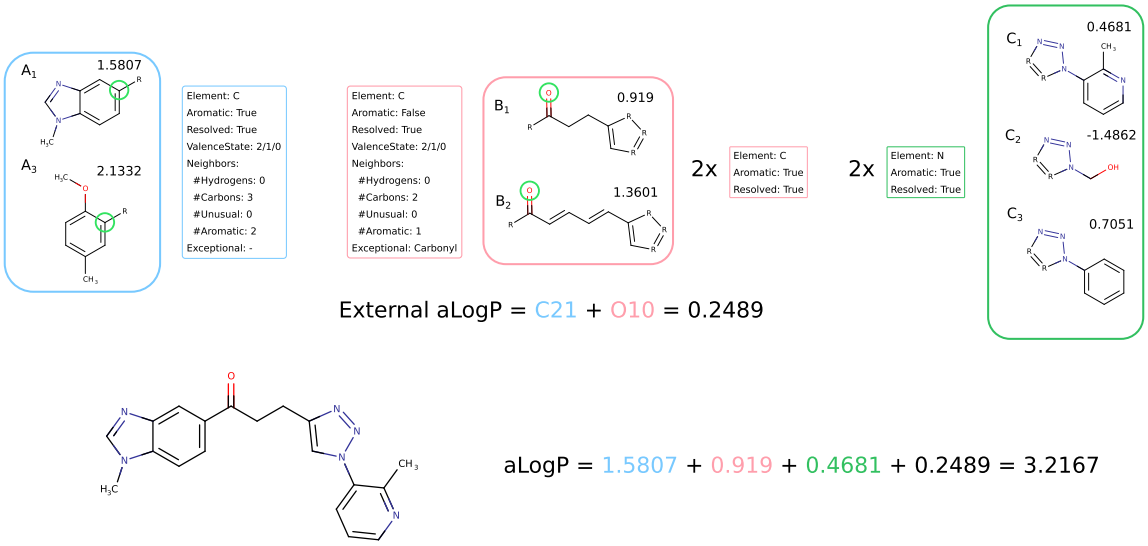


Figure 4. External aLogP component calculation for one combination of fragment groups from Figure 3. The resolved boundary information is shown instead of the unresolved one from Figure 3. The atoms contributing to the external aLogP value are encircled in green in the corresponding fragments. The derived atom types share their color with the group that contributes the corresponding atoms. The product generated by fragments A_1, B_1, and C_1 is given together with the sum for its aLogP value. The internal aLogP values are colored with the group of the corresponding fragment. The external aLogP value is colored in black.

fragments can be binned in the first step of the algorithm. As a result, the number of sums that have to be calculated equals the number of products contained in the topological fragment space. In this case, the algorithm cannot avoid the combinatorial explosion and suffers from long computational run times. For properties like the number of electron acceptors, electron donors, or heavy atoms, the range of fragment values is usually far smaller than the number of fragments. As a result, many fragments with the same internal property component are binned together, and the product property values can be calculated efficiently. For properties like molecular weight and aLogP, the rate of fragments with identical property values can be much lower because of the larger value range due to decimal number representation. We represent the molecular weight of a fragment as a number with four decimal places and the aLogP value of a fragment as a number with five decimal places. In the approximate approach, we reduce the range of possible internal property components by rounding the inner molecular weight component of a fragment to two decimal places and its inner aLogP component to three decimal places. Although this speeds up the calculation, rounding errors in the product property values can occur. Since molecular weight product values are rounded to integers, the rounded molecular weight of a product can differ by one between the exact and the approximate approach. The aLogP product values are rounded to one decimal place, so the rounded aLogP of a product can differ by 0.1 between the exact and the approximate approach. In the Results section, we will analyze how large the increase in efficiency and the error introduced by rounding is for the approximate approach in comparison to the exact approach.

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

RESULTS

In this section, we apply the SpaceProp algorithm to generate physicochemical property distributions of well-known large combinatorial chemical spaces, but first, we validate the method by comparing the distributions generated by SpaceProp to distributions generated by enumerating all products and calculating their physicochemical properties individually. The combinatorial chemical spaces considered in this publication are far too large to be economically enumerable. Therefore, we generate smaller chemical subspaces and use them for the validation instead.

