Supporting Experts in Early Drug Discovery: Algorithmic and Visualization Approaches for Improved Synthetic Accessibility

Cumulative Dissertation

with the aim to achieve the degree

Dr. rer. nat.

at the Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

of Universität Hamburg

submitted by

Uschi Dolfus

born in Aachen

Hamburg, August 2024

1. Reviewer: Prof. Dr. Matthias Rarey
2. Reviewer: Prof. Dr. Andrew Torda
3. Reviewer: Prof. Dr. Paul Czodrowski

Date of thesis defense: 12.12.2024

Kurzfassung

Seitdem ein großer Teil der frühen Medikamententwicklung am Computer stattfindet, muss sichergestellt werden, dass die virtuell entwickelten Strukturen nicht nur theoretisch wirksam, sondern auch praktisch herstellbar sind. Um dies zu gewährleisten, muss möglichst früh die Synthetisierbarkeit von Wirkstoffkandidaten mit in den Design-Prozess einbezogen werden. Synthetisierbarkeit ist jedoch eine komplexe Fragestellung, die ein tiefes Verständnis von Chemie, praktische Erfahrung und häufig auch Kreativität abverlangt; alles Eigenschaften, die sich schwer automatisieren lassen. Aus diesem Grund befassen sich bestehende Methoden und Ansätze aus unterschiedlichen Richtungen mit dem Problem und versuchen, verschiedene Teilaspekte zu lösen. In dieser Arbeit wurden Methoden entwickelt, um die Integration der Synthetisierbarkeit in den frühen Medikamentenentwurf zu fördern. Ziel ist es, synthetische Chemiker während der Designphase neuer Strukturen maschinell zu unterstützen.

Die Veränderung und Optimierung von Leitstrukturen ist ein grundlegendes Konzept für die moderne Arzneimittelentwicklung. Dazu gehört die systematische Erforschung und Verfeinerung der chemischen Struktur, um ihre pharmakologischen Eigenschaften wie Wirksamkeit, Selektivität und metabolische Stabilität zu verbessern. In einem ersten Ansatz wurde eine Methode zur Generierung synthetisch zugänglicher Strukturanaloga ausgehend von einer Leistruktur entwickelt, um die effiziente Synthese von Strukturanaloga während des Design-Make-Test-Analyse Zyklus zu ermöglichen. Für den entwickelten Ansatz wurde entschieden, explizit keine neuen, künstlich konstruierten Synthesewege zu erstellen, sondern dem synthetischen Chemiker die Auswahl eines geeigneten Syntheseweges zu überlassen und nur Strukturanaloga zu generieren, welche über den gewählten Syntheseweg wahrscheinlich erstellbar sind. Mit Synthesewegen, die bereits in den eigenen Laboren getestet wurden und gut etabliert sind, bekommt man eine Reihe von Strukturanaloga, welche nicht nur gewünschte physikochemische Eigenschaften aufweisen, sondern möglichst rasch und effizient durch Experimente getestet werden können. Dabei können gewünschte Reaktanten ebenfalls individuell ausgewählt werden, um die eigene Bibliothek an Bausteinen zu bevorzugen oder nur kommerziell erwerbare zuzulassen. Die Fähigkeit der Methode nicht nur synthetisch zugängliche, eigenschafts-spezifische, chemische Räume aus Strukturanaloga zu erstellen, sondern ebenfalls für die Analyse von dem synthetischen Aufwand von Molekülreihen eingesetzt zu werden, wird gezeigt.

Das Konzept von Synthesefähigkeit basiert auf chemischen Reaktionen. Generische Reaktionsmuster sind häufig gewählte Formate um chemische Reaktionen darzustellen, sodass ein Computer diese lesen, analysieren und anwenden kann. Die korrekte Erstellung und das menschliche Verständnis dieser Muster ist essentiell, damit der Computer die richtigen Anweisungen bekommen kann. Allerdings sind diese Zeichenketten-basierten Darstellungen selbst für trainierte Chemiker oder Entwickler oft schwer zu lesen und zu interpretieren. Um die Verwendung und Verbreitung dieser Darstellungen chemischer Reaktionen zu unterstützen und eine einfache Möglichkeit zu schaffen, diese Muster zu verstehen, wurde ein Algorithmus für die Visualisierung dieser Muster entwickelt. Die Einfachheit der Interpretation von Reaktionsmustern mit Hilfe der gewählten Visualisierungsstrategie wird an verschiedenen Beispielen erläutert. Zudem werden zwei verbreitete Reaktionsdatensätze vollständig visualisiert und bereit gestellt.

Synthesewege werden üblicherweise konstruiert und angepasst, indem Chemiker ihr umfangreiches Wissen über gängige Synthesemethoden und ihre praktische Erfahrung einsetzen. Dies umfasst die sorgfältige Auswahl von Ausgangsmaterialien, die Berücksichtigung gut bekannter chemischer Reaktionen und die Vermeidung von Strukturen, die bekanntermaßen schwer zu synthetisieren sind. In einem dritten Ansatz ist eine Methode entwickelt worden, um synthetische Wege zu modifizieren und an individuell enstehende Bedürfnisse und Gegebenheiten anzupassen. Dabei wird Funktionalität bereit gestellt, um sowohl alle Strukturen, als auch alle Reaktionen in einem Syntheseweg auszutauschen. Berechenbare physiko-chemische Eigenschaften aller Strukturen können beeinflusst werden. Die einzelen Anpassungsmöglichkeiten werden anhand von Beispielen erläutert. Zudem wird ein weiterer Anwendungsfall gezeigt, bei dem die Methode genutzt wird, um zu analysieren, welche Gerüststrukturen aus einer gegebenen Menge für eine spezifische Zielstruktur für "Scaffold-Hopping" synthetisch zugänglich sind.

Abstract

Since early drug development largely takes place on the computer, it must be ensured that the virtually developed structures are not only theoretically effective but also practically producible. To ensure this, the synthesizability of candidates must be included in the design process as early as possible. However, synthesizability is a complex issue that requires a deep understanding of chemistry, practical experience, and often creativity; all characteristics that are difficult to automate. For these reasons, existing methods and approaches address the problem from different directions and attempt to solve different aspects of it. In this work, algorithms were developed to promote the integration of synthesizability into early drug design. The aim is to support synthetic chemists during the design phase of new structures.

The modification and optimization of lead structures is a fundamental concept for modern drug development. This includes the systematic exploration and refinement of the chemical structure to improve its pharmacological properties such as efficacy, selectivity and metabolic stability. In the first approach, a method for generating synthetically accessible structural analogues was developed, starting with a lead structure, to enable the efficient synthesis of structural analogues during the design-make-test-analysis cycle. For the developed approach, it was decided explicitly not to create new, artificially constructed synthetic pathways, but to leave the selection of a suitable synthetic route to the synthetic chemist and only generate structural analogues that can in theory be produced via the selected pathway. With synthetic pathways that have already been tested in own laboratories and are well established, structural analogues can be generated, that not only have the desired physicochemical properties but can be tested quickly and efficiently in experiments. Desired reactants can be individually selected to favor one's own library of building blocks or to allow only commercially available ones. The ability of the method not only to create synthetically accessible, property-specific chemical spaces of structural analogues, but also to be used for the analysis of the synthetic effort of molecule series is demonstrated.

The concept of synthesizability is based on chemical reactions. Generic reaction patterns are commonly chosen formats to represent chemical reactions so that a computer can read, analyze, and apply them. The correct creation and human understanding of these patterns is essential for the computer to receive the correct instructions. However, these string-based representations are often difficult to read and interpret, even for trained chemists or developers. To support the use and distribution of these representations of chemical reactions and to provide an easy way to understand them, an algorithm for the visualization of chemical reaction patterns has been developed. The simplicity of interpreting reaction patterns using the chosen visualization strategy is explained using various examples. In addition, two common reaction data sets are provided fully visualized.

Synthetic routes are typically constructed and adapted by chemists using their extensive knowledge of common synthesis methods and practical experience. This includes careful selection of starting materials, consideration of well-known chemical reactions, and avoidance of structures that are notoriously difficult to synthesize. In a third approach, a method has been developed to modify synthetic routes and adapt them to individual needs and circumstances. Functionality is provided to exchange all structures as well as all reactions in a synthetic pathway. The physicochemical properties of all structures can be influenced. The individual customization options are explained using examples. In addition, a further use case is presented in which the method is used to analyze which scaffold structures from a given set are synthetically accessible for scaffold hopping with a specific target structure.

Acknowledgements

I would like to take this opportunity to thank all those who have supported me over the past few years and who have made it possible for me to present this thesis.

First of all, I would like to thank my supervisor, Matthias Rarey, for his continuous support and guidance, and for always keeping his door open to discuss any and all challenges that I have faced.

I would like to thank Hans Briem for his excellent mentoring, his perspectives beyond the academic world, and for his support far past the official end of our collaboration.

Further, I would like to thank Bayer AG for funding this project.

A big thank you goes to the members of the AMD working group and the occasional students for the great years of scientific discussions, the always available advice and the extra hours of leisure activities. Special thanks go to Torben, Eddie and Emanuel, who have answers to all my problems, whether I needed insight into a chemist's head or how to brew my coffee. Thank you for your general support and encouragement in my life. I would also like to thank Melanie, who always helped where she could and always offered a listening ear.

Thank you to my parents and brothers for giving me the feeling that no matter what happens, you will support me. Thank you, Rafael, for always being there for me, for never stopping to talk against my head, for reassuring me that I can do whatever I want, and for reminding me how beneficial breaks can be. And thanks to Lola, Raya and Toni, without you it would have been much more efficient to write this thesis, but I probably wouldn't have made it through.

Contents

Chapter 1

Introduction

To save time and resources, the modern drug development process relies on the results of computer-assisted methods. Virtually designed candidates need to be producible in the laboratory. Even drug candidates with ideal pharmacological properties are worthless if they cannot be synthesized. In general, it is more difficult to start the DMTA (design-make-test-analyze) cycle with a small molecule drug candidate with ideal pharmacological properties but low synthetic accessibility than with candidates with a less favorable pharmacological profile but higher synthetic accessibility. It is often possible to identify and test strategies to circumvent undesirable properties if the compounds can be synthesized. However, if the compounds are difficult to synthesize, the testing phase is restricted to a limited number of options. [1] A popular example is the generation of Pfizer's clinical candidate SARS-CoV-2 (severe acute respiratory syndrome coronavirus-2) inhibitor for the treatment of COVID-19. In the corresponding publication by Owen et al. [2], they explicitly state that a candidate with high synthetic accessibility was preferred over candidates with better activity but lower synthetic accessibility. This had an impact not only on the design phase, which was under great time pressure but also on the rapid production of the drug once it was on the market. [2] Nevertheless, synthesizability is still sometimes overlooked during *in silico* drug design or treated as an afterthought once the design phase is complete. This problem is particularly common in modern generative design approaches using machine learning algorithms, where synthetic accessibility is often neglected. [3]

This thesis is a contribution to address the challenge of synthesizability during the *in silico* drug design process. Three algorithmic approaches were developed, resulting in two software applications. First, an algorithm to generate data sets of synthetically accessible structural analogues was designed, with the motivation to make the DMTA cycle more efficient. Second, a visualization technique for reaction patterns was developed, to support the understanding and analysis of essential data for *in silico* synthesis design. Third, algorithmic solutions to modify synthetic pathways were integrated into the software for generation of synthetically accessible structural analogues. The focus here is on the integration of individual, case-specific needs of the synthetic chemist.

In the following, relevant background information as well as state-of-the-art methods for different approaches to overcome the bottleneck of synthetic accessibility of drug candidates are presented. This is followed by a detailed motivation for all three algorithmic methods in this thesis, together with a categorization of which synthetic feasibility problems are addressed. Finally, the structure of the remaining part of this thesis is explained.

1.1 Chemical Data and Representations

In computer-aided drug discovery, having languages understandable by both humans and machines, representing and expressing chemical data, is essential for a successful design process. For this thesis, the encoding of molecular and reaction information into machine-readable formats is fundamental. Therefore, the following sections will provide an overview of relevant concepts and algorithms. In addition, significant data sets will be presented and visualization techniques will be discussed.

1.1.1 Molecular Representations

Wigh et al. [4] identified four classes of molecular representations that are relevant to computational chemistry today: String, compound table, feature-based, and computer-learned representations. Molecular string representations consist of ASCII characters and are constructed according to grammatical rules. Chemical compound table representations provide atomic coordinates and bonding information in tabular form. Feature-based molecular representations list relevant and representative molecular properties in various encoded formats. Computer-learned molecular representations are numerical formats generated by neural network architectures. In this thesis, only string representations of molecules written in the Simplified Molecular Input Line Entry System (SMILES) [5, 6] are used. Other string-based representations of molecules are the IUPAC (International Union of Pure and Applied Chemistry) [7, 8] nomenclature or the InChI (International Chemical Identifier) [9] language. Established examples from the other classes are the Structural Data File (SDF), a chemical table representation, Extended-Connectivity Fingerprints [10], a feature-based representation, and Continuous and Data-Driven Descriptors [11], a computer-learned molecular descriptor.

The SMILES language is one of the most common approaches for representing molecules, as it is easy to read for both computers and humans. Essential information can be provided in a compact way. Atoms are written as one or two letters based on the periodic table. Various characters are used for bonds, rings, branching or to describe stereochemistry. A fixed vocabulary and grammatical rules enable standardized use. [5]

While the SMILES language describes individual, specific molecules, the SMILES Arbitrary Target Specification (SMARTS) [12] language, an extension of SMILES, was introduced to enable the representation of molecular patterns. The language permits the generation of queries for pattern matching and substructure searches within molecular structures. Placeholders and wildcards, as well as logical operators and recursive expressions, can be used to create patterns at different levels of specificity. [12]

1.1.2 Chemical Reaction Representations

Both SMILES and SMARTS are easily adapted to include reaction representations. For reasons of readability, popularity, and consistency chemical reactions are presented in this thesis using the SMILES and SMARTS language. The resulting patterns are called Reaction SMILES, Reaction SMARTS, or are written in the SMIRKS [13] language, an extension of the SMARTS language. Reaction SMILES and Reaction SMARTS inherit the specific requirements and properties of the languages on which they are based. This means that Reaction SMILES present specific chemical reactions and Reaction SMARTS are generic reaction patterns.

A generic reaction pattern is a way of representing a class or type of chemical reaction without precisely specifying the actual chemical structures involved. A generic reaction pattern represents a class or type of chemical reaction that describes the reactive groups of a reaction without specifying the entire chemical structures involved. It serves as a template or generalized form that can represent a broad category of reactions and allows for a more abstract description. Therefore, generic reactions are often called reaction templates. [14]

Reaction SMILES, Reaction SMARTS and SMIRKS patterns can contain atom mapping in the form of atom labels. This allows matching atoms between reactants and product structures to be specified. Usually, these labels are used to specify the reaction center, which is the site within a molecule where chemical bonds are broken or formed and where atoms undergo changes in connectivity or electronic configuration [15].

The SMIRKS language is a restricted form of Reaction SMARTS designed to create patterns that are used for generating new reactions, manipulating molecules, and facilitating the creation of new molecular structures on a large scale. Additional rules ensure the interpretability of a reaction graph and the derivability of atomic and bond changes directly from the pattern [13]. The specific rules are discussed in Section 2.4 and can be found in B.2.2. However, Reaction SMARTS and SMIRKS are often used synonymously in computer-aided drug development.

Typically, a reaction is written as a concatenation of structure patterns. Reactants and products or their representing patterns are separated by two arrows ('≫') and the individual structural patterns, e.g. in the case of several reactants, by a dot ('.'). Figure 1.1 shows an example of a specific chemical reaction, an O-acylation to ester reaction, together with a matching generic reaction pattern. Other popular reaction representations are the MDL reaction file format (.rxn) [16], the International Chemical Identifier for Reactions (RInChl) [17], or the CHMTRN/PATRAN [18] language.

1.1.3 Reaction Data Sets

One of the main source of available reaction data is the United States Patent and Trademark Office (USPTO) [20]. Over the years, both publicly and commercially available subsets have been extracted. Examples of publicly available data sets are the general purpose USPTO_FULL [21] or the USPTO_MIT [22]. Both are based on the extraction of chemical reaction data from the USPTO by Lowe [23]. In the USPTO_FULL dataset, reactions with multiple products are represented multiple times, each instance containing only a single product. After removing all reactions which contain wrong atom mappings, the USPTO_FULL data set includes roughly one million unique reactions[21]. The USPTO_MIT dataset removed contextual chemical information and saved reactions as reactants and products only. Duplicates and reactions with incorrect atom mappings are removed, resulting in 140,284 unique reaction templates [22].