Chemical Spaces. In this work, we consider four chemical spaces, three of which are chemical compound make-on-demand spaces. Enamine's REAL Space[19] spans a product space of more than 20 billion commercially available compounds by using 156 synthesis protocols and over 104 000 building blocks. OTAVA's CHEMriya[21] make-on-demand chemical space applies 44 in-house reactions to 30 000 building blocks, resulting in 11 billion products. WuXi designed the GalaXi[20] virtual chemical space by using 30 different reaction types and more than 155 000 building blocks. The space is comprised of 2.3 billion compounds. In addition, the publicly available KnowledgeSpace[18] contains over 100 trillion compounds. The space is generated using 117 reaction schemes known from literature that are applied to 142 050 selected building blocks from the eMolecules[32] collection.

Validation. We validate the distribution calculation of SpaceProp by showing its agreement with a classic one-by-one compound physicochemical property calculation. The four chemical spaces described in the previous paragraph are too large for enumeration, and therefore, classic calculation methods cannot be applied. For this reason, we generate two smaller enumerable chemical subspaces for each of the four input spaces and use them for validation.

A single reaction, together with its compatible building blocks, forms a chemical subspace of the full chemical space. For each space, the reactions are identified, for which the chemical subspace contains at most 100 000 products. All of these reactions are used collectively to form the first type of chemical subspace for each of the four chemical spaces. For further validation, we generate a second type of chemical subspace. For these subspaces, we select all reactions and pick a subset of building blocks to which the reactions are applied. If a single reaction together with the compatible building blocks generates at most 10 000 products, the reaction together with all its compatible building blocks will be part of the chemical subspace. If the reaction generates more than 10 000 products, compatible building blocks are removed randomly until the number of generated products is at most 10 000. Note that due to combinatorial reasons, the exact number of 10 000 products cannot always be achieved. In REAL Space, the same reaction might be applied to different collections of building block sets. In this case, the described procedure is applied to each collection independently, and the resulting chemical subspace can contain up to 10 000 products for each collection. Because of this reason, the second subspace of REAL Space contains more compounds than the 156 employed reactions multiplied by 10 000 products. In Table 1, the number of products contained in the eight chemical subspaces is listed.

The products are enumerated for all eight chemical subspaces. For each product, the number of hydrogen bond acceptors and the number of hydrogen bond donors in the definition of Lipinski[2] are determined. In addition, the molecular weight, the aLogP, and the number of heavy atoms are calculated. The molecular weight is rounded to the nearest integer, and the aLogP value is rounded to one decimal place. All product property values are binned in five distributions for each of the five properties. The five distributions generated by SpaceProp are identical for all eight chemical subspaces. These results show that SpaceProp generates exact physicochemical property distributions across all products of a combinatorial fragment space. The two chemical subspaces of KnowledgeSpace are given in the Supporting Information as topological fragment spaces as well as enumerated product SMILES lists together with the generated property distributions for reproducibility.

Table 1. Number of Products Contained in the Two Types of Chemical Subspaces for REAL Space, GalaXi, CHEMriya, and KnowledgeSpace

fragment space	first subspace size	second subspace size
REAL Space	1 036 275	2 415 745
GalaXi	55 672	345 125
CHEMriya	279 700	479 863
KnowledgeSpace	787 042	1 086 114

Distributions. We apply SpaceProp to generate the physicochemical property distributions of the full REAL Space, GalaXi, CHEMriya, and KnowledgeSpace. The generated distributions account for the number of possible fragment combinations with the given property value. Note that if two different fragment combinations generate the same product, the property value of this product is counted twice.

All property distributions are displayed in Figure 5. The distributions span over a large range for all five properties, and all of the four spaces that were taken into account. Among the "rule of five" criteria (logP, molecular weight, number of hydrogen bond acceptors and donors), the biggest difference in distribution was observed for the molecular weight spanning between fragment-like compounds with 200 g/mol and very large ones with up to 1000 g/mol. The highest percentage of compounds with molecular weight below 500 g/mol was found in REAL Space (85%), and the lowest amount was found in KnowledgeSpace (<0.01%). However, due to the total size of the latter, the total number of compounds fulfilling this "rule of five" constraint is still in the range of 14 billion. The percentage and number of compounds fulfilling each individual "rule of five" criterion is given for all four chemical spaces in Table S1 in the Supporting Information.

Summarizing all remainders in Table S1 except the heavy atom count, we find that at least 74% of the Enamine REAL Space products fulfill all constraints of Lipinski's "rule of five". If we summarize all remainders except that for the constraint on the molecular weight and heavy atom count, we see that at least 89% of the REAL Space products meet three out of four criteria and thus fulfill Lipinski's "rule of five" itself. Note that this is only a lower bound, and the actual percentage of compounds fulfilling the rule might be even higher. As products failing one constraint can also fail another one, subtracting individual remainders can underestimate the actual percentage of "rule of five" compounds. Higher numbers of halogens significantly affect the molecular weight. Therefore, we also calculated the distribution of heavy atoms. Although Lipinski did not take this property into account, follow-up research calculated a number of 36 heavy atoms to correspond to a molecular weight of 500 g/mol.[30] The

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

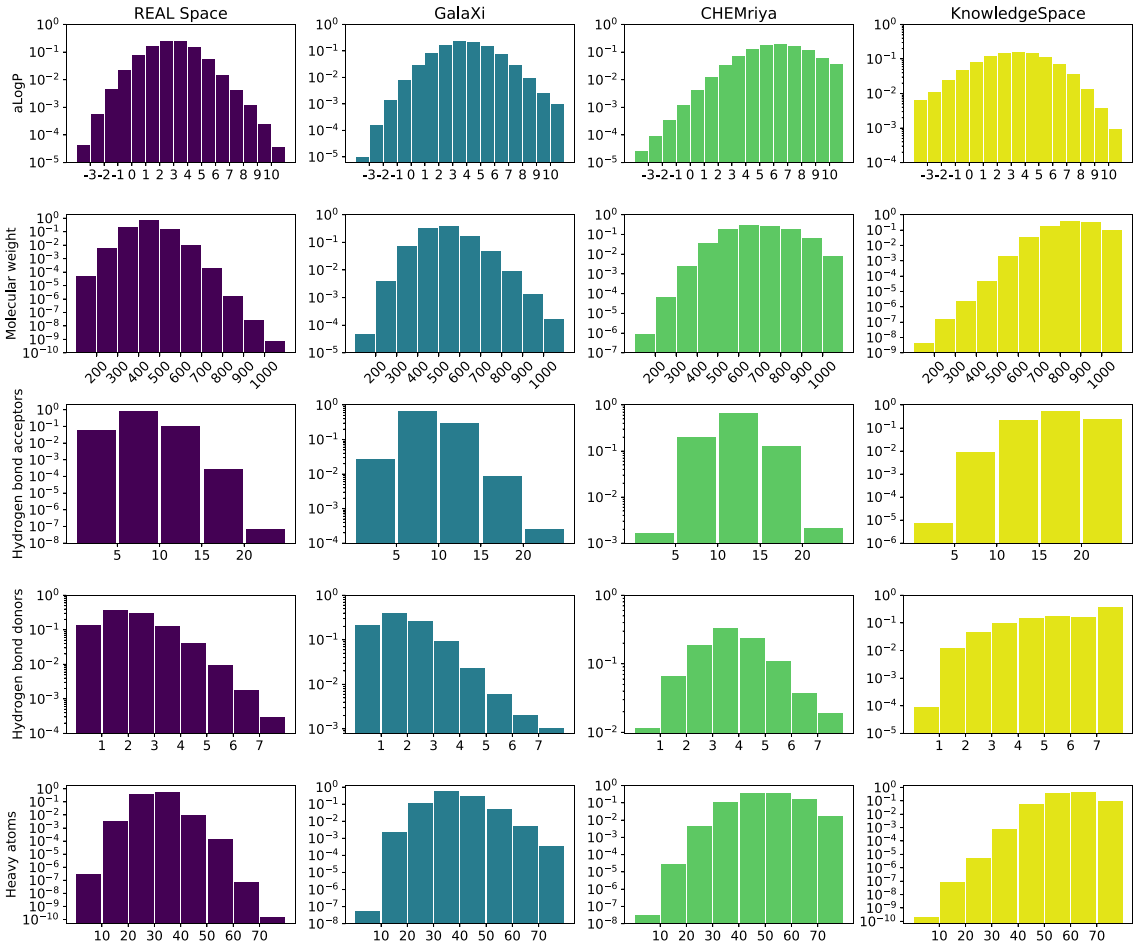


Figure 5. Physicochemical property distributions of REAL Space, GalaXi, CHEMriya, and KnowledgeSpace. The bins conclude the products of one chemical space with a property value smaller than the value on the *x*-axis to the right of the bin (if present) and bigger or equal to the value on the *x*-axis left of the bin (if present). The *y*-axis is displayed in log-scale due to the large differences in bin size. The values on the *y*-axis give the relative size of the bins compared to the total number of products in the chemical space.

percentage and number fulfilling this constraint are given for all four spaces in Table S1 in the Supporting Information.