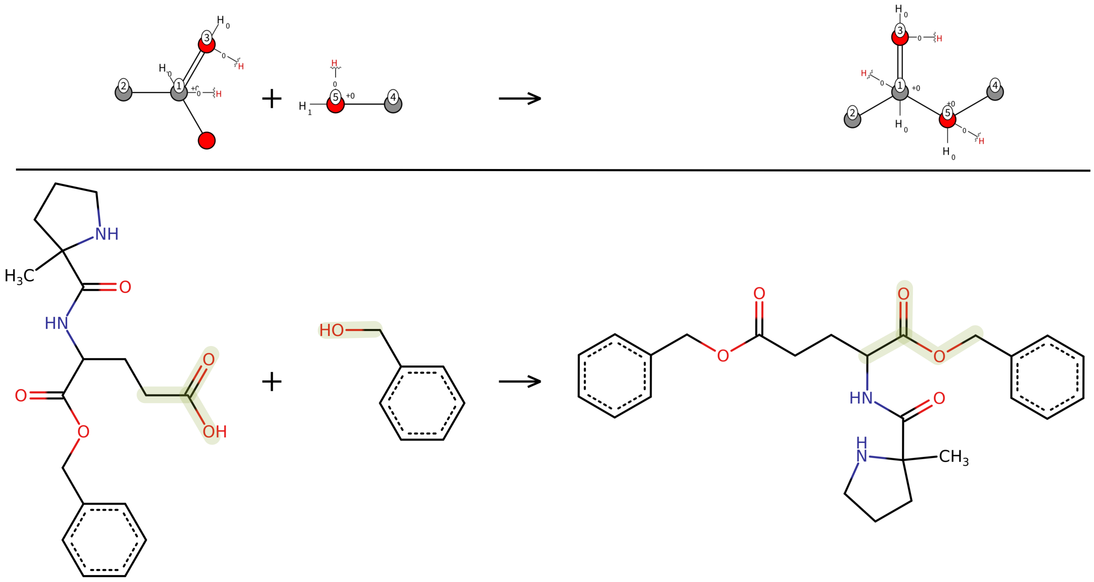


Figure 1.1: Visualization of an O-acylation to ester reaction, both a reaction with specific structures (bottom), as well as a matching reaction pattern (up) are visualized. The pattern used for this visualization was adapted from data provided with the AiZynthFinder [19] software.

In addition, several other subsets with specific purposes were extracted. For example, Schneider et al. [24] provides a subset that includes reaction classifications. More recently, Schwaller et al. [25] designed a dataset based on the USPTO, consisting of the 1000 most frequent reaction templates utilized as reaction classes. Other smaller data sets, consisting of hand-written reaction rules, are the organic synthesis reactions presented by Hartenfeller et al. [26] or the SMARTS collection of the BRICS [27] algorithm (see Section 1.3 for further information).

Commercially available chemical reaction data are provided for example by NextMove with the Pistachio software [28] or the Reaxys [29] and SciFinder [30] databases. Another source of information about reaction data is general literature-extracted datasets, where all information is displayed as text and has to be converted into chemical data. Often, this involves a cleaning and interpretation step [31]. Lastly, a new platform to share and access reaction data, called the Open Reaction Database (ORD) [32] has been published recently. Developed as an open-access platform, ORD serves as a central repository for reaction information, hopefully facilitating data sharing, collaboration, and analysis within the scientific community [33]. In general, only positive reactions with high yields are usually available in the presented reaction datasets. However, it would be highly useful for all scientists, especially in the context of machine learning processes, to have access to failed reactions or reactions with low yields. The challenge is that it is often unclear whether poor results stem from human error and limitations in experimental tools, or whether the reaction is inherently difficult or impractical to carry out.

1.1.4 Visualization of Chemical Data

A practical way to enable easy understanding of the described chemical pattern languages (see Section 1.1.1 and 1.1.2) without losing the computer readability, is automated visualization. Nowadays software can display chemical data in 2D, or 3D [34] and even in virtual reality [35]. Since 2D representations are sufficient for a complete description of chemical reactions, other representations are not considered further. Numerous different visualization software exists for molecules, both commercially available as well as open-source and web-based: Examples are PubChem Sketcher [36], ChemSpider [37], MolView [38], ChemAxons molecule visualizer (Marvin) [39] or the visualization components of RDKit [40, 41]. However, the more abstract the language gets, the sparser the software tools available. There are few tools able to handle molecular patterns and even fewer that can visualize generic reaction patterns. In the following three popular options will be presented; the visualization components in RDKit, [40, 41] MarvinSketch [42] and the SMARTSviewer [43]/ ReactionViewer [D1]. To discuss the different visualization strategies and abilities all three tools got four strings describing chemical data with different abstraction levels:

1. A SMILES string describing the structure of Caffeine
   `Cn1cnc2c1c(=O)n(C)c(=O)n2C`

2. A Reaction SMILES string of esterification extracted from the DayLight documentation [44]
   `(C(=O)O).(OCC)≫(C(=O)OCC).(O)`

3. A SMARTS pattern describing a thiazene extracted from a collection of Pan Assay Interference Compounds (PAINS) by Baeli and Holloway [45]
   `[#6]-1(=[#6](-![#6]=[#7])-[#16]-[#6](-[#7]-1)=[#8])-[$([F,Cl,Br,I]),$([#7+](:[#6]):[#6])]`

4. A Reaction SMARTS pattern describing a Niementowski quinazoline reaction provided by Hartenfeller et al. [26]
   `[c:1](-[C;$(C-c1ccccc1):2](=[OD1:3])-[OH1]):[c:4](-[NH2:5]).[N;!H0;!$(N-N);!$(N-C=N);!$(N(-C=O)-C=O):6]-[C;H1,$(C-[#6]):7]=[OD1]≫[c:4]2:[c:1]-[C:2](=[O:3])-[N:6]-[C:7]=[N:5]-2`

In the following, the different methods are described shortly, and the visualized molecular data is shown. A comparative discussion can be found in Chapter 2.

1.1.4.1 RDKit

RDKit [40, 41] is an open-source cheminformatics software library. It provides a set of diverse functionalities to work on research questions regarding topics from computational chemistry to molecular modeling. The visualization components of RDKit include several tools for generating images of chemical structures and reactions. These tools allow developers to visualize individual molecules from SMILES or SMARTS strings, highlight substructures, and display chemical reactions. Figures 1.2 and 1.3 present a visualization of the four example patterns described in the previous section generated with RDKit.

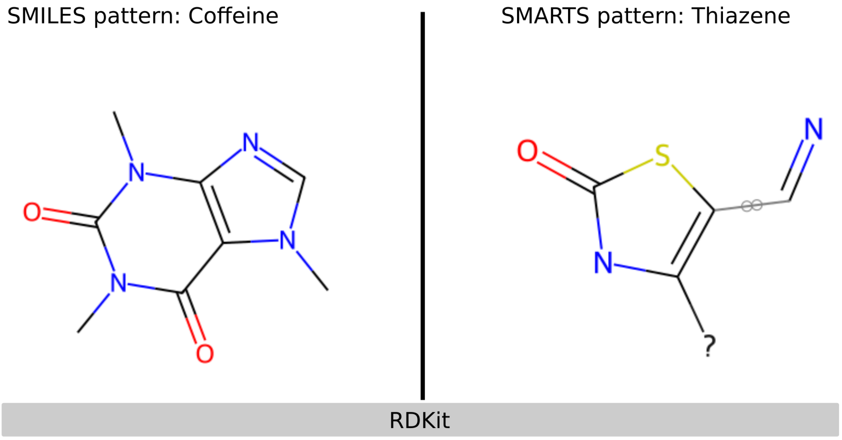


Figure 1.2: Left: Visualization of a SMILES string describing Coffeine. Right: Visualization of a PAINS SMARTS pattern extracted from Baeli et al. [45]. Both images are generated with the RDKit visualization components.

1.1.4.2 MarvinSketch

MarvinSketch [39, 42] is an advanced chemical editor developed by ChemAxon. It allows users to draw, edit, and analyze chemical structures and reactions with a graphical user interface. MarvinSketch supports a variety of chemical formats. It offers drawing tools, including 3D visualization functions. Figures 1.4 and 1.5 show a visualization of the four example patterns described in the previous section generated with the online accessible version of MarvinSketch [46].

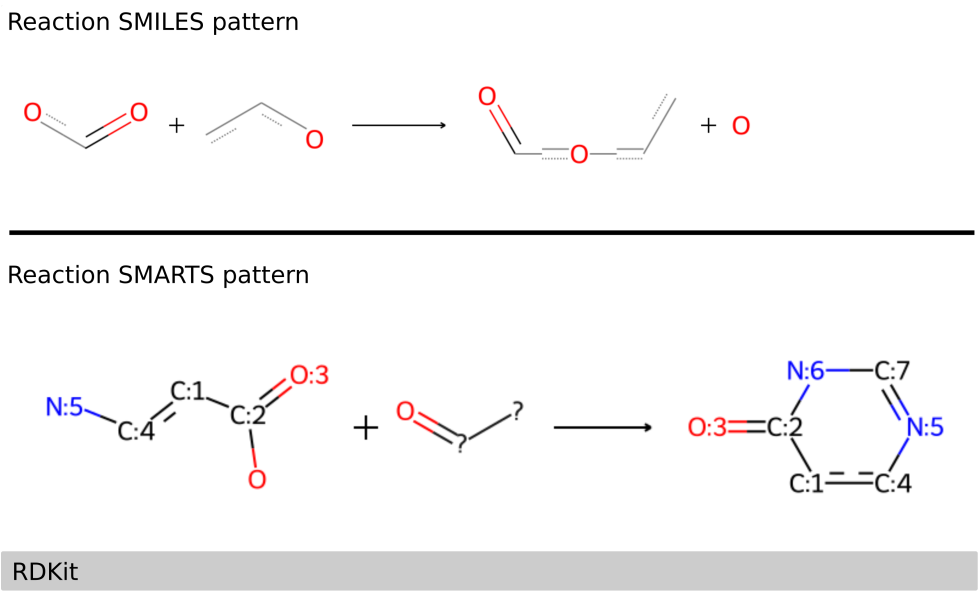


Figure 1.3: Top: Visualization of a Reaction SMILES string describing a intermolecular esterification extracted from the DayLight documentation [44]. Bottom: Visualization of a Reaction SMARTS pattern describing Niementowski quinazoline reaction provided by Hartenfeller et al. [26]. Both images are generated with the RDKit visualization components.

1.1.4.3 SMARTSviewer

SMARTSviewer [43] is a software method specially developed for visualizing SMARTS patterns (and thus SMILES expressions). In addition to the visualization, detailed explanations of each component of the SMARTS pattern are provided, thereby supporting the understanding and interpretation of complex SMARTS expressions. The SMARTSviewer functionality is available online as part of the SMARTS.plus [47, 48] software server and as a downloadable software package. As a result of the second publication [D1] of this work, SMARTS.plus has been extended to visualize reaction patterns since 2022 (for further descriptions, see Chapter 2). As a result of thesis, SMARTS.plus includes functions for visualizing reaction patterns since 2022. The corresponding tool based on SMARTSviewer is called ReactionViewer [D1]. Further details are provided in Chapter 2 or in [D1]. Figures 1.6 and 1.7 show a visualization of the four example patterns described in the previous section generated with the SMARTSviewer, respectively the ReactionViewer.

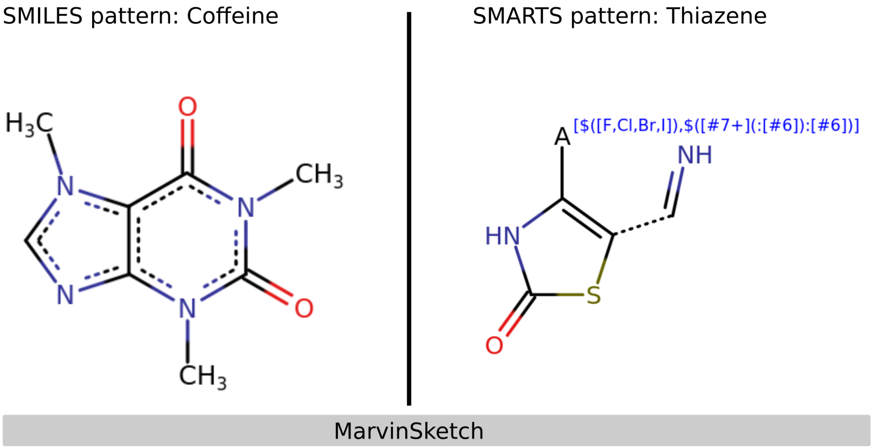


Figure 1.4: Left: Visualization of a SMILES string describing Coffeine. Right: Visualization of a PAINS SMARTS pattern extracted from Baeli et al. [45]. Both images are generated with MarvinSketch.

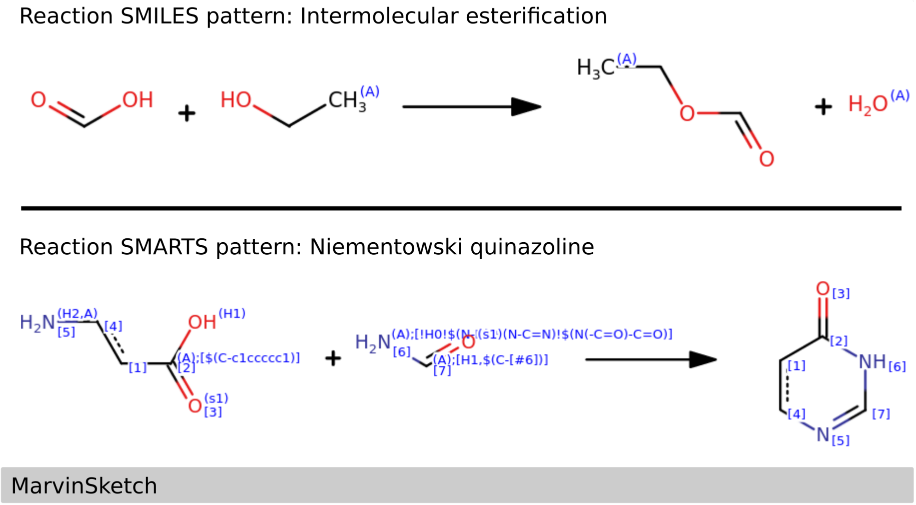


Figure 1.5: Top: Visualization of a Reaction SMILES string describing a intermolecular esterification extracted from the DayLight documentation [44]. Bottom: Visualization of a Reaction SMARTS pattern describing Niementowski quinazoline reaction provided by Hartenfeller et al. [26]. Both images are generated with MarvinSketch.

1.2 Computer-Aided Synthesis Planning

Computer-aided synthesis planning (CASP) methods are being introduced to cope with the huge amount of possible choices in synthesis planning. These computational approaches provide techniques to support the design and analysis of synthesis pathways for target compounds. In the following, basic concepts, as well as algorithmic approaches and computational methods are introduced.

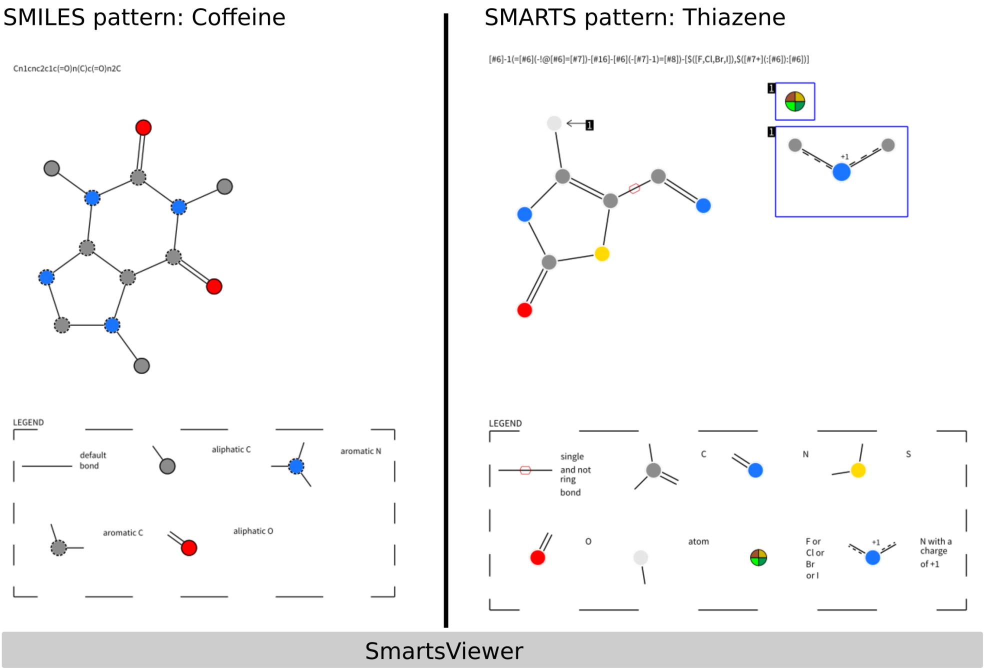


Figure 1.6: Left: Visualization of a SMILES string describing Coffeine. Right: Visualization of a PAINS SMARTS pattern extracted from Baeli et al. [45]. Both images are generated with the SMARTSviewer.

1.2.1 Forward and Retrosynthesis

This thesis includes algorithms based on both forward synthesis and retrosynthesis. Therefore, an introduction to these concepts is provided. Forward Synthesis refers to synthesizing a target compound in a process where initial building blocks are combined in a forward manner. This can include one or more chemical reactions which transform one or more reactant structures into new product structures. Retrosynthesis or backward synthesis is a technique that breaks down a target compound into smaller or simpler precursor compounds that can be handled more easily. The aim is to end up only with compounds that are trivial in their synthesis or commercially available. [49]

Retrosynthetic analysis is a formalized concept in which a retrosynthetic pathway is created. In 1963, Vléduts [50] paved the way by introducing the idea of reaction coding and computer-assisted synthesis planning. In 1969, Corey and Wipke [51] built on this foundation and introduced the first computer-aided logical retrosynthetic route planning method. The recursive partition of the target compound is achieved by applying formally reversed chemical reactions as structural transformations. When finished and

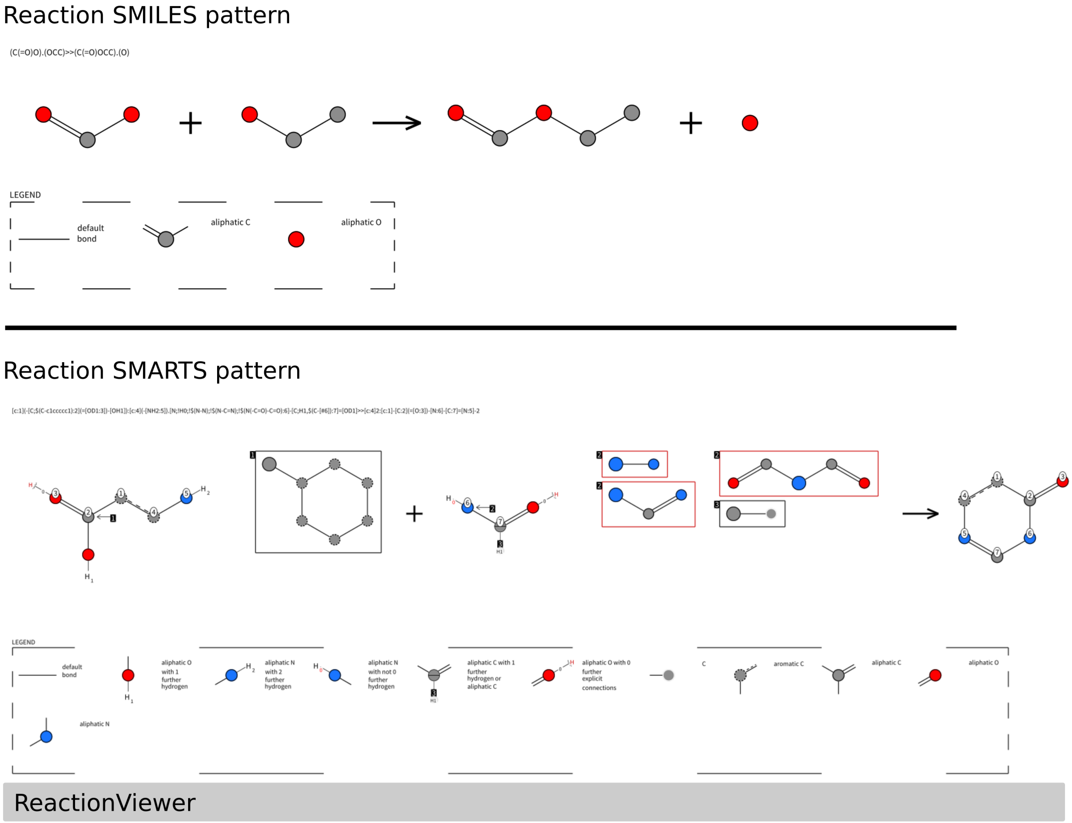

Reaction SMILES pattern
(C(=O)O).(OCC)>>(C(=O)OCC).(O)
LEGEND
default bond
aliphatic C
aliphatic O
Reaction SMARTS pattern
[c:1](-[C;$(C-c1ccccc1):2](=[OD1:3])-[OH1]):[c:4](-[NH2:5]).[N;!H0;!$(N-N);!$(N-C=N);!$(N(-C=O)-C=O):6]-[C;H1,$(C-[#6]):7]=[OD1]>>[c:4]2:[c:1]-[C:2](=[O:3])-[N:6]-[C:7]=[N:5]-2
LEGEND
default bond
aliphatic O with 1 further hydrogen
aliphatic N with 2 further hydrogen
aliphatic N with not 0 further hydrogen
aliphatic C with 1 further hydrogen or aliphatic C
aliphatic O with 0 further explicit connections
C
aromatic C
aliphatic C
aliphatic O
aliphatic N
ReactionViewer

Figure 1.7: Top: Visualization of a Reaction SMILES string describing a intermolecular esterification extracted from the DayLight documentation [44]. Bottom: Visualization of a Reaction SMARTS pattern describing Niementowski quinazoline reaction provided by Hartenfeller et al. [26]. Both images are generated with the ReactionViewer.

only simple structures are left, the target compound can be created by forward synthesis, following the generated pathway. [51]

Although CASP methods are often equated with retrosynthetic analyses, there is a much broader set of research topics related to the CASP field. One example is the recently labeled "above-the-arrow" [52] problem: predicting reaction conditions to improve efficiency, quality, and yield. This includes calculations regarding solvent or temperature. Other examples are CASP methods regarding predictions about the type of the products and their quantities, covering the reaction outcome prediction problem (see Section 1.2.3). [53]

The following sections start with a description of how to determine the synthesizability of structures using scores and a brief introduction to existing reaction outcome and yield prediction methods. Next, an overview of the algorithms used for retrosynthetic analyses in CASP methods is given, together with a summary of selected examples of software packages. Due to the lack of reference to the approaches of this thesis, additional algorithms and approaches of the CASP topic are not discussed further. Recently published, detailed reviews on the topic of CASP tools, including latest advances and successes can be found here [33, 53–55].

1.2.2 Synthetic Accessibility Scores

Synthetic accessibility scores are computational metrics that try to quantify the ease with which a molecule can be synthesized. They take various factors, such as chemical complexity, number of synthetic steps, and reagent availability, into account to provide an estimate of a compound's synthetic feasibility. There are a variety of scoring methods, from rule-based algorithms to modern machine learning models. Skoraczyński et al. [56] divides synthetic accessibility scoring approaches into structure-based and reaction-based methods. Structure-based scores focus on the analysis of the molecular structure. Factors such as the complexity of the structure, the presence of certain functional groups and associated potential challenges are evaluated. They provide an estimate of how well the architecture of a molecule can be created with a manageable number of steps, established standard reactions and based on known building blocks. Popular examples are the Synthetic Accessibility Score (SAscore) by Ertl and Schuffenhauer [57], the Synthetic Bayesian Accessibility (SYBA) score by Voršilák et al. [58] or the Graph Attention-based Assessment of Synthetic Accessibility (GASA) introduced by Yu et al. [59]. [56]

Reaction-based scores consider the availability and similarity of reaction pathways documented in chemical databases. These scores rely on databases of known chemical reactions and synthetic pathways. They evaluate how well the structure of a molecule matches reaction patterns found in these databases and indicate whether analogues reactions can be applied to the synthesis of the target molecule. Popular examples are the SCScore presented by Coley et al. [60], the retrosynthetic accessibility score (RAScore) by Thakker et al. [61] or RetroGNN by Liu et al. [62]. [56]

1.2.3 Reaction Outcome and Yield Prediction

Reaction outcome prediction methods aim to predict the most likely products of chemical reactions based on given reactants and possibly additional information. Reaction yield methods, on the other hand, calculate the percentage of reactants that are successfully converted into the desired products. With modern machine learning algorithms both tasks can be solved more successfully than ever, and therefore will be introduced shortly in the following.

Schwaller et al. [33] divides reaction outcome prediction methods three categories; template-based approaches, graph-edit approaches, and sequence-based approaches. Template-based approaches [22, 63, 64] predict reaction outcomes by matching precursors to predefined reaction patterns, often including a ranking of potential products generated from multiple pattern matches and feature engineering to refine predictions. Graph-edit-based approaches [65, 66] predict changes in molecular structures by analyzing bond modifications in the molecular graph, while sequence-based approaches [67, 68] use textual representations of molecules (e.g. SMILES) to translate precursor sequences into product sequences using natural language processing models. [33]

A simple way to evaluate reaction performance is the expected yield. Machine learning models have been developed to predict chemical reaction yields, leveraging extensive datasets and various (manually annotated) descriptors [69]. Other examples are models based on molecular fingerprints [70], or Reaction SMILES Transformer models [71]. [33]

1.2.4 Algorithmic Approaches for Retrosynthetic Analysis

Looking at the history of retrosynthetic analysis in CASP tools, it is important to distinguish between expert-driven and data-driven approaches. Early approaches relied solely on human expertise as a source of information, such as the work of Corey et al. [72]. Manually coded reaction rules were used to span the accessible space of synthetic routes. The human influence represents both the major advantages and disadvantages of this approach: Rules written by experts tend to be detailed and contain information about when, where, and how a reaction can be applied. Each rule is checked manually before it is entered into the database. However, this process is time-consuming and limits the number of reactions that can be included. In addition, expert-driven systems are often unable to provide meaningful analysis for new inputs outside of their knowledge base and can take a significant amount of time to solve complex routes. A popular example of an expert-driven system is Synthia [73], formerly known as Chematica, where chemists have been feeding their expertise into their databases for over 15 years. [53]

Data-driven approaches, on the other hand, extract or learn applicable reactions from experimentally validated chemical reaction databases. They are usually more efficient and scalable than approaches based on expert knowledge, which is particularly advantageous when incorporating individual or new data sets. Continuous updating when new reaction data is published is easily possible. However, errors can easily occur during automatic extraction. In addition, the automatically extracted reactions often lack accuracy and information about the chemical context. What both expert- and data-driven approaches have in common is that they are highly dependent on the information provided, which means that the performance of the resulting software is only as good as the data it is based on. [53]

Modern machine learning approaches for retrosynthesis prediction are usually categorized as data-driven approaches. They again can be sorted into two categories: template-based and template-free algorithms. Template-based retrosynthetic prediction [21, 74–76] methods rely on predefined reaction templates or rules derived from known chemical reactions. Typically, these are extracted automatically to meet the extensive data requirements of machine learning models, though manually codified rules are sometimes integrated as well. In contrast, template-free retrosynthetic prediction tools [77–80] predict reaction pathways without relying on specific templates. instead, retrosynthesis is defined as a sequence generation problem. These sequences can be SMILES strings or molecular edit actions on a molecular graph. [54]

1.2.4.1 Single-Step vs Multi-Step Retrosynthesis

In single-step retrosynthetic analysis, two main challenges exist: determining the reaction center of the product structure and generating valid reactants based on the identification of the reaction center. The difficulty in determining the reaction center lies in the fact that there are several ways to decompose a molecule, not all of which can lead to an optimal synthetic pathway. When generating valid reactants and reagents, the chemical context and the feasibility of the reaction must be ensured. [55]

Since the number of molecules that can be synthesized in a single step is limited, multi-step retrosynthetic analysis is required. Multi-step retrosynthesis algorithms must build a directed acyclic graph that starts with the target molecule and navigates to simple building blocks, usually commercially available ones. These search algorithms often repeatedly call single-step retrosynthesis methods until suitable starting materials are found. Solving the complex problem of predicting multi-step retrosyntheses, presents several challenges: The exponentially large search space for possible retrosynthetic pathways, different criteria for a good synthetic route depending on the chemists' point of view or chemical scenarios and a lack of reliable retrosynthetic route data sets. [54, 55]

1.2.5 Software

Recent advances in machine learning and artificial intelligence have led to the development of more advanced models for predicting synthetic pathways for novel compounds and general CASP-related tasks. Both open-source software and commercially available tools are established. The open source frameworks in particular have made considerable efforts to improve the accessibility of synthesis planning models. Some relevant open-source packages are presented below, as well as an overview of available closed-source CASP programs.

1.2.5.1 ASKCOS

Coley et al. [81] presented in 2019 an open-source framework called ASKCOS for CASP-related tasks. Its core module, automated multistep retrosynthesis, is includes a template-based single-step solver and a root-parallelized MCTS. The initial implementation was similar to Segler et al. [74]. Reaction templates are automatically extracted from the Reaxys database [29] and the USPTO. The building blocks are a database of buyable chemicals available from eMolecules [82] or Simga-Aldrich [83]. As additional features, ASKCOS includes software for reaction condition prediction, product prediction, atom mapping, synthetic complexity evaluation, and a chemical lookup including commercially available structures. In contrast to AiZynthFinder, ASKCOS provides a graphical web interface. [81]

1.2.5.2 AiZynthFinder

AiZynthFinder [19] was first published in 2020 as an open-source retrosynthetic analysis software. The tool consists of a single-step, template-based model, which utilizes predefined reaction templates, and a Monte Carlo Tree Search (MCTS) algorithm. The search algorithm performs the breakdown of the target molecules into suitable building blocks, guided by a neural network prioritizing reaction templates. The initial algorithm is a reimplementation of the approach presented by Segler et al. [74]. The used reaction template set is extracted from the US Patent and Trademark Office (USPTO) [23]. As acceptable building blocks, a subset of compounds of the ZINC database are provided [84]. The authors created a robust and transparent tool with simple usage requirements. The software is open-source and includes nowadays various implementations of multi-step search algorithms, including depth-first proof-number search (DFPN) and A* algorithms, with the aim of expansion. [19, 85]

1.2.5.3 Closed-sourced Retrosynthesis Prediction Tools

There is an increasing number of commercial CASP tools, featuring automatic retrosynthetic analysis, but requiring paid licenses. The underlying algorithms, (hand-coded) reaction templates, or databases are usually not accessible. Examples are Synthia [73] from Merck, ICSynth [86] from DeepMatter, synthesis planning [29] from Reaxys, SciFinder^n [30] from CAS, RXN [87] from IBM, Spaya [88] from Iktos and Chemical.AI's ChemAIRS [89].

1.2.6 Benchmarking Multi-step Retrosynthesis

Comparing the results of retrosynthetic analysis algorithms and thereby synthetic routes is a complex and difficult task. First, the general problem of accessing the quality of synthetic routes is discussed. Followed, by an introduction to benchmark approaches for machine learning-based multi-step retrosynthesis tools.

The effectiveness of a retrosynthetic analysis is often subjective and depends on the chemist's judgment. What one chemist considers a reasonable or efficient synthetic route may differ from another chemist's opinion. Synthetic routes that are feasible in one laboratory may not be successful in another. To make matters worse, there are often multiple synthetic routes for a target, the success of which depends on various conditions. And even if no synthetic route can be generated for a target structure, this does not mean that there is none that just needs to be found. An objective ranking of synthetic routes is not trivial. Is a synthetic route with more but simple reaction steps worse than one with fewer but difficult reactions? What is a simple reaction and what is a difficult reaction? Are transformations from one structure to another better than reactions with several reactants? Does the route include chemically unreasonable steps, and what is chemically unreasonable? [49]

To date, there are no satisfactory objective answers to these questions. The quality of a synthetic route always depends on the circumstances of its application. This subjectivity makes it difficult to establish universally valid criteria for theoretical evaluation since the only proof for the success and quality of a synthetic pathway is its experimental validation. Unfortunately, this type of verification is time-consuming and expensive, which is why this method is rarely chosen. [90, 91]

Segler et al. [74] presented a data-driven, template-based retrosynthetic analysis approach in 2018. Using an MCTS algorithm, an expansion policy network that guides the search, and a filter network to pre-select the most promising retrosynthetic steps, synthetic routes for molecules are predicted. With the publication, they also presented for the first time a performance evaluation for a machine learning retrosynthesis prediction software, similar to a Turing test: The effectiveness of the retrosynthetic planner presented was evaluated in a preference test with 45 organic chemists. Nine synthetic routes taken from the literature were compared with nine routes predicted by the retrosynthesis model for the same targets. Chemists were asked to choose their preferred route based on feasibility and personal preference. No statistically significant preference was found in selecting a route from either category, suggesting that automatically generated routes can achieve the same quality as human-written ones [74]. Mikulak-Klucznik et al. [92] conducted a similar study in 2020, with the same result that chemists are not able to distinguish between the paths predicted by tools and experimentally validated paths. Although this intuitive approach is an important evaluation method for the adoption and acceptance of synthetic routes proposed by machine learning algorithms, it is still too time-consuming for large-scale experiments. [90]

In 2022, Genheden and Bjerrum [91] presented the PaRoutes framework for automated benchmarking of retrosynthesis route predictions. They present a pipeline for the evaluation of multi-step retrosynthesis methods. In addition to proposing new metrics for quality assessment, they also perform an evaluation of established machine learning algorithms based on their implementations. The focus is particularly on comparing the underlying algorithms of prediction tools independently of test and training data. For this purpose, training and test data are provided and a retraining of the models is proposed. The pipeline consists of three steps: preparing the models, e.g. training the given test data, solving the given prediction tasks, and comparing the results using the proposed metrics. These include the average search time to reach convergence in the number of solved targets, the number of solved targets, the top-N accuracies, and a diversity metric based on route clusters. The top-N accuracies are calculated based on the sorting of routes by length and presence of source materials, followed by the calculation of tree-edit distances between the predicted and reference routes. The available training and test set including the reference routes is extracted from the USPTO. [91]

Maziarz et al. [90] proposed a similar platform called Syntheseus in 2023. It supports the benchmarking of single- and multi-step retrosynthesis prediction methods and provides a set of best practice advice. The focus of the benchmarks is on the evaluation of models as they would be used in the final CASP use. In addition, their evaluation methods are intended to be incorporated more fully into the development process of retrosynthesis prediction tools. They provide means to automatically wrap metrics around each component of a retrosynthesis prediction tool so that, for example, a developer changing some parameters in the search algorithm gets direct feedback on the performance changes of the individual component and the overall model. Nevertheless, the authors emphasize that experimental validation or quality assessment by chemists is the most appropriate method to evaluate retrosynthesis prediction methods. [90]

1.2.7 Modification of Synthetic Pathways

Modifying (retro-)synthetic pathways is part of the daily work of synthetic chemists, developing more efficient and practical ways to synthesize molecules or adapting pathways for the synthesis of similar target structures. The modification can optimize reaction conditions, reduce the number of steps or improve the overall yield. This process not only improves the practicability of synthetic routes, but also facilitates the discovery of new compounds. [93]

As already described in previous sections (see sections 1.2.4 and 1.2.5), methods that can generate synthetic pathways for novel compounds have been under development for years. However, there are only a few tools that support the guided modification of synthetic routes. Especially nowadays, when numerous synthetic routes can be created in minutes, methods for adapting routes to individual needs are required.