In contrast to REAL Space, CHEMriya and KnowledgeSpace exhibit a high percentage of product molecules for beyond "rule of five" applications. Those can be, for instance, PROTACS,[35] peptidomimetics,[36] or lipidomimetics.[34] For each of the four spaces, a molecule that fulfills the "rule of five" criteria and another one for a specific beyond "rule of five" application were extracted and are displayed in Figure 6. The examples were retrieved from the chemical spaces by means of a topological similarity search with the tool SpaceLight.[12] As examples for REAL Space, the antiplatelet drug Clopidogrel and an analog of a thalidomide-based PROTAC were found. In the group of lipidomimetics, an analog of an HIV-1 entry inhibitor IBS70[34] was found in CHEMriya. Besides this larger compound, a close analog of Rivaroxaban was retrieved from CHEMriya as well. GalaXi and KnowledgeSpace were used to find peptidomimetics with Brilacidin,[33] an investigational antibiotic, and Zankiren, a blood pressure reducing drug, as queries. As representatives for "rule of five" complying compounds, Fentanyl was found in GalaXi, and in KnowledgeSpace, a close analog of Zonisamide which is used in the treatment of epilepsy and Parkinson's disease was found.

Optimization. With the insight gained from the physicochemical property distributions, we aim at identifying ways to generate chemical spaces that are optimized for a given application. As a first step in this direction, we included the possibility to remove a percentage of fragments with the highest internal property component from a topological fragment space for the five physicochemical properties considered.

In Figure 7, the optimization results are shown for KnowledgeSpace. Before optimization, the largest bin indicates products with an aLogP of five or higher. With an increasing percentage of fragments removed, the higher aLogP value bins decrease in size suggesting that the average product is more hydrophilic. When removing fragments, the number of overall products contained in the topological fragment space decreases. The optimized chemical spaces with 70% of fragments

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

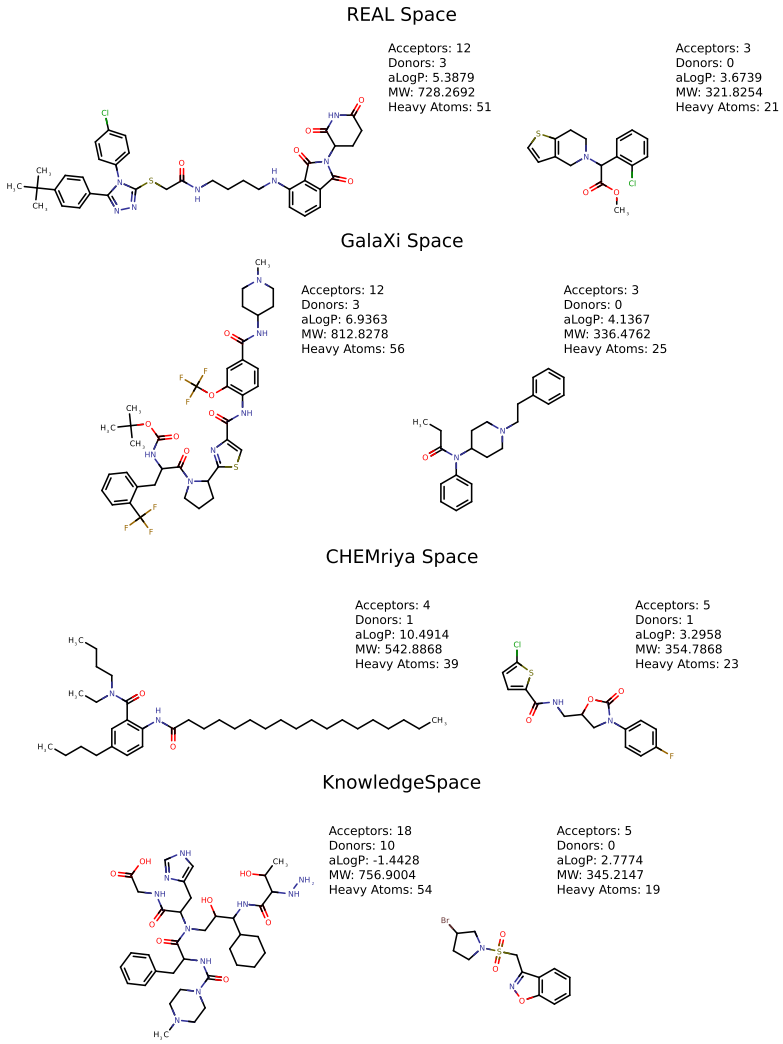


Figure 6. For each of the four topological fragment spaces, two contained products are depicted. The five physicochemical properties are given for each product. Compounds for REAL Space are Clopidogrel and an analog of a thalidomide-based PROTAC. For GalaXi, Fentanyl and an analog of Brilacidin[33] are shown. For CHEMriya, a close analog of Rivaroxaban and an HIV-1 entry inhibitor IBS70 are shown.[34] For KnowledgeSpace, a close analog of Zonisamide and Zankiren are depicted.