Linked Chemical Information (LinChemIn) [94] is a toolkit designed for managing, analyzing, and modifying chemical reaction networks and synthetic pathways, with a program interface that integrates several CASP tools. All operations are based on the data structure, SynGraph, a directed graph-based class, saving the connectivity between reactions and chemicals. SynGraph supports various operations essential for synthetic chemistry, including merging, extraction, and comparison. Merging is the combination of multiple SynGraph instances to form larger synthetic trees or catalogs of routes. Extraction is used to isolate specific subgraphs to identify distinct synthetic routes and by comparing SynGraphs the structural equality and similarity between different synthetic routes using graph-based metrics can be accessed. In 2024 Pasquini et al. [95] published an extension of the software package, to support route arithmetic operations. This includes the modification of synthetic routes by adding or removing chemical reaction nodes, while chemical consistency is ensured. [94, 95]

1.3 Synthetic Accessibility of Lead Structures

Lead structures usually exhibit a certain degree of biological activity and serve as a starting point for further structural optimization processes. Structural analogs are chemically similar to the lead or target structures, but have slight structural changes. These make it possible to investigate the structure-activity relationship (SAR) and optimize properties such as efficacy, selectivity and pharmacokinetic characteristics. Lead structures and their repeatedly adapted structural analogues can go through several rounds of the DMTA (design-make-test-analyze) cycle before they fulfill all the desired requirements. It is therefore important that the structures in question can be synthesized efficiently despite their structural modifications. By integrating synthetic accessibility assessments during lead modification, lead compounds that not only exhibit desirable pharmacological profiles but are also suitable for efficient synthesis can be prioritized. In the following, a selection of algorithmic approaches and concepts, which include the question of synthesizability in the generation process of structural analogues are presented. The thematic division of the methods was partly adapted from Levin et al. [96].

1.3.1 Combinatorial Chemistry

Already in 1993 Gallop et al. [97] described a technique, called combinatorial chemistry, *"as the systematic and repetitive, covalent connection of a set of different "building blocks" of varying structures to each other to yield a large array of diverse molecular entities."* [97]. In other words, combining a limited number of building blocks in all possible ways results in the creation of a larger library of new and more diverse compounds. Fragment spaces are an example of an approach based on the principles of combinatorial chemistry. Fragment spaces consist of collections of molecular fragments and the corresponding connection rules that dictate how these fragments can be combined to form new compounds. This approach ensures that the generated compounds are synthetically accessible by using established chemical reactions and readily available building blocks. Early methods generated fragment spaces based on breaking into retrosynthetically interesting chemical substructures (BRICS) [27], where the generated fragments can then be recombined into target structures. Recently, the concept of fragment spaces has gained popularity, not only due to ’make-on-demand’ spaces that provide billions of molecules that can be easily synthesized and shipped by the manufacturer, but also due to their space-saving way of storing this number of possible molecules [98]. However, special algorithms [99–102, E1] are required to work with the architecture of fragment spaces and to enable virtual screening without resorting to full enumeration. Examples of commercially available fragment spaces are the REAL Space [103] from Enamine Ltd. or the GalaXi Space [104] from WuXi LabNetwork.

Another example is the RECAP (Retrosynthetic Combinatorial Analysis Procedure) algorithm, that was introduced in 1998 by Lewell et al. [105]. RECAP can generate sets of building blocks based on biologically active molecules, which can be used as the basis for the synthesis of novel biological motifs in combinatorial chemistry approaches. Starting with a set of structures with desired activity values, a set of rules is applied to generate building blocks. These building blocks are then further analyzed to determine the most frequent fragments and patterns. The set of fragmentation rules is based on eleven chemical bond types from common chemical reactions. Therefore, the generated building blocks should be synthesizable by common chemistry and recombined to synthetic accessible novel compounds. [105]

1.3.2 Fragment-Based Enumeration Techniques

Targeted fragment-based enumeration techniques [106, 107] are established tools to systematically generate libraries of structural analogues. Starting from a strategic fragmentation of a target molecule, the fragments generated are assembled into new, synthetically accessible compounds using specific reaction rules.

Two recently published examples are Renate [108] and MegaSyn [109]. Renate, builds pseudo-retrosynthetic routes for a reference ligand, which leads to novel structural analogues. The algorithm consists of four steps: first the reference ligand is fragmented, second a search is performed to find the most similar building blocks to the generated fragments, third a novel structure is generated based on the identified building blocks and reaction vectors, fourth the generated structure is scored. The fragmentation of the ligand structure is based on the BRICS [27] algorithm. [108]

MegaSyn [109] is a suite of automated tools for molecular design and lead optimization and includes three main components: a SMILES-based generative model with recurrent neural network utilization, an analogue generation software and a retrosynthetic and fragment analysis to score synthetic feasibility. To generate structural analogues based on a target molecule the following steps are performed: Starting with an initial model, which is trained on data extracted from ChEMBL, the model is primed to work with the individual target structure. For this, the target structure is fragmented using RECAP [105] rules, and the initial model is trained only on the generated substructures. With the primed models novel structures are generated and ranked by score. The top 10% of the ranked molecules are fed back into the models for further training and optimization. This process is iterated multiple times. The final pool of structures is used for a lead expansion step, performed by a Pipeline Pilot [110] protocol. Here, structural analogues are generated through bioisosteric replacements and applied transformations, while being evaluated for undesirable functional groups and synthetic feasibility. [109]

1.3.3 Synthetic Pathway-Based Enumeration Techniques

Synthetic pathway-based enumeration techniques use reaction-based approaches to explicitly promote the synthesizability of the generated molecules. Structural analogues are generated by running different combinations of building blocks through an initial synthesis plan. This ensures that all enumerated molecules can theoretically be synthesized using the same sequence of chemical reactions. This technique was chosen for [D2].

Another recently published example is the EASIE, Exploration of chemical Analog Space, Implicitly and Explicitly, approach by Levin et al. [96]. EASIE is a method to generate synthetically accessible analogues for focused library expansion. The workflow consists of three parts: Evaluation and selection of a suitable synthetic route, prediction of the distributions of the properties in an enumerated space based on the route, constraining the parameters based on the prediction, and enumeration of the resulting structural analogues. The focus lies on estimating 'diversifiability' within the number of generatable structural analogues based on the route to support an implicit enumeration framework for computational efficiency. Diversifiability is calculated as the number of possible combinations of building blocks compatible with the synthetic route. The actual enumeration process follows the same algorithm as described in [D2]. The diversifiability metric can be used as an additional metric when comparing CASP route proposals, as it is computationally cheap, making the size of accessible structural analogue space a parameter for selecting suitable routes. In addition, specific properties, like molecular weight, topological polar surface area (TPSA), and LogP of the structural analogue space can be predicted based on available building blocks. Levin et al. rely on an additive approach concerning the building blocks, where summing these properties with a correction factor closely approximates the properties of the final molecule. This allows for efficient property distribution estimation using convolution of probability density functions. [96]

PathFinder [111] combines fragment-based with synthetic pathway-based enumeration techniques. The method was introduced in 2019 by Konze et al. [111] and generates novel compounds in synthetically accessible chemical space. Starting with a lead molecule PathFinder creates a 'saturated' retrosynthetic tree by applying all possible retrosynthetic reactions from a database recursively. The process stops when a user-defined depth of the tree is achieved. In the tree, each chemical structure node is connected to all possible reactions that can make the structure in one step. By following all possible paths in the 'saturated' retrosynthetic tree a set of synthesis trees is generated for the lead molecule, where each chemical node is only followed by one reaction node. From this set of synthesis trees, the most promising route(s) are selected to start an enumeration. Based on a curated library of building blocks, extracted from eMolecules and possibly enriched by the user, the enumeration tool generates a library of structural analogues by simulating the chosen synthetic route. [111]

1.3.4 Iterative Approaches

In contrast to fragment-based enumeration methods, iterative approaches [112–115] for generating synthetically accessible structural analogues do not generate fragments or building blocks themselves. They start with a given set of building blocks and a series of reactions and create a synthetic pathway in forward manner. This explores all achievable structures based on the starting data.

Popular examples are Synopsis by Vinkers et al. [116] or Barking up the right tree by Bradshaw et al. [117]. Synopsis starts with a database of available building blocks and a set of generic reactions, and simulates synthesis steps to generate molecules by using a genetic algorithm. A user-definable fitness function guides and evaluates the design process, leading to the optimization of desired properties in the molecules. [116]

Bradshaw et al. [117] combine synthetic routes with a deep generative model to search for synthetically available, novel molecules with desired properties. Synthetic routes are presented as directed, acyclic graphs (DAGs) where building blocks are recursively combined via reactions to form more complex structures. The deep generative model is trained to output novel molecules together with a synthesis DAG. Building a synthesis DAG is divided into three actions: the addition of nodes (building blocks or products), the specification of the molecular identity of building block nodes, and the choice of connectivity between reactant and product nodes. A probabilistic model is used to parameterize probability distributions over each action, where each action is predicted as a function of the previous actions. A joint recurrent neural network computes a context vector, which is then used by feed-forward action networks to predict each action. [117]

1.4 Motivation and Thesis Structure

With the provided information about existing methods and approaches the following limitations regarding three different aspects of synthetic accessibility of virtual designed drug candidates were identified:

The most important information for synthetic accessibility calculations are chemical reactions. These are usually in SMILES or SMARTS format. The latter in particular can quickly become difficult for the human eye to interpret and read as more details are described. A simple solution is to visualize the reactions in the form of structure diagrams, including all available additional or pattern information. Existing methods struggle with visualizing SMARTS-specific information such as logical operations or recursive chemical environment descriptions.

The creation of synthetically accessible structural analogues based on a target structure or from scratch is a topic that has been addressed using various methods. In general, new compounds are generated along with a synthetic pathway based on a set of reaction rules. In some cases, additional information about the structural properties can be specified or the generated compounds can be ranked. Common to all is that the synthetic route is automatically constructed, and at most a set of reaction rules can be given to calculate the route. Often, however, the synthetic chemist already has an idea or an established way of synthesizing the scaffold of a lead structure. At the time of the first publication of this thesis, there was no method for generating structural analogues based on predefined synthetic routes. Incorporating existing synthetic routes makes it easier to respond to individual laboratory conditions or the preference of synthetic chemists and makes the synthesis of sets of structural analogs very efficient.

It is not only the modification of lead structures that can be of interest, but also the modification of their synthetic routes, especially in times when CASP tools offer the possibility to quickly calculate routes for arbitrary compounds. So far, the resulting routes can often only be used as a source of inspiration and have to be tailored to suit individual ideas. The synthetic chemist lacks a guided method to adapt all components of a synthetic route, be it individual reactants, reactions or the target structure itself, to their own needs and to be presented with suitable alternatives.

In summary, this thesis focuses on providing algorithmic approaches and software solutions for the following research objectives:

1. Support research related to generic reaction patterns written in the Reaction SMILES, Reaction SMARTS, or SMIRKS languages by providing simple means to understand, interpret, and analyze these patterns.

2. Enable synthesis-aware lead structure modification and the creation of structural analogue spaces based on common synthetic pathways for efficient synthesis efforts adapted to individual circumstances.

3. Involve the richest source of expertise, namely the knowledge of chemists, in the modification and design process not only of lead structures but also of their synthetic pathways.

The three publications of this cumulative thesis are presented in the following chapters. The publications are grouped according to research topics and therefore do not follow the order of publication. First, Chapter 2 describes [D1], a method for the visualization of generic reaction patterns. Second, in Chapter 3 the publication [D2], an algorithm for lead structure modification based on given synthetic routes is discussed. Thirdly, in Chapter 4 the work of [D3] extending the previous algorithm so that not only the lead structure but also the entire synthetic route can be modified is presented. Each chapter contains a further discussion and positioning as well as a comparison of the method with the approaches presented in this chapter. The last chapter is a general conclusion of this thesis.

Chapter 2

Visualizing Generic Reaction Patterns

Chemical reactions pose a major challenge when it comes to processing and analyzing them in a computer-readable format. As explained in Section 1.1.2, especially generic chemical reaction patterns are essential for modern CASP tools. These can either be written by hand, which is labor intensive and requires considerable expertise, or obtained by extracting experimentally validated reactions and translating them from the existing literature using (semi-)automatic methods [118, 119]. However, this often still requires manual supervision by a synthetic chemist to ensure accuracy and correctness. The Reaction SMARTS or SMIRKS language is the community standard for formulating generic reaction patterns that are accessible and interpretable by computers, but in human-readable text form. However, even experts sometimes struggle to read or write these patterns due to their complexity, which hinders the development of much needed generic reaction patterns. Taking advantage of the computer readability of the reaction languages SMARTS and SMIRKS, a graphical language for automated visualization of generic reaction patterns was developed, resulting in the publication [D1]. The following chapter summarizes the underlying algorithm and discusses the results.

2.1 Methodical Summary

The algorithm developed during this work resulted in a software application called ReactionViewer. The ReactionViewer was developed as a means of visualizing generic reaction patterns, which are processed and utilized in the course of this thesis. As already mentioned in Section 1.1.4, there are few existing methods for the visualization of chemical reactions and none that specialize in the underlying SMARTS language and its complexity. The concept of the ReactionViewer was derived from an existing approach named SMARTSviewer by Schomburg et al. [43]. The SMARTSviewer can convert single SMARTS and thus SMILES expressions into graphical representations. See Chapter B for details on the graphical design choices and implementation of SMARTSviewer and ReactionViewer.

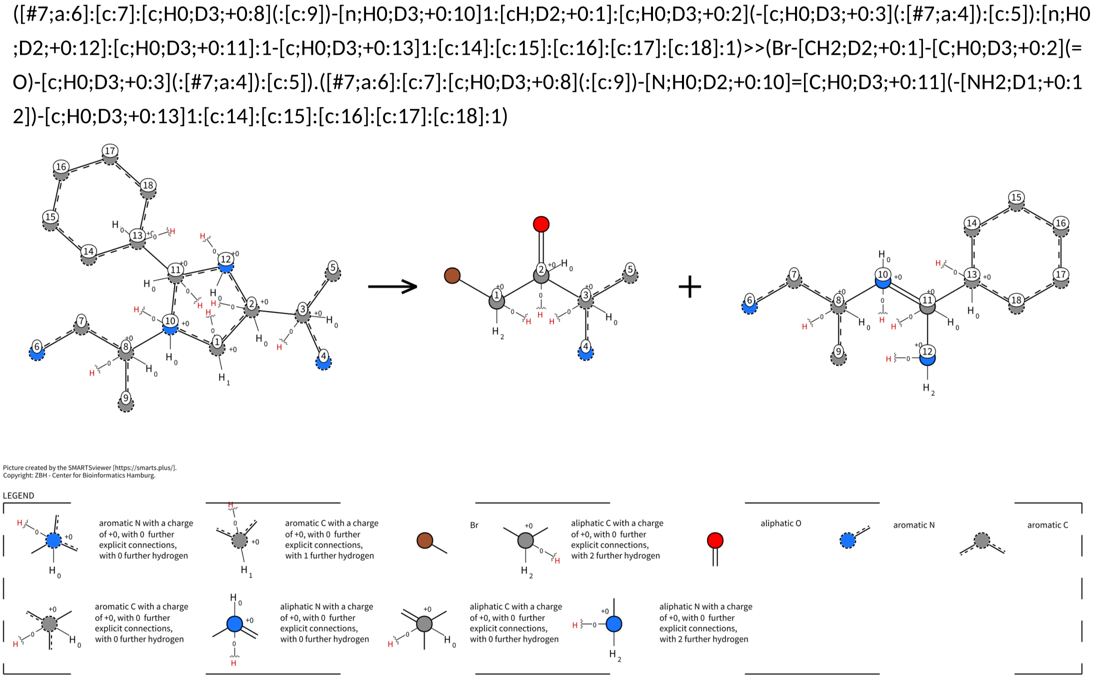
2.1.1 Visualizing Chemical Reaction Patterns

The ReactionViewer breaks down the given reaction pattern into independent SMARTS expressions and bases their visualization on the existing method, the SMARTSViewer by Schomburg et al. [43]. Figure 2.1 displays a visualization of a N-containing heterocycle formation reaction pattern [19] written in retrosynthetic form, created using the following algorithm: The symbol '≫', which separates the reagent patterns from the product patterns, is replaced by a dot in an initial parsing phase. This converts the reaction pattern into multiple disconnected SMARTS patterns, each of which can be interpreted independently. Each SMARTS expression in the unconnected pattern is then converted into a tree-like data structure called a SMARTS graph, which represents the semantics of the pattern. The algorithm not only stores all relevant information extracted by modeling the language as context-free grammar but also the position of the last pattern before the '≫' symbol to store the transition from the reactant to the product pattern. Each generated SMARTS graph is checked for validity and simplified, with redundant information removed if necessary. [D1]

The algorithm creates a legend for each SMARTS graph describing the used symbols. The legend is shorted to remove multiple occurrences of the same symbol originating from different patterns. For each SMARTS graph, a graphical representation is generated [120] and arranged in a row, aligned with the geometric middle of the largest compound. Reaction symbols (plus and arrow, following the IUPAC's "Compendium of Chemical Terminology" definition [121]) are inserted between the layouts of the SMARTS graphs. The information about the position of the last reactant is used to place these reaction symbols correctly between the SMARTS graphs. [D1]

2.1.2 Usage of the Software

The described algorithm is integrated into the graphical user interface of the SMARTS-viewer tool, resulting in the software application ReactionViewer. This allows not only single SMILES and SMARTS patterns to be visualized, but also Reaction SMILES, Reaction SMARTS, and SMIRKS patterns. ReactionViewer can visualize not only individual patterns, but also complete reaction data sets, with their visualization being



([#7;a:6]:[c:7]:[c;H0;D3;+0:8](:[c:9])-[n;H0;D3;+0:10]1:[cH;D2;+0:1]:[c;H0;D3;+0:2](-[c;H0;D3;+0:3](:[#7;a:4]):[c:5]):[n;H0;D2;+0:12]:[c;H0;D3;+0:11]:1-[c;H0;D3;+0:13]1:[c:14]:[c:15]:[c:16]:[c:17]:[c:18]:1)>>(Br-[CH2;D2;+0:1]-[C;H0;D3;+0:2](=O)-[c;H0;D3;+0:3](:[#7;a:4]):[c:5]).([#7;a:6]:[c:7]:[c;H0;D3;+0:8](:[c:9])-[N;H0;D2;+0:10]=[C;H0;D3;+0:11](-[NH2;D1;+0:12])-[c;H0;D3;+0:13]1:[c:14]:[c:15]:[c:16]:[c:17]:[c:18]:1)

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
aromatic N with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aromatic C with a charge of +0, with 0 further explicit connections, with 1 further hydrogen
Br
aliphatic C with a charge of +0, with 0 further explicit connections, with 2 further hydrogen
aliphatic O
aromatic N
aromatic C
aromatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic N with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic N with a charge of +0, with 0 further explicit connections, with 2 further hydrogen

Figure 2.1: Visualization of a N-containing heterocycle formation reaction pattern created using the ReactionViewer tool. The figure includes the representation of atom mapping and a legend. The pattern used for this visualization was adapted from the transformation data of AiZynthFinder [19]. The image is extracted from [D1].

saved as a PDF file. Visualization of individual reaction patterns can also be exported as SVG or PNG files. Furthermore, the algorithm has been integrated into the visualization process of the SMARTS.plus server [47]. This allows the ReactionViewer to be used free of charge by the public via the server's web interface at `https://smarts.plus/`. The server provides the option to visually compare two generic reaction pattern, although the available comparison algorithm for single SMARTS pattern of Schmidt et al. [122] cannot yet be used with reaction patterns. [D1]

2.2 Validation

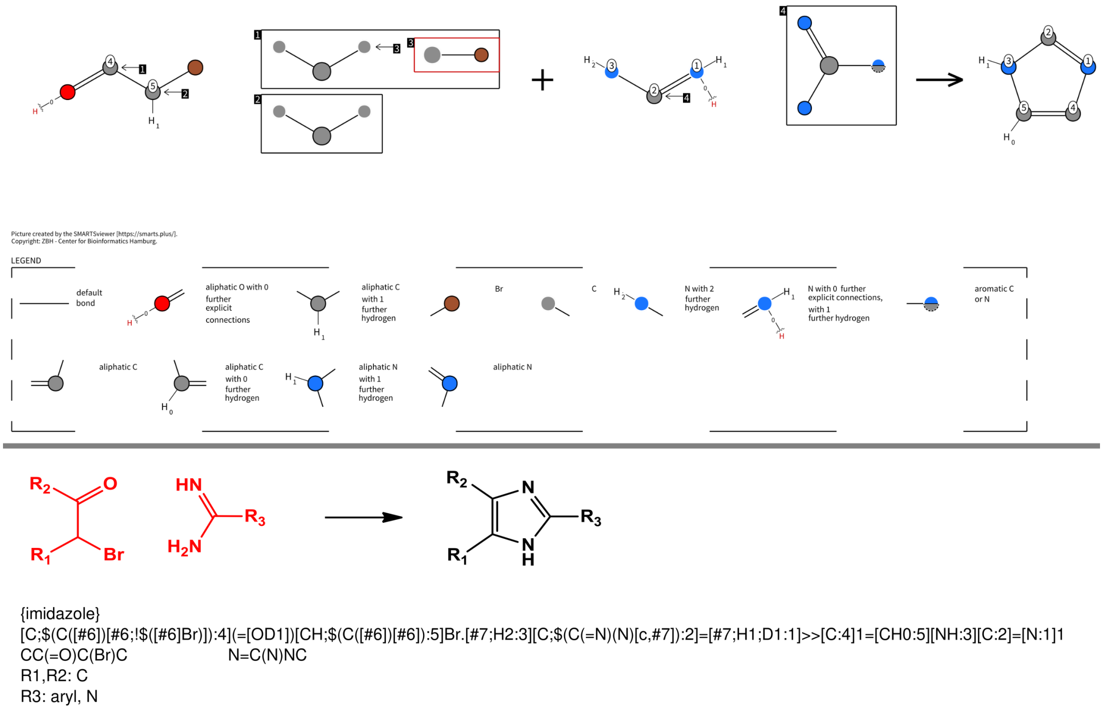
Validating a visualization algorithm is not standardized and therefore can be difficult. For the ReactionViewer publication [D1] two experiments were performed to present the performance and utility of the algorithm based on two datasets: The first containing 46695 generic reaction patterns, provided by the open-source retrosynthetic planning software called AiZynthFinder [19] (see Section 1.2.5.2 for further information). The second dataset includes 58 reaction patterns, together with a visualization, provided by Hartenfeller et al. [26].

The visualization routine of the ReactionViewer was effectively applied to all reaction patterns in both datasets, despite the wide variety of reaction types present. The visualized datasets can be found in the Supporting Information of [D1] or downloaded from [123]. Figure 2.1 shows one example generic reaction pattern, extracted from the AiZynthFinder dataset, visualized by the ReactionViewer. The images generated from the datasets provide a quick overview of general information such as the number of reactants and products and their overall structures. They further allow the simple display of detailed information such as the valence or charge of a particular atom. Both can also be read from the string, but require significantly more time and skill. [D1]

The dataset provided by Hartenfeller et al. [26] was used not only to present a generic reaction with a recursive pattern, but also to compare the designed graphical representation with another approach: Hartenfeller et al. present their own schematic representations of the 58 chemical reactions created. It should be noted that the motivation for the publication by Hartenfeller et al. was not to visualise the reaction patterns, but to generate and evaluate the Reaction SMARTS pattern. The visualization provided is only intended to show the features of the ReactionViewer visualization in comparison to a common visualization technique. Three example reactions were chosen, a Suzuki coupling, a carboxylic acid or ester reaction of benzimidazole derivatives and an Imidazole synthesis, to discuss the differences in the visualization approaches. All three visualized examples can be found in the publication [D1]. The visualizations of the full data sets can be found in the Supporting Information of [D1]. Figure 2.2 shows an example from the comparative study, the visualization of the Imidazole synthesis.

The visualization provided by Hartenfeller et al. employs explicit elements or letter abbreviations to represent atoms within a chemical reaction. A two-color scheme is used to visualize the reaction center, information that is not given by the Reaction SMARTS pattern and must be the result of human interpretation. The graphical representation lacks an atom mapping visualization, which can be important in understanding atom rearrangements. Additionally, the visualization of recursive patterns in Hartenfeller et al. is irregular and influenced by human interpretation, which can lead to variations in how patterns are presented. Neither are logical operations directly visualized. [D1]

In contrast, ReactionViewer fully translates the given pattern with comprehensive explanations, ensuring that all details are included in the visual representation. It


LEGEND
default bond
aliphatic O with 0 further explicit connections
aliphatic C with 1 further hydrogen
Br
C
N with 2 further hydrogen
N with 0 further explicit connections, with 1 further hydrogen
aromatic C or N
aliphatic C
aliphatic C with 0 further hydrogen
aliphatic N with 1 further hydrogen
aliphatic N
R2
O
R1
Br
HN
R3
H2N
N
NH

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.

```
{imidazole}
[C;$(C([#6])[#6;!$([#6]Br)]):4](=[OD1])[CH;$(C([#6])[#6]):5]Br.[#7;H2:3][C;$(C(=N)(N)[c,#7]):2]=[#7;H1;D1:1]>>[C:4]1=[CH0:5][NH:3][C:2]=[N:1]1
CC(=O)C(Br)C                    N=C(N)NC
R1,R2: C
R3: aryl, N
```

Figure 2.2: Two graphical representations of an Imidazole synthesis. The upper image shows the visualization generated by the ReactionViewer. Below is the visualization provided by Hartenfeller et al. [26]. The image is extracted from [D1].

explicitly visualizes recursive patterns by presenting them as independent molecular graphs, making them easier to understand. ReactionViewer is customization, allowing users to adapt the visualization to their specific needs. For example, users can choose to use a color scheme (default) or element symbols to represent atoms. Visualizations are generated automatically, resulting in the translation of as much or as little information as specified in the pattern. [D1]

In conclusion, although both the images provided by Hartenfeller et al. and those generated by ReactionViewer serve the purpose of visualizing chemical reactions, they differ significantly in their characteristics and interpretability. The visualization provided by Hartenfeller et al. is influenced by human interpretation of the given pattern. Although this may add further information, it may also lead to inconsistencies in the visual representations. ReactionViewer offers an automated approach that focuses on a one-to-one translation of the reaction pattern, explaining every detail. [D1]

2.3 Discussion

To the best of current knowledge, ReactionViewer is the first software to provide an algorithm specializing in the visualization of generic reaction patterns. While there is plenty of software available for visualizing molecules, software for visualizing molecular patterns is more sparse, and only a few can handle reaction patterns. In Section 1.1.4, three relevant examples are presented: RDKit, MarvinSketch, and the SMARTSviewer. The first two are software solutions that mainly provide and focus on utilities for visualizing structural data, not molecular or reaction patterns; although both are able to do the latter. The SMARTSviewer and the ReactionViewer on the other hand specialize explicitly in the visualization of chemical patterns. This difference in focus is obvious at first glance: both RDKit and MarvinSketch provide easily readable structural diagrams explicitly naming heavy atoms with letter abbreviations and following the guidelines of the IUPAC recommendations [8] (compare Figure 1.2 and 1.4 left side).

In contrast, the SMARTSviewer visualizes with a higher level of abstraction even for simple SMILES strings (see Figure 1.6 left), where atoms are displayed as circles and distinguished by color and the legend provided. This may be unintuitive when looking at fully defined molecular structures, where the structural diagrams provided by MarvinSketch and RDKit meet the expectations of chemists. However, molecular patterns usually include more and diverse information than a simple structure written in SMILES. For example, the SMARTS pattern describing a thiazene, contains information about a substituent that can be either a fluorine, chlorine, bromine, iodine or a nitrogen with a charge of +1 and two aromatic bonds. Both RDKit and MarvinSketch have difficulty displaying the amount of information given by the SMARTS pattern. The former displays only a placeholder at the atom position (see Figure 1.2 right), while the latter displays the information as a simple string next to a placeholder (see Figure 1.4 right). Both variants are difficult to read and interpret. The SMARTSviewer (see Figure 1.6, right), on the other hand, provides not only a detailed visualization of the information (e.g. a circle divided into four colors to show the four possible elements), but also a legend with additional explanations of the visualized components.

Please note again that the visualization of SMARTS patterns was previously developed by Schomburg et al. [43]. The above comparisons are only made because the described differences in the level of representation and explanation are also reflected in the visualization of the reaction patterns: Both RDKit and MarvinSketch can visualize the simple Reaction SMILES string without leaving out any information (compare

Figures 1.3 and 1.5, top). However, it should be mentioned that MarvinSketch adds information, especially hydrogens, which are not specified in the string.The motivation is probably to get the chemistry right, but this complicates interpretation and the ability to check the correctness of the pattern itself (see Figure 1.5 top: Esterification second product, single oxygen defined, but dihydrogen oxide is shown). Looking at the Reaction SMARTS pattern, the situation is the same as for the SMARTS pattern. RDKit does not display all the information given in the pattern and uses placeholders for non-translatable information (compare Figure 1.3, bottom). MarvinSketch displays all the given information, but most of it is in the form of strings attached to the atoms, defeating the intention to provide easily understandable information (see 1.5, below).

Both MarvinSketch and RDKit visualize the given atom mapping, which is particularly important for reaction patterns, as this indicates the reaction center and provides information on how the atoms are rearranged. ReactionViewer contains all the given information, including the atom mapping, both in visualized form and with additional explanations. It even shows in a clear and ordered form the different options for and additional information for each atom given in the recursion notation of the SMARTS language. This accurately visualizes the described chemical environment of the atom, including a color code for the exclusion of presence of neighboring groups (compare Figure 1.7, bottom).

In summary, the main advantage of the ReactionViewer is the focus on clarity and precision in visualizing reaction patterns, which can be especially beneficial for chemists who need to interpret or debug complex definitions. Unlike MarvinSketch and RDKit, SMARTSviewer and ReactionViewer are not as versatile for displaying molecular structures, but excel at their purpose: visualizing chemical pattern data.

2.4 Current Limitations and Further Directions

At the time of publication, ReactionViewer lacked support for handling reaction patterns containing agent structures. DayLight defines agents as molecules that neither provide atoms to the product nor receive atoms from the reactants [44]. They are often used as catalysts, solvents, or other additives that participate indirectly in a reaction. These agent structures are denoted by a single '>' symbol, and the complete pattern follows the structure 'reactant > agent > product'. When attempting to process patterns with agent structures, the application produced an error message. Since then, simple adaptations have been made to the parsing mechanism to accommodate reaction patterns containing agent structures. The original pattern is now converted into the format 'reactant . agent ≫ product', ensuring a conflict-free parsing process by converting the agent to a simple reactant and visualizing it as such. Users are notified of this change in the semantics of the pattern. Future work could explore the graphical design and implementation of a direct visualization of reaction agents, commonly depicted above or below the reaction arrow. This would ensure a more chemical-knowledge-based representation of agent compounds.

ReactionViewer can directly support the synthetic chemist in the interpretation, creation, or correction of reaction patterns. Currently, each component of a given reaction pattern, considered as an independent SMARTS pattern, is checked for semantic or syntactic errors. The entire reaction pattern is only checked to see if it conforms to the correct format ('reactant . reactant ≫ product'). Daylight's SMIRKS language [13] is a restricted version of Reaction SMARTS [12], which defines five additional rules, used to ensure a distinct application [44]. Integrating these rules as an additional check could be extremely useful to ensure that the patterns are written correctly in the SMIRKS language and to distinguish between Reaction SMARTS and SMIRKS patterns(compare Chapter B). The inclusion of these rules in the current implementation is discussed in Chapter B.2.2.

The next step after visualization would be the editing of chemical patterns. An interactive graphical editor for SMARTS already exists [124]. The processing of Reaction SMARTS is not yet possible, but could be a useful application for the community.