remaining contain 32% of the products of the whole KnowledgeSpace which is still more than 90 trillion products. Due to the ultralarge character of these chemical spaces, their size is measured in orders of magnitude. As a result, even removing half of their products might be a viable strategy for optimization.

Run Times and Approximate Approach Analysis. In this section, we discuss run times of the exact and approximate histogram calculation. Additionally, we analyze the error in the approximate distribution introduced by internal property component rounding. The calculations are carried out on the full, nonenumerable chemical spaces REAL Space, GalaXi, CHEMriya, and KnowledgeSpace.

In Table 2, the run times of the exact and approximate approach are given. For this experiment, we used Ubuntu 20.04.2 LTS on a 64-bit machine with 16 GB memory and Intel Core i5-4690K CPU architecture (3.5 GHz) using four threads. The exact distribution calculation of KnowledgeSpace takes, with roughly 2 h, far longer than the exact calculation for the other three chemical spaces. The exact distribution calculation of GalaXi takes less time compared to the other spaces. This reflects the ordering in size of the chemical spaces. However, KnowledgeSpace exceeds GalaXi by 4 orders of magnitude in size, and its run time results only by roughly 2 orders of magnitude. This is likely caused by the rather large value range of the aLogP with five decimal places in the exact approach. As a result, the time requirements depend to some extent on the number of products within a chemical space. For the approximate approach, this difference in run times is reduced, and the computational time needed for processing REAL Space is actually higher than that for KnowledgeSpace. Due to rounding, more fragments have an identical internal property component and are thus binned together. As a result, less sums

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

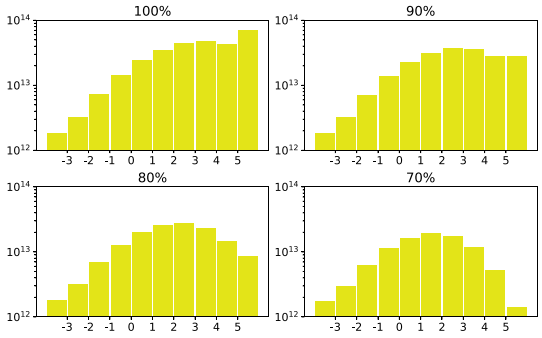


Figure 7. Optimization of the aLogP distribution of KnowledgeSpace. In the upper left corner, the original aLogP distribution of KnowledgeSpace is shown. The bins are assigned the same way as in Figure 5. From left to right and top to bottom, 10%, 20%, and 30% of fragments with the highest internal aLogP component were removed, and the resulting aLogP distribution is depicted.

of internal property components have to be generated, and the number of products of KnowledgeSpace plays a smaller role for the calculation. SpaceProp calculates the approximate distribution for each of the four chemical spaces in roughly 2 min. Finally, the error percentage is calculated by adding the absolute differences between counts for all values in the exact and approximate distributions. The determined sum is divided by the total number of products. The average error for the two properties is derived by multiplying the percentage by 0.1 for the aLogP and 1 for the molecular weight as these factors are the only possible rounding error for a single product. The percentage of differing entries in the distributions and the average errors show that the approximate approach gives a good estimate of the exact distribution. The approximate approach can thus be used when operating on ultralarge chemical spaces containing trillions of compounds or more. For smaller chemical spaces with a number of products in the single digit billions, the exact distribution calculation will usually be fast enough.

CONCLUSION

In this work, we described SpaceProp, a novel method for the calculation of physicochemical property distributions of combinatorial fragment spaces. With our combinatorial approach, we are able to calculate distributions of ultralarge chemical spaces that are far too large to be practically enumerated.