In the context of this thesis, the visualization of not only single generic reaction patterns, but of entire series of reactions including the structures of the reactants and resulting products is of interest. In other words, the visualization of complete (retro-)synthetic routes would be useful. This would allow the visual inspection of generated structural analogues together with adapted retrosynthetic routes given by the Synthesia algorithm (see Chapters 3 and 4 or [D2, D3]). To achieve such a visualizaton the ReactionViewer, included as a command line application, or by calling the provided RestAPI of the SMARTS.plus server, needs to be combined with a simple tree traversing algorithm. Reactants and products can either be visualized by the ReactionViewer or by integrating an image generator for molecular structures. The challenge is to adapt the image size to the depth of the given retrosynthetic route. A prototype for the visualisation of retrosynthetic routes has already been created. This is not part of this work, but may lead to a publication in the future.

Chapter 3

Synthesis-Aware Generation of Structural Analogues

The generation of lead candidates that are synthetically accessible is a requirement for their successful transition from virtual to experimental studies. To achieve structural modification without compromising the synthesizability of the modified compound, this work [D2] presents an algorithm for the generation of structural analogs based on a given retrosynthetic route of a starting lead structure. Structural analogs that are still synthetically accessible are generated by effectively enumerating the retrosynthetic route by replacing selected reactant or intermediate structures with suitable substitutes. In the following, the algorithm is summarized and the results are discussed. Details regarding the implementation can be found in Chapter B.

3.1 Methodical Summary

The algorithm developed during this work resulted in a software application called Synthesia. Synthesia takes a target lead structure, together with a synthesis route and a set of suitable building blocks to generate structural analogs. Theoretically, these analogs can be synthesized following the same, given route. Synthesis routes are represented internally as acyclic graph structures, called retrosynthetic trees, including chemical and reaction nodes in child-parent relationships. Structures have to be given in SMILES, reactions in SMIRKS [13]. SMIRKS is a restricted form of Reaction SMARTS [12] with five rules to ensure that the SMIRKS pattern can be interpreted as a reaction graph, allowing atom and bond changes to be derived from it [13]. The rules can be found in B.2.2. For Synthesia, two additional rules have been established that the SMIRKS patterns must fulfill in order to ensure an unambiguous generation of product structures.

For example, no logical operations are permitted for atoms in the SMARTS pattern that corresponds to the product structure. Further details can be found in B.3. In the following, the term SMIRKS pattern is used for generic patterns that follow the additional rules. The reaction pattern is used to create the modified intermediate and finally lead structures. To guarantee a correct transformation from the reactant patterns to new product structures, additional rules were set. They are described in Section B.3.2 where further details regarding the generation of the structures based on the generic pattern are given. [D2]

3.1.1 Modification of Target Structures Utilizing Synthetic Pathways

To change the lead structure at the root of a retrosynthetic tree while maintaining the predetermined architecture of the synthetic pathway, nodes at lower levels are adjusted, affecting chemical structures further up the tree. The main challenge is to control these effects and align them with the desired structural optimization. Single chemical nodes in the tree, either starting materials or intermediate structures, are exchanged to introduce structural change. Given a set of building blocks as potential substitutes, the algorithm performs the following tree-traversing steps:

1. A chemical node for exchange is selected by the synthetic chemist.

2. For each potential substitute, confirm whether the SMARTS pattern of the original reactant, given by the SMIRKS pattern of the parent reaction node, matches with the substitute. If not, move on to the next candidate.

3. Use the complete reaction pattern together with the possible remaining reactants and the substitute candidate to create a modified product compound. A more detailed description of this step can be found in B.3.

4. Replace the next chemical parent node structure with the newly created product structure and start again from step 2, verifying that the newly modified product structure is a suitable reactant for the next reaction.

5. Continue until a modified target structure has been created in the root or the candidate substitute is incompatible with a reaction pattern of the tree and is therefore discarded.

Figure 3.1 shows an example of the steps of the algorithm, performed with a simplified retrosynthetic tree. With the described algorithm, the validity of the modified retrosynthetic route, meaning its viability or applicability, can be confirmed. [D2]

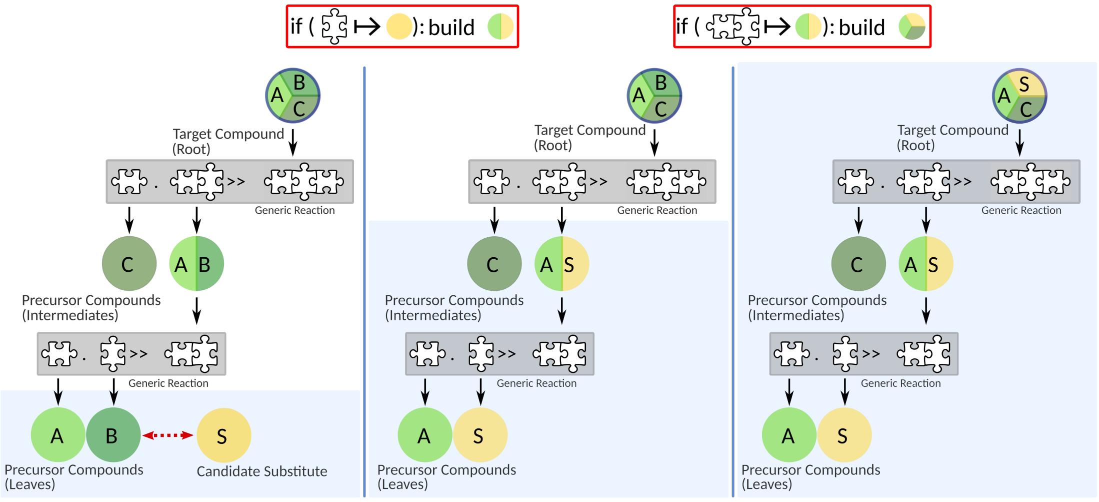


Figure 3.1: Visualization of the changes in a retrosynthetic tree caused by the substitution of a reactant compound (leaf). The three different states of the process (blue rectangles) show how the introduced substitute compound affects the compounds as it moves up the tree. The circles symbolize compound nodes, while the gray rectangles represent reaction schemes. The potential substitute is highlighted in yellow, and the remaining compounds are shown in green. The rectangles outlined in red visualize the first and second steps of the algorithm described above. The image is extracted from [D2].

3.1.2 Structural Constraints

To control modifications for target structure optimization, desired properties can be specified as constraints as a search query, e.g. the topological polar surface area, the LogP value, the molecular weight, or a similarity measure comparing the initial structures with potential substitutes. In addition, SMARTS patterns can be specified to filter out unwanted substructures (e.g. PAINS [45]). Currently, 29 constraint options are available (details can be found in Supporting Information of [D2]). The constraints can be combined, even in subsets such as "a minimum of 3 out of 4 constraints". They can primarily be used to influence the structural properties of the modified target structure and to generate structural analogue spaces with specific feature distributions. However, the constraints can also be used to allow only certain building blocks with desired properties. If constraints have been set, the modified target structures can be scored and ranked together with their retrosynthetic route. For further details see [D2].

3.2 Validation

The evaluation of the presented algorithm for the generation of synthetic accessible structural analogues is challenging for several reasons. A major problem is defining appropriate evaluation criteria that accurately reflect the efficacy and practical utility of the algorithm. Since the synthetic chemist specifies the desired structural properties, evaluations can only be used to test whether structural analogues with the specified properties can be generated. However, the results are based on available building blocks and the quality of the given synthesis route, which is difficult to assess even without applied modifications, as discussed in Section 1.2.6. With the given information the method can only find what is present in the data and can only utilize the given information, which is not unusual for computational methods in drug design, but needs to be kept in mind. While the functionality of the method can be evaluated, it is difficult to rate the performance of the method as "good" or "bad" because there are no quantifiable metrics to accurately measure its quality. To validate Synthesia, the functionality is demonstrated and additional use cases are presented to show the utility of the method.

First, a proof of concept was carried out to verify the integrity of the algorithm. In a further experiment, Synthesia was shown to be able to generate structural analogues with predefined molecular properties while maintaining their theoretical synthetic accessibility. In addition, Synthesia was used to analyze the synthetic compatibility of a series of patent structures to maximize synthetic efficiency.

For all three experiments, the building blocks from Enamine's REAL Space [125] (214,557 structures, in stock in Europe) served as potential substitute candidates. A set of target compounds was generated for testing, comprising 250 structures from DrugBank, selected by dissimilarity, and their 250 most similar structures from REAL Space. In addition, two patent-derived structure series called daurismo [126] and CDK7 [127] were used as test sets. Daurismo, a benzimidazole derivative, is used to treat acute myeloid leukemia. Pyrazolo-triazine derivatives as cyclin-dependent kinase (CDK7) inhibitors are primarily for infectious disease treatment. More details on the datasets can be found at [D2].

Retrosynthetic paths were generated using the open-source software AiZynthFinder [19] with default parameters and a pre-trained model (for more information see Section 1.2.5.2). All generated routes were not further analyzed, so no statement can be made about their quality. In a real-world scenario, Synthesia relies on the ability of the synthetic chemist to select and provide suitable routes. Routes are expected to be field-tested or feasible in in-house laboratories. However, when starting from scratch, machine learning tools such as AiZynthFinder are a good basis for generating initial routes for further modifications. The experiments and results are summarized below.

For detailed descriptions see [D2].

The proof of concept was performed using 100 randomly selected targets with retrosynthetic routes from the described dataset. All reactants that were stored as starting materials in these routes were considered substitute candidates. Synthesia was able to successfully reconstruct all of the original lead structures in the root, showing the method's ability to maintain the integrity of the retrosynthetic route during the search and reconstruction process.

The next experiment evaluated the ability of the algorithm to achieve structural modification goals for different lead structures while maintaining synthetic accessibility. A set of 14 different search query constraints, including more complex constraints such as the rule of five and the rule of three, were applied to 100 randomly selected target compounds from the described dataset. Synthesia was able to generate structural analogues with the desired structural properties based on the given synthetic pathways. The experiment showed that even with restrictive constraints, suitable substitutes could be found, confirming the applicability of the method for different optimisation goals.

The main application scenario for Synthesia is to generate structural diversity for lead structures while keeping the resulting structural analogues synthetically accessible. However, Synthesia can also be used to maximize synthetic efficiency for multiple structures by analyzing compatibility with specific retrosynthetic routes. The strategy of exchanging reactant structures is ideal to maximize common retrosynthetic steps. As proof of concept for this application scenario, an analysis was performed to determine the minimum number and distribution of retrosynthetic pathways required to theoretically synthesize all active structures within a patent series. The CDK7 and daurismo patent series were used as target structures to perform the experiment. For all structures in the patent series, all possible structural analogues were calculated based on the initial retrosynthetic routes, spanning the structure space accessible by the routes. The original patent structures were searched in the generated space and then grouped by retrosynthetic pathway to find the minimum number of clusters. In this way, the groups of structures theoretically synthesizable by the same retrosynthetic route were identified. The cluster analysis of the target structures from the daurismo patent structure is shown in Figure 3.2. The clusters generated with the patent structures of CDK7 can be found in [D2]. The results show that more than one-third of all structures in each patent series could share a common retrosynthetic pathway, while only a few

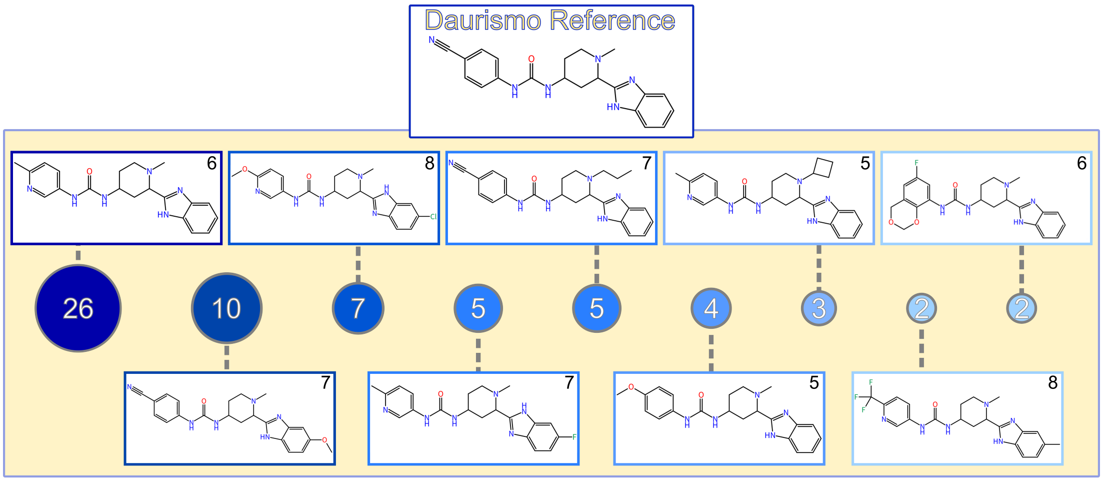


Figure 3.2: Visualization of the results of clustering the daurismo patent series structures according to their compatibility with specific retrosynthetic routes. On top is the original structure of daurismo. Following are the number of structures per cluster, together with the structure the retrosynthetic route was originally calculated for. Singletons are not visualized. The image is extracted from [D2].

structures require unique pathways. This approach of clustering retrosynthetic routes could help synthetic chemists estimate the synthesis effort and select appropriate routes for efficient synthesis of sets of target compounds. [D2]

3.3 Discussion

During the design-make-test-analyze cycle of drug development, synthesizability can be the most costly and time-consuming part, as synthetic pathways must be found and executed for each candidate. With Synthesia, the focus shifts from the design of single pathways for specific molecules to the discovery and utilization of synthetic pathways that enable the production of a variety of analogues. By incorporating the constraints of a synthetic pathway into the design and modification process, libraries of analogues with desired properties and efficient synthetic accessibility can be created.

However, targeted enumeration techniques already exist and are being used to create libraries of synthetically accessible structural analogues. There are fragment-based approaches that build new molecules by fragmenting query molecules into substructures and reassembling them to form novel structures. Incorporating chemical reaction knowledge into the process ensures a degree of synthetic accessibility of the newly created structures. However, the resulting tools (see Section 1.3) are limited by the number of cutting rules available. This limits their ability to fully understand the entire structure of certain molecules or the chemical relationships between reactions, thus restricting the prediction of synthetic accessibility. The example of the software application Renate [108], which uses the BRICS [27] cutting rules created 14 years earlier, or MegaSyn [109], which uses the RECAP [105] rules designed in 1998, further emphasizes the difficulty of creating novel, meaningful and useful rules.

PathFinder [111] is a synthetic pathway-based enumeration technique that combines the described fragmentation approach with a similar retrosynthetic tree enumeration strategy like Synthesia. In contrast to Synthesia, PathFinder designs multiple synthesis routes from scratch before employing different enumeration strategies. Synthesis routes cannot be given by the user and the location of exchange in the synthesis route cannot be specified, both of which negate the opportunities to benefit from the expertise of synthetic chemists.

Iterative approaches, in comparison to the fragmentation of query molecules, start from the opposite direction and generate molecules in a forward enumeration process. Starting from building blocks and chemical reaction rules, novel molecules are gradually constructed and evaluated in several cycles. In the process, a complete reaction tree is built that provides novel structures and a synthetic pathway (compare Section 1.3.4). As with all these tools, the result depends heavily on the input: If the building blocks are biased towards specific types of structures or functional groups, redundant structures may occur, limiting the diversity of the resulting space. In addition, novel molecules are explicitly not based on lead structures, making them more suitable as ideation methods rather than for generating structural analogues. Fragment Spaces follow the same strategy. However, they mostly combine building blocks based on one-step reactions. More complicated combinations of synthesis steps are possible but not yet established, which can limit the level of detail and the complexity of the molecules that can be produced.

All of the approaches presented generate novel synthetic routes together with their final set of molecules. However, automatically constructed synthetic routes can neglect dependencies between reactions and their order of application. Routes may be constructed inefficiently. Cycles of protection and deprotection reactions can occur. In addition, each enumerated molecule generated by this type of algorithm will have its own unique synthetic route, which will not necessarily be similar. These problems can be avoided by using established routes rather than generating new ones. This has the advantage that practical synthetic routes can be selected based on the availability of educts or catalysts, or tailored to individual needs, or already established in in-house protocols.

Basing the generation process on a single synthetic pathway has the additional advantage of maximizing synthetic efficiency, so that all generated molecules can theoretically be synthesized following the same sequence of reactions. For these reasons, this strategy was chosen for Synthesia. Not only does it avoid most of the disadvantages described above, but it also allows the synthetic chemist to bring their individual experience as to which routes to select. Obvious choices would be those that are successfully established in the laboratory or are likely to lead to high yields. The option to decide which reactants in the synthetic pathway should be varied provides a further opportunity to include one's own intuition.

Synthesia cannot generate novel synthetic pathways, and the output that can be generated is also heavily constrained by the given input. However, this is an opportunity for chemists to influence the resulting space of structural analogues rather than a disadvantage. Of course, chemists are expected to select not only suitable synthetic routes, but also a set of building blocks that are trivial to or available either commercially or in-house.

Since Synthesia's publication [D2], its algorithm has been re-implemented by Levin et al. [96], who use it to identify and score synthetic routes that provide access to a large accessible space of structural analogues. Levin et al. emphasize the importance of diversification of accessible structural analogues. Identifying routes that lead to a wider accessible chemical space can help to allocate synthetic resources more effectively, allowing early selection of robust synthesis plans and speeding up the discovery process [96]. This argument follows the same line of reasoning as for synthetic efficiency as discussed in section 3.2. In addition, Levin et al. developed property models to predict the distribution of properties within the structural analogue space without the need for enumeration, thus aiding in the selection and possible restriction of reactants. The latter can be achieved by filtering the chemical structures involved in the route, as in Synthesia, to create only structural analogues (or use building blocks) with desired properties. However, predicting properties to get an idea of what the accessible space looks like without having to fully enumerate it is helpful in choosing between different available synthetic routes and significantly shortens the required run time. This is not yet possible in Synthesia.

Compared to commercially available options such as PathFinder, Synthesia does not have a graphical user interface (GUI). PathFinder can be used via a GUI that is integrated into the software suite provided by Schrödinger [128]. Synthesia is available as a command line tool that returns files without graphical representations of the results. However, future approaches for visualization of structural analogues together with their synthetic routes have already been discussed in Section 2.4.

3.4 Current Limitations and Further Directions

Rather than separating molecular design and synthesis, Synthesia offers the ability to commit to synthetic pathways early on, with easy access to many structural analogues. This has the potential to speed up the DMTA cycle and eliminate the need for the development of entirely new synthetic pathways for each test candidate. Structural analogues are generated by exchanging building blocks or intermediate structures in the initial synthetic pathway. New structures are built solely according to the rules provided by the generic reaction pattern written in SMIRKS. Here Synthesia has the same problem as many CASP methods (see Section 1.2), where the outcome is highly dependent on the quality of the generic reaction patterns. Depending on the specificity of the given pattern, these rules may not be sufficient to check the feasibility of reactions involving the newly selected reactants and may lead to the generation of improbable results. It is highly recommended to use the recursion option of the SMARTS language to additionally describe the relevant chemical environment.

As discussed in Section 1.2.6, ranking synthetic paths is not a trivial task and therefore difficult to incorporate. Established synthetic accessibility scores (see Section 1.2.2) could be used as additional filters to increase confidence in the proposed solutions. In addition, assessments of reaction outcomes or yield predictions (see Section 1.2.3) could be beneficial for all reactions that are performed according to the retrosynthetic route. If a model is to be included to predict the reaction outcome, a template or sequence-based approach should be chosen to provide a different view of the problem. Template-based approaches are less ideal as their information base is already covered by the reaction pattern in the route. The most likely product structures predicted by the model can then be compared with the product structures in the route. However, the advantageous predictions would have to be balanced against the resulting increase in run time to maintain overall performance and utility. Nevertheless, even with these adjustments, as with all CASP tools, a true evaluation of the quality of the routes generated can only be achieved through experiments or the expertise of chemists. As the latter are responsible for providing the initial data and influence the design and modification process of both the structure and the route, there is a considerable degree of confidence in the quality of the results.

A further improvement could be to include not only calculable physicochemical properties but also costs or delivery times for building blocks. As these properties have to be provided by the chemist, a simple filtering on his part could be sufficient to make the process more practical. A more interesting idea would be a yield or cost estimate for the entire synthesis, but with the limited information available, only a trivial, additive approach seems feasible.

Looking at the exchange routine itself, currently only the single exchange of one reactant at a time is realized. However, there may be applications where multiple reactants or even reaction templates need to be exchanged to fulfill design requests. Therefore, the ability of synthetic chemists to manipulate all components of a retrosynthetic pathway is the focus of the third publication [D3] in this thesis. Further explanations and discussions on this topic can be found in Chapter 4.

Chapter 4

Full Modification Control over Retrosynthetic Routes for Guided Optimization of Lead Structures

In times when synthetic routes no longer have to be developed and written by hand, but can be predicted by CASP tools (compare Chapter 1.2.4), the functionality for adapting predefined synthetic routes to individual requirements with the suggestion of suitable alternatives is needed. Building on the algorithm and data structures described in [D2], the third publication [D3] of this thesis allows to customize not only the producible space of the synthetically accessible structural analogues but also their synthetic routes. Taking full advantage of the chemist's expertise, all components of a synthesis can be specified for modification. Additional features to simplify the synthetic routes and to optionally facilitate the application of the algorithm by addressing individual needs are also included. In the following, the underlying algorithms are described, example applications are presented and results are discussed.

4.1 Methodical Summary

In this section, algorithms from the publication [D3] are described that enable two different starting points: Either the exact positions in the retrosynthetic route where changes are desired must be specified, or a substructure to be modified within the lead structure is selected. In the latter case, the algorithm automatically identifies corresponding components in the tree and suggests modification options. Both approaches offer the possibility of replacing or omitting reaction nodes, making changes to multiple reactant structures simultaneously, and defining a target function that specifies desired or undesired substructures within the structural analogues to be generated. All additional algorithms published in [D3] can be utilized in conjunction with the structural constraints described in 3.1.2 or [D2]. This integration ensures that the resulting structural analogues or the selected substitute reactant structures meet desired physicochemical profiles.

In addition to the expected input data already discussed in [D2], the algorithm requires possible substitute reaction patterns if a reaction node is to be exchanged. Pharmaceutical companies often have their own set of in-house applicable reactions from internal laboratory notebooks. However, there are also open-source reaction data sets that can be used. For further information see Section 1.1.3.

4.1.1 Exchange Single Reactant Structures

Replacing a single reactant structure in the retrosynthetic tree is enabled by the already described tree traversing algorithm of Synthesia [D2](see Section 3.1.1). It involves checking the integrity of the route and generating the modified target structure with the introduced changes. The algorithm is summarized in Chapter 3 or described in full detail in [D2]. To increase efficiency, the algorithm has been extended to include a fast filtering step of the given building block set, which onyl considers substitute candidates that match the SMARTS expression of the original reactant structure. [D3]

4.1.2 Simultaneous Exchange of Multiple Reactant Structures

After a single reactant exchange, the next step is to enable the simultaneous exchange of multiple reactant structures. This not only allows further individualisation of the synthesis route, but also opens up the possibility of exploring a larger structural analogue space. The algorithm is based on the same steps as the single reactant exchange algorithm. However, now multiple subtrees starting from all exchanged reactant structures have to be considered instead of just one (see Figure 3.1). All nodes are sorted in reverse topological order to traverse the tree and check its validity. Due to the combinatorial nature of the exchange possibilities, this algorithm is computationally expensive. Therefore, multithreading has been included to parallelize computations. In addition, to limit this complexity, it is advisable to define structural constraints to restrict the number of suitable substitute candidates. At the start of each exchange routine, the chemist is given information about the number of possible combinations to be calculated and can adjust the parameters, if necessary, to achieve acceptable run times. It should be kept in mind that the calculation times can vary considerably depending on the given data and parameters. [D3]

4.1.3 Exchange Reaction Templates

The reaction exchange algorithm is designed to customize synthetic pathways by replacing reactions within the corresponding retrosynthetic tree. To run this algorithm, in addition to a list of potential building blocks and the initial retrosynthetic route, potential reaction substitutes, and optionally predefined filtering criteria are required. The reaction exchange algorithm includes four steps, as described below. The first step is an optional pre-filtering of the given reaction substitutes. Possible filter criteria include reaction names and numerical classification schemes based on the NameRxn [129, 130] software. The second step is to evaluate the compatibility of the proposed substitutes with the retrosynthetic tree. Third, new trees are generated for each reaction substitute. In a final step reactant structures are substituted, if specified. Here the algorithm for the simultaneous exchange of multiple reactant structures or the algorithm for the exchange of single reactant structures can be added, depending on the input of the synthetic chemist and the requirements of the newly selected reaction. [D3]

4.1.4 Skip Reaction Nodes

Focusing on reactions in synthetic routes, it was found that certain potential substitutes in the exchange algorithms did not work due to deprotection/protection reactions and the presence or absence of corresponding protecting groups in the offered substitutes. To solve this problem, the option of 'reaction skipping' (see Figure 4.1) has been introduced to avoid unnecessary transformations. While traversing the tree, the algorithm automatically recognizes reaction nodes that block otherwise suitable substitutes and skips these nodes where possible: If a reaction cannot be used to generate a modified product structure with the current reactant the algorithm checks whether the reactant structure can be used with the subsequent reaction. If possible, the first reaction is skipped, maintaining the integrity of the route and composition. This extension of the algorithm allows the route to be shortened and simplified if suitable reactant structures are available. Currently, only reactions that transform one structure to another can be skipped. [D3]

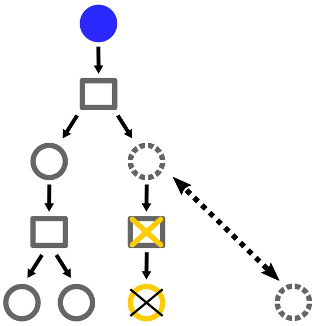


Figure 4.1: The diagram illustrates the generic representation of skipping a reaction node. An abstract retrosynthetic pathway is visualized, with circles representing chemical structures and rectangles representing generic reactions. The root of the tree (blue) represents the lead structure, while the component open for exchange is highlighted in yellow. Components indirectly affected by the exchange are shown as dashed elements. Components crossed out are dismissed during the algorithm. The image is extracted from [D3].

4.1.5 Determine Site of Modification Automatically

Taken together, the algorithms described above allow modifications to be made to all components of a retrosynthetic route. Chemists can specify changes in the physicochemical properties of the lead structure together with the locations of the modifications in the retrosynthetic route. This modification strategy requires prior expertise and knowledge of the route. Alternatively, the focus can be solely on the lead structure and only the substructures to be modified within the synthetically accessible structural analogues to be generated are known. The product exchange mode has been developed to support this application scenario. For the resulting algorithm, the substructure to be modified within the lead structure must first be defined. This can be done using a target function written as a SMARTS pattern. The algorithm identifies the relevant nodes or sub-trees of the retrosynthetic tree with an internal atom mapping. Without further input from the synthetic chemist, the appropriate exchange and modification process is started and structural analogues matching the target function are generated. The general idea is visualized in Figure 4.2. [D3]

4.2 Validation

As discussed in the previous chapter, it is not an easy task to validate Synthesia's algorithms, especially with regard to the modified retrosynthetic routes. The following section summarizes the overall validation tactics chosen for the newly added algorithms. First, a general overview of the advantages of the algorithms and their functionalities

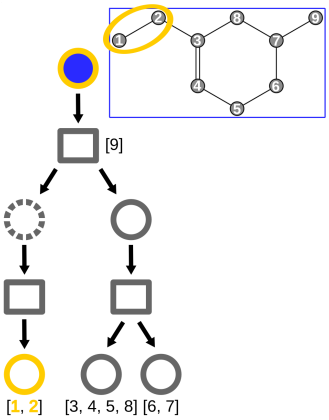


Figure 4.2: The diagram illustrates the use of a target function to start the modification process. An abstract retrosynthetic pathway is visualized, with circles representing chemical structures and rectangles representing generic reactions. The root of the tree (blue) represents the target structure, while the component open for exchange is highlighted in yellow. Components indirectly affected by the exchange are shown as dashed elements. The target function is circled yellow in the target structure in the blue rectangle. Atom mappings are visualized by number. The image is extracted from [D3].

is given. Then, a study is carried out to analyze the different structural space coverage between the different exchange options. Finally, another application scenario is presented which studies the possibility of synthetically feasible scaffold hopping. As in the previous publication [D2], the initial retrosynthetic routes were created using the AiZynthFinder software [19] and a Enamine Building Blocks Collection [125] was used as a source of potential substitute reactant structures. As potential substitute reaction patterns, the templates used for model training provided by the AiZynthFinder software extracted from the USPTO were used [23].

Using an exemplary target, futibatinib [131], the results of each newly added algorithm or functionality are illustrated in [D3]. Futibatinib is a kinase inhibitor used for the treatment of intrahepatic cholangiocarcinoma. Figure 4.3 shows a structural analogue of futibatinib generated by the exchange of a reaction node together with a reactant structure. To illustrate the usefulness of reaction exchange, the focus has been on reaction 4, an 'N-acylation to amide' reaction. The given pattern is rather specific, only reactants with acyl halides and vinyl substructures are permitted. The list of possible exchange reactions was pre-filtered by reaction name. All remaining candidates were run through the exchange routine described in 4.1.3. Among others, a reaction from the same class was identified in which a carboxylic acid can be used as the first reactant. This led to the generation of a patent-registered futibatinib analogue that would not have been accessible via the unmodified pathway or the single/multiple

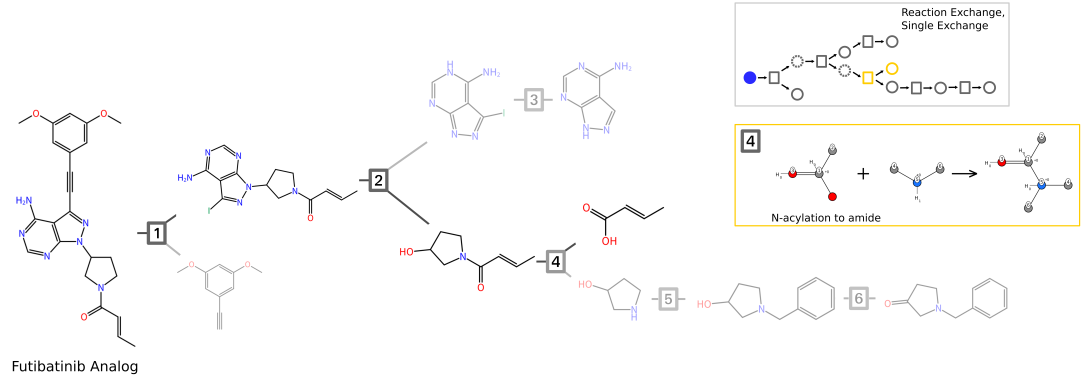


Figure 4.3: Visualization of a futibatinib analogue with a retrosynthetic route. Unmodified parts of the route (in comparison to the initial one) are grayed out. The structural analogue was generated due to the exchange of a reaction scheme and a single reactant. The exchanged components are marked yellow in the abstract representation of the route in the upper right corner. The newly added reaction scheme is visualized in a yellow rectangle using functionality described in [D3]. The image is extracted from [D3].

reactant exchange algorithms alone. [D2]

Figure 4.4 shows an example of the automatic determination of the modification site, using a SMARTS pattern to describe the pyrrolidine substructure to be exchanged. The generated futibatinib analogue lacks the original substructure. Besides the target function, no further specifications were needed for this result. The original, unmodified route and other examples can be found in [D3].

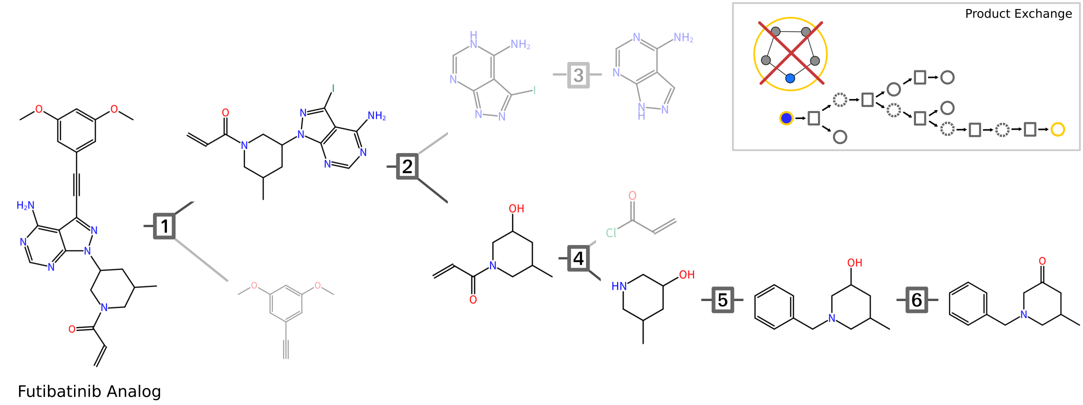


Figure 4.4: Visualization of a futibatinib with a retrosynthetic route. Unmodified parts of the route (in comparison to the initial one) are grayed out. A SMARTS pattern, identifying the pyrrolidine substructure in the original compound is used as a target function for the automatic determination of the site of modification in the retrosynthetic tree. The identified and exchanged component is marked yellow in the abstract representation. The image is extracted from [D3].