The method was validated based on five physicochemical properties, demonstrating SpaceProp gives an exact distribution across all products of a chemical space. The property distributions of three well-known make-on-demand vendor spaces and a publicly available chemical space were discussed. REAL Space has a high percentage of drug-like molecules following Lipinski's "rule of five". In addition to reported synthetic success rates,[37] this makes the chemical space of 20 billion make-on-demand compounds an ideal candidate for the incorporation into drug discovery projects. Examples of beyond "rule of five" compounds belonging to well-known drug families were extracted from each of the chemical spaces. This emphasizes the diversity of the considered chemical spaces and their broad scope of potential applications.

The SpaceProp algorithm can in principle be applied to other molecular properties, such as the number of rotatable bonds or the topological polar surface area (TPSA).[38] These properties could be used to analyze conformational complexity of ultralarge chemical spaces for the first time. A prerequisite for the applicability of SpaceProp is some degree of additivity of a property. If the sum of fragment property values has no connection to the derived product's property value, SpaceProp cannot operate on the fragment level. Because of this, strongly nonadditive molecular properties like shape[39] or activity data[40] might not be compatible with a combinatorial approach like SpaceProp. In addition, the range of fragment property values must be small enough, such that enough fragments are grouped together. Otherwise, the run time of SpaceProp scales with the number of products in a chemical space. The multiplied distribution of aLogP with five decimal places and molecular weight with four decimal places is most likely not practically computable with SpaceProp due to its large value range. The value range would increase even more, if all four properties of Lipinski's "rule of five" would be combined to a multiplied distribution for the exact calculation of the relative and absolute number of products meeting the rules criteria. However, lower bounds to these numbers were calculated with the existing individual distributions.

In contrast to existing approaches, SpaceProp employs a combinatorial strategy and can determine the exact overall product property distribution of a combinatorial chemical space without considering each of its products individually. If we compare the run time of SpaceProp to the sequential approach using library enumeration, the advantages become clear from a computer scientist's point of view. SpaceProp takes 2 h to calculate the five exact property distributions of the 100 trillion products contained in KnowledgeSpace using a 3.5 GHz CPU and four parallel threads. We translate the run time to CPU cycles in order to make the comparison to the sequential approach more clear. Using all four parallel threads, up to 16 billion CPU cycles can be completed within a second. Even if enumeration of all products of KnowledgeSpace would practically be possible, more than eight products must be processed within 10 CPU cycles in the sequential approach, to finish calculation in 2 h like SpaceProp. Only loading one

Table 2. Computing Times of the Exact and Approximate Distribution Calculation for REAL Space, GalaXi, CHEMriya, and KnowledgeSpace[a]

fragment space	exact (s)	approximate (s)	aLogP errors	molecular weight errors
REAL Space	191	130	0.009% (9×10^{-6})	0.821% (0.00821)
GalaXi	27	10	0.004% (4×10^{-6})	0.618% (0.00618)
CHEMriya	269	17	0.003% (3×10^{-6})	0.087% (0.00087)
KnowledgeSpace	7353	77	0.016% (1.6×10^{-5})	0.041% (0.00041)

[a]In addition, the percentage of error in distributions is displayed for the approximate aLogP and molecular weight calculation as well as the average error across all entries in the distributions enclosed in brackets.

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

product from disk and calculating all five properties for it most likely takes several hundred CPU cycles. To calculate the same distribution in 77 s like the approximate version of SpaceProp, more than 80 products would have to be processed within a single CPU cycle. With these numbers in mind, we think approaches of combinatorial character like SpaceProp are a necessity to process chemical libraries that are ever growing in size.

SOFTWARE AND DATA AVAILABILITY

SpaceProp will be integrated into the SpaceCompare tool which is available for Linux, MacOS, and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and is free for academic use and evaluation purposes. KnowledgeSpace in its topological fragment space representation can be accessed at https://www.zbh.uni-hamburg.de/forschung/amd/datasets.html. For a detailed description of the chemical spaces used in this publication, see https://www.biosolveit.de/infiniSee#chemical_spaces. Enamine's REAL Space, WuXi's GalaXi Space, and OTAVA's CHEMriya Space cover commercially available on-demand molecules and can be obtained from the respective compound providers or BioSolveIT GmbH.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c00334.

Method for calculating molecular weight and number of hydrogen bond donor distribution of topological fragment spaces; property distributions for Enamine's REAL Space, GalaXi, CHEMriya, and KnowledgeSpace; topological fragment spaces and enumerated products for two chemical subspaces of KnowledgeSpace together with generated distributions; and percentage of compounds fulfilling "rule of five" contraints for Enamine's REAL Space, GalaXi, CHEMriya, and KnowledgeSpace (ZIP)

AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *Universität Hamburg, ZBH - Center for Bioinformatics, Research Group for Computational Molecular Design, 20146 Hamburg, Germany;* orcid.org/0000-0002-9553-6531; Email: rarey@zbh.uni-hamburg.de

Authors

Louis Bellmann − *Universität Hamburg, ZBH - Center for Bioinformatics, Research Group for Computational Molecular Design, 20146 Hamburg, Germany;* orcid.org/0000-0002-7920-1889

Raphael Klein − *BioSolveIT GmbH, 53757 Sankt Augustin, Germany;* orcid.org/0000-0002-5087-8730

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c00334

Notes

The authors declare no competing financial interest.

REFERENCES

(1) Leo, A.; Hansch, C.; Elkins, D. Partition Coefficients and Their Uses. *Chem. Rev.* **1971**, *71*, 525−616.

(2) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3−25.

(3) Wiener, H. Structural Determination of Paraffin Boiling Points. *J. Am. Chem. Soc.* **1947**, *69*, 17−20.

(4) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Model.* **1999**, *39*, 868−873.

(5) Leo, A.; Jow, P.; Silipo, C.; Hansch, C. Calculation of Hydrophobic Constant (log P) From π and f Constants. *J. Med. Chem.* **1975**, *18*, 865−868.

(6) Klopman, G.; Li, J.-Y.; Wang, S.; Dimayuga, M. Computer Automated log P Calculations Based on an Extended Group Contribution Approach. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 752−781.

(7) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20-A Free Ultralarge-scale Chemical Database for Ligand Discovery. *J. Chem. Inf. Model.* **2020**, *60*, 6065−6073.

(8) Hoffmann, T.; Gastreich, M. The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discovery Today* **2019**, *24*, 1148.

(9) Brenner, S.; Lerner, R. A. Encoded Combinatorial Chemistry. *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89*, 5381−5383.

(10) Rarey, M.; Stahl, M. Similarity Searching in Large Combinatorial Chemistry Spaces. *J. Comput. Aided Mol. Des.* **2001**, *15*, 497−520.

(11) Schmidt, R.; Klein, R.; Rarey, M. Maximum Common Substructure Searching in Combinatorial Make-on-Demand Compound Spaces. *J. Chem. Inf. Model.* **2022**, *62*, 2133.

(12) Bellmann, L.; Penner, P.; Rarey, M. Topological Similarity Search in Large Combinatorial Fragment Spaces. *J. Chem. Inf. Model.* **2021**, *61*, 238−251.

(13) Bellmann, L.; Penner, P.; Gastreich, M.; Rarey, M. Comparison of Combinatorial Fragment Spaces and Its Application to Ultralarge Make-on-Demand Compound Catalogs. *J. Chem. Inf. Model.* **2022**, *62*, 553−566.

(14) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with Feature Trees. *J. Chem. Inf. Model.* **2009**, *49*, 270−279.

(15) Boehm, M.; Wu, T.-Y.; Claussen, H.; Lemmen, C. Similarity Searching and Scaffold Hopping in Synthetically Accessible Combinatorial Chemistry Spaces. *J. Med. Chem.* **2008**, *51*, 2468−2480.

(16) Warr, W. A. *NIH Meeting "Ultra-Large Databases in Chemistry"*. 2020. https://chemrxiv.org/articles/preprint/Report_on_an_NIH_Workshop_on_Ultralarge_Chemistry_Databases/14554803 (accessed 2022-05-25).

(17) Grebner, C. *Webinar: "Exploration and Mining of Large Virtual Chemical Spaces"*. 2018. https://youtu.be/fMrI11SXwpU (accessed 2022-05-25).

(18) Detering, C.; Claussen, H.; Gastreich, M.; Lemmen, C. KnowledgeSpace-a Publicly Available Virtual Chemistry Space. *J. Cheminformatics* **2010**, *2*, O9.

(19) *Enamine REAL Space*. https://enamine.net/library-synthesis/real-compounds/real-space-navigator (accessed 2021-08-05).

(20) *GalaXi Space*. https://www.labnetwork.com/frontend-app/p/#!/library/virtual (accessed 2021-08-05).

(21) *CHEMriya Space*. https://www.otavachemicals.com/products/chemriya (accessed 2021-08-23).

(22) Miyaura, N.; Suzuki, A. Stereoselective Synthesis of Arylated (E)-Alkenes by the Reaction of Alk-1-Enylboranes with Aryl Halides in the Presence of Palladium Catalyst. *J. Chem. Soc., Chem. Commun.* **1979**, 866−867.

(23) D'Alterio, M. C.; Casals-Cruañas, È.; Tzouras, N. V.; Talarico, G.; Nolan, S. P.; Poater, A. Mechanistic Aspects of the Palladium-Catalyzed Suzuki-Miyaura Cross-Coupling Reaction. *Chem.—Eur. J.* **2021**, *27*, 13481−13493.

(24) Blangetti, M.; Rosso, H.; Prandi, C.; Deagostino, A.; Venturello, P. Suzuki-Miyaura Cross-Coupling in Acylation Reactions, Scope and Recent Developments. *Molecules* **2013**, *18*, 1188−1213.

(25) Diels, O.; Alder, K. Synthesen in der Hydroaromatischen Reihe. *Justus Liebigs Annalen der Chemie* **1928**, *460*, 98−122.
(26) Ghose, A. K.; Crippen, G. M. Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships I. Partition Coefficients as a Measure of Hydrophobicity. *J. Comput. Chem.* **1986**, *7*, 565−577.
(27) Ghose, A. K.; Crippen, G. M. Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships. 2. Modeling Dispersive and Hydrophobic Interactions. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 21−35.
(28) Ghose, A. K.; Pritchett, A.; Crippen, G. M. Atomic Physicochemical Parameters for Three-Dimensional Structure-Directed Quantitative Structure-Activity Relationships III: Modeling Hydrophobic Interactions. *J. Comput. Chem.* **1988**, *9*, 80−90.
(29) Mannhold, R.; Poda, G. I.; Ostermann, C.; Tetko, I. V. Calculation of Molecular Lipophilicity: State-Of-The-Art and Comparison of log P Methods on More than 96,000 Compounds. *J. Pharm. Sci.* **2009**, *98*, 861−893.
(30) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the Size of Drug-Like Chemical Space Based on GDB-17 Data. *J. Comput. Aided Mol. Des.* **2013**, *27*, 675−679.
(31) Bellmann, L.; Penner, P.; Rarey, M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* **2019**, *59*, 4625−4635.
(32) *eMolecules*. https://www.emolecules.com/ (accessed 2021-08-05).
(33) Mensa, B.; Howell, G. L.; Scott, R.; DeGrado, W. F. Comparative Mechanistic Studies of Brilacidin, Daptomycin, and the Antimicrobial Peptide LL16. *Antimicrob. Agents Chemother.* **2014**, *58*, 5136−5145.
(34) Nieto-Garai, J. A.; Glass, B.; Bunn, C.; Giese, M.; Jennings, G.; Brankatschk, B.; Agarwal, S.; Börner, K.; Contreras, F. X.; Knölker, H.-J.; Zankl, C.; Simons, K.; Schroeder, C.; Lorizate, M.; Kräusslich, H.-G. Lipidomimetic Compounds Act as HIV-1 Entry Inhibitors by Altering Viral Membrane Structure. *Front. Immunol.* **2018**, *9*, 1983.
(35) Ishida, T.; Ciulli, A. E3 Ligase Ligands for PROTACs: How They Were Found and How to Discover New Ones. *SLAS Discov* **2021**, *26*, 484−502.
(36) Vagner, J.; Qu, H.; Hruby, V. J. Peptidomimetics, a Synthetic Tool of Drug Discovery. *Curr. Opin. Chem. Biol.* **2008**, *12*, 292−296.
(37) Klingler, F.-M.; Gastreich, M.; Grygorenko, O. O.; Savych, O.; Borysko, P.; Griniukova, A.; Gubina, K. E.; Lemmen, C.; Moroz, Y. S. SAR by Space: Enriching Hit Sets from the Chemical Space. *Molecules* **2019**, *24*, 3096.
(38) Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* **2000**, *43*, 3714−3717.
(39) Sauer, W. H.; Schwarz, M. K. Molecular Shape Diversity of Combinatorial Libraries: a Prerequisite for Broad Bioactivity. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 987−1003.
(40) Kramer, C.; Fuchs, J. E.; Liedl, K. R. Strong Nonadditivity as a Key Structure−Activity Relationship Feature: Distinguishing Structural Changes from Assay Artifacts. *J. Chem. Inf. Model.* **2015**, *55*, 483−494.

https://doi.org/10.1021/acs.jcim.2c00334
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Hamburg, den 9. Juni 2022

Louis Bellmann