In the subsequent experiment the synthetically accessible structural analogue space of oteseconazole, a cytochrome P450 (CYP) 51 inhibitor, was explored, by performing all possible exchanges for the given target and retrosynthetic route without additional structural constraints. All results can be found in [D3]. In the following, the resulting conclusions are summarized. The results show that the number of structural analogues generated varies considerably for different exchange algorithms and nodes. In general, more degrees of freedom lead to more analogues, although some nodes generate significantly more structures due to their position in the tree or the restrictiveness of the subsequent reaction. The average similarity of the generated analogues to the original lead structure decreases with more simultaneous exchanges, with the highest similarities observed when only one or two starting structures are exchanged. Greater dissimilarity is observed when the exchange occurs closer to the root node. In conclusion, different exchange methods reach different parts of the structural analogue space and serve different purposes. [D3]

In the context of synthesis, modifying the molecular scaffold is often considered to be more complex than altering terminal groups, especially when maintaining biological activity is critical. To demonstrate Synthesia's ability to modify a molecule's scaffold, while limiting the exchange to bioisosteric replacements and still keeping the resulting structural analogue synthetically accessible, a final experiment was performed. A list of common linkers in bioactive molecules from Ertl et al. [57] was used together with the target structure abrocitinib [132], an approved janus kinase 1 (JAK1) inhibitor that has a role in the treatment of dermatitis. First, all linker substructures present in the target molecule that divide the molecule into segments of at least three heavy atoms are identified. Two of the four linker substructures identified in abrocitinib are shown on the left-hand side of the Figure 4.5, marked in color in the target structure. Synthesia was then used to generate all possible structural analogues for abrocitinib and identify those in which one of the originally identified linker substructures was replaced by an alternative linker from the set provided by Ertl et al. The identified replacement linkers for two of the original abrocitinib linkers are shown in Figure 4.5 on the right. All replacement linkers shown are synthetically accessible via the original retrosynthetic route with modifications calculated by Synthesia. Additional results for two other linker substructures can be found in [D3]. Studies of this type are important due to the inherent complexity associated with linker substructure replacement. The identification of synthetically feasible linker structures from a predetermined set, guided by a specific

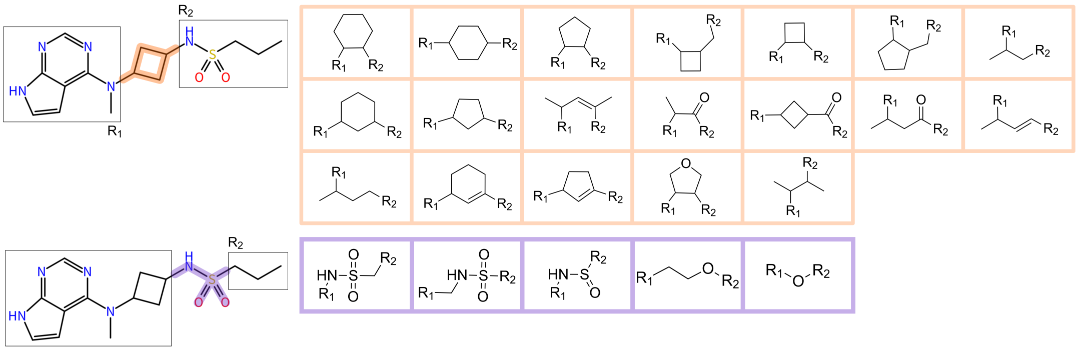


Figure 4.5: Visualization of the results of the synthetic accessibility assessment of potential linker substitutes for bioisosteric linker replacement. On the left side is the original structure, abrocitinib, visualized twice with identified linker structures marked in color. On the right, are the corresponding sets of synthetic accessible linker substitutes for the specified linker structures. The image is extracted from [D3].

retrosynthetic route for a target compound, has the potential to significantly improve and optimize the process of implementing bioisosteric replacements. [D3]

4.3 Discussion

Synthetic chemists adapt and modify synthetic routes for a variety of reasons: to improve yield, selectivity or other physicochemical properties of the final product, or to reduce overall costs. Synthetic challenges can be circumvented and specific regulatory requirements can be met while still achieving the desired target modification. The work presented in [D3] provides chemists with a software tool that supports the modification process of complete synthetic routes based on the chemist's expertise, but adds value by automatically calculating suitable substitutes. This supports both individual case studies where a specific target is to be modified by adjusting the synthetic route in a certain direction, as well as the generation of even broader structural analogue spaces based on given synthetic routes with additional enumeration options compared to the first version of Synthesia [D2].

The foundational principle of Synthesia's algorithm has already been discussed and compared to other approaches in Chapter 3. Of the software tools presented in Section 1.3, only LinChemIn can explicitly modify synthetic routes using, among others, route arithmetic operations. LinChemIn is a Python toolkit, designed for cheminformatics activities on synthetic routes and reaction networks. It facilitates conversion between various data formats and models, enabling route-level analysis and operations such as route comparisons and descriptor calculation, none of which are possible with Synthesia.

Unsurprisingly, since it is the obvious choice, LinChemIn and Synthesia share the same architecture for the core data structure for synthetic routes, a directed acyclic graph, with chemical and reaction nodes linked together (compare Chapter 3). In contrast to Synthesia, where only one synthetic route at a time can be processed, LinChemIn offers the possibility to combine synthetic routes into synthetic forests that have different roots but common intermediate structures, which can then be one of several possible connected subgraphs in a chemical reaction network. Particularly relevant for this comparison are the single route editing options of LinChemIn, where users can add or remove chemical reaction nodes from the graph, while the chemical consistency of the resulting synthetic route is guaranteed by the software. Published after this work, LinChemIn follows the same motivations as Synthesia and states *'editing routes [...] is a key requirement for any informatics system that aims to leverage the knowledge and experience of scientists [...]'.* The difference lies primarily in the implementation; Synthesia allows the replacement of certain reaction nodes and has only an automatic routine to remove unsuitable reactions (see Section 4.1) and no explicit option to remove or add reaction and chemical nodes.

In addition, LinChemIn provides functionality in the form of node descriptors and metrics to compare routes with each other, especially after the user made modifications, while Synthesia provides the synthetic chemist with a set of complete synthetic routes to choose from, resulting from the desired modification. Both encourage explicit synthetic route modification and design to allow the synthetic chemist to customize synthetic routes to their needs rather than starting from scratch. Finally, it should be mentioned that the compared approaches have a different focus. LinChemIn is designed as a suite of functionalities. It serves as a library that developers can integrate into their programs to achieve a specific purpose. Synthesia on the other hand offers ready-to-use software with an algorithmic solution for specific research questions.

4.4 Current Limitations and Further Directions

With the additional functionality presented, Synthesia enables the synthetic chemist to customize lead structures together with their retrosynthetic routes. with the reaction exchange options, an additional verification of the viability of the reaction (see section 1.2.3) for each reaction in the tree, going beyond the chemical consistency rules provided by the SMIRKS pattern, could be helpful. As already discussed in Section 3.4, such a verification step could estimate the practicability of the proposed reactions and thus improve the overall reliability of the synthetic route. The presented solutions to the synthetic chemist could thereby be ranked and sorted. Ultimately, however, the chemists will use their expertise to select the most appropriate option. As they will have already used their knowledge to define the position and possibly the direction of the modification (determining the type or class of the substituted new reaction), this may be sufficient as a feasibility check.

The presented exchange of reactions and the simultaneous introduction of multiple new reactants can significantly alter the original route. This approach can lead to greater variability, potentially allowing the exploration of a broader chemical space of structural analogues. However, the underlying concept of Synthesia is based on the assumption that the synthetic chemist has selected an already viable synthetic route that only needs some degree of customization, either in the steps of the route or in the properties of the target structure. In order not to deviate too far from the original route, there is currently no function for exchanging several reaction nodes simultaneously. On an implementation level this could be added easily. However, this would add further combinatorial possibilities for the synthetic chemist to orchestrate.

Concerning the functionality of reaction substitution, a further point needs to be discussed: the availability and quality of potential replacement generic reaction pattern lists. Not all synthetic chemists have access to lists of suitable or in-house reactions written in the expected format, and extraction or writing them by hand is time-consuming. The quality of the results of the Synthesia reaction exchange results is highly dependent on the quality of the patterns provided. Incorrect or incomplete reaction patterns can lead to unreliable syntheses or no results at all, which undermines the reliability of the approach. However, it should be noted that there are some publicly available reaction datasets (as described in Section 1.1.3) that are of acceptable quality.

An extension with additional functions is always imaginable: For example, the option to add or remove complete reaction nodes in the synthetic route, as in LinChemIn, could simply be included in the existing implementation. This would allow for more flexibility in the modification and offer the possibility to investigate different pathways more thoroughly.

A significant current limitation of the approaches described is the potential time required for the simultaneous exchange o multiple reactants. The run time is highly dependent on the number of reactants to be exchanged, the available number of suitable substitutes compatible with the synthetic route, and the settings of search query constraints, either limiting the physicochemical properties of the possible substitutes or the generated target structure. Especially without physicochemical constraints, due to the combinatorial nature of the enumeration approach, the run time can become impracticable. At present, the number of possible modified product structures that can be generated is provided directly at the beginning of the calculations to allow the synthetic chemist to estimate the runtime and decide whether the calculation is feasible or whether it is better to set further constraints. An additional approach could be to integrate active learning approaches, as used by Levin et al. [96] to predict the properties of the generatable structural analogue space and use this information to select the appropriate parameters for the exchange routine.

Chapter 5

Conclusion

Synthesizability is crucial in virtual drug design as it ensures that the proposed compounds can be produced in a laboratory environment, making the transition from *in silico* to real-world testing feasible. Without considering synthesizability, there is a risk of venturing in the virtual ivory tower and designing molecules that are theoretically interesting but impossible or too costly to synthesize, resulting in a waste of resources and time. With this thesis, three different algorithmic approaches were presented to support further integration of the question of synthesizability into the virtual drug design process.

The first research objective of this thesis was to provide algorithmic solutions for understanding, interpreting, and analyzing generic reaction patterns in the form of Reaction SMILES, Reaction SMARTS, or SMIRKS patterns. These ways of expressing chemical reactions are well established in computer-aided drug design as they are readable by both humans and machines. The languages provide a standardized way to encode chemical reactions and make it possible to specify how certain substructures within molecules should be modified. However, depending on the degree of generalization or specificity, they can be difficult to interpret straight away even for the trained human eye. The algorithm of the resulting work [D1] is described in Chapter 2. The generated software application called ReactionViewer offers a simple way to visually inspect generic reaction patterns and thus understand and verify them. The advantages of the chosen design over other visualization software are discussed using various examples. In addition, two popular generic reaction datasets are fully visualized to support their further understanding and analysis in the community.

In a second approach, the algorithm from [D2] for the synthesis-aware generation of structural analogues is presented. The resulting software application is called Synthesia. Generated structural analogues are in theory all synthesizable with the same sequence of reactions, i.e. with the same general synthetic pathway. Synthesia does not generate synthetic routes itself but builds on established routes that have to be provided by the synthetic chemist. In this way, fundamental knowledge,such as how a scaffold structure can be synthesized under ideal conditions in individual laboratories, is incorporated into the design and modification process of the lead structure in the hope of making the resulting structural analogues more practicable. The generation process involves either systematic or user-defined replacement of reactant or intermediate structures in a retrosynthetic route and forward reconstruction of the modified target structure. The physicochemical properties of the building blocks used or of the space of structural analogues can be tailored to the specific needs by specifying constraints. Synthesia was designed to fulfill the objectives of the first research topic described in Section 1.4. From the results of the experiments described in Section 3, it can be confidently stated that Synthesia is capable of generating chemical data sets based on synthetic routes that cover the space of available structural analogues that in theory can be synthesized using the same synthetic route. Furthermore, it was shown that the provided algorithm can also be used to analyze the required synthesis effort for a range of targets (compare Section 3.3).

In the third publication of this thesis [D3], an algorithmic approach for the modification of synthetic routes based on individual needs and wishes is presented. Based on the algorithm and data structures of [D2], functionalities are provided to customize all components of a retrosynthetic route, both structures (reactants, intermediates, and products) and reactions, by replacing them with suitable substitutes. The synthetic chemist can make full use of their expertise by specifying exactly which part of the route is to be adapted and how. Unnecessary reactions can be automatically detected and removed if necessary. In addition, functionalities have been added to simplify the application if required (compare Section 4.1.5). With the resulting software, a way to fulfill the third research objective was designed, combining the strengths of computer-based automation with the understanding and case-specific knowledge of synthetic chemists. Examples are given of the various exchange options where hand-defined modification sites in the synthetic route lead to advanced results customized to individual needs (compare Section 4.2). The software provided could bridge the gap between automatized route generation and human expertise by providing means to improve the routes generated by CASP tools based on external information or individual expertise and chemical intuition.

The synthetic accessibility of drug candidates is still a current research topic and consists of many smaller sub-problems. In this thesis, possible solutions for three sub-problems of this topic have been presented. However, it is not only the invention of new methods and algorithms that will drive research forward, but above all the explicit communication with the users of these solutions. It is crucial to ask synthetic chemists about their needs and to involve them in the design phase of new algorithms. It is often found that the freedom to make decisions during application and the opportunity to contribute one's own specialist knowledge are preferred to automated and ready-made solutions. Only if it is ensured that these methods are useful in practice can they support the drug discovery process. Ultimately, only time will tell whether the software approaches developed in this thesis have succeeded in doing this and whether they will support synthetic chemists in their daily tasks.

Bibliography

[1] R. L. Apodaca. *Predicting Synthetic Accessibility.* Accessed June 19, 2024. 2010. URL: https://depth-first.com/articles/2010/10/28/predicting-synthetic-accessibility/.

[2] D. R. Owen, C. M. Allerton, A. S. Anderson, L. Aschenbrenner, M. Avery, S. Berritt, B. Boras, R. D. Cardin, A. Carlo, K. J. Coffman, et al. "An oral SARS-CoV-2 Mpro inhibitor clinical candidate for the treatment of COVID-19". In: *Science* 374.6575 (2021), pp. 1586–1593.

[3] W. Gao and C. W. Coley. "The synthesizability of molecules proposed by generative models". In: *Journal of Chemical Information and Modeling* 60.12 (2020), pp. 5714–5723.

[4] D. S. Wigh, J. M. Goodman, and A. A. Lapkin. "A review of molecular representation in the age of machine learning". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12.5 (2022), e1603.

[5] D. Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36.

[6] Daylight. *SMILES - A Simplified Chemical Language.* Accessed June 08, 2024. URL: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

[7] I. U. of Pure and A. C. (IUPAC). *IUPAC.* Accessed August 03, 2024. URL: https://iupac.org.

[8] J. Brecher. "Graphical representation standards for chemical structure diagrams (IUPAC Recommendations 2008)". In: *Pure and Applied Chemistry* 80.2 (2008), pp. 277–410.

[9] I. U. of Pure and A. C. (IUPAC). *International Chemical Identifier (InChI).* Accessed August 03, 2024. 2024. URL: https://www.inchi-trust.org.

[10] D. Rogers and M. Hahn. "Extended-connectivity fingerprints". In: *Journal of Chemical Information and Modeling* 50.5 (2010), pp. 742–754.

[11] R. Winter, F. Montanari, F. Noé, and D.-A. Clevert. "Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations". In: *Chemical Science* 10.6 (2019), pp. 1692–1701.

[12] Daylight. *SMARTS - A Language for Describing Molecular Patterns.* Accessed June 08, 2024. URL: `https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html`.

[13] Daylight. *SMIRKS - A Reaction Transform Language.* Accessed June 08, 2024. URL: `https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html`.

[14] W.-D. Ihlenfeldt and J. Gasteiger. "Computer-assisted planning of organic syntheses: the second generation of programs". In: *Angewandte Chemie International Edition in English* 34.23-24 (1996), pp. 2613–2633.

[15] W. L. Chen, D. Z. Chen, and K. T. Taylor. "Automatic reaction mapping and reaction center detection". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 3.6 (2013), pp. 560–593.

[16] M. I. Systems. *MDL Reaction File Format.*

[17] G. Grethe, G. Blanke, H. Kraut, and J. M. Goodman. "International chemical identifier for reactions (RInChI)". In: *Journal of Cheminformatics* 10.1 (2018), pp. 1–9.

[18] P. N. Judson, W.-D. Ihlenfeldt, H. Patel, V. Delannée, N. Tarasova, and M. C. Nicklaus. "Adapting CHMTRN (chemistry translator) for a new use". In: *Journal of Chemical Information and Modeling* 60.7 (2020), pp. 3336–3341.

[19] S. Genheden, A. Thakkar, V. Chadimová, J.-L. Reymond, O. Engkvist, and E. Bjerrum. "AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning". In: *Journal of Cheminformatics* 12.1 (2020), p. 70.

[20] United States Patent and Trademark Office. *USPTO Patent Database.* Accessed August 10, 2024. 2024. URL: `https://www.uspto.gov/patents/search`.

[21] H. Dai, C. Li, C. Coley, B. Dai, and L. Song. "Retrosynthesis prediction with conditional graph logic network". In: *Advances in Neural Information Processing Systems* 32 (2019).

[22] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen. "Prediction of organic reaction outcomes using machine learning". In: *ACS Central Science* 3.5 (2017), pp. 434–443.

[23] D. M. Lowe. *Chemical Reactions from US Patents (1976-Sep2016) (Version 1).* Accessed June 31, 2024. 2017. URL: `https://doi.org/10.6084/m9.figshare.5104873.v1`.

[24] N. Schneider, D. M. Lowe, R. A. Sayle, and G. A. Landrum. "Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity". In: *Journal of Chemical Information and Modeling* 55.1 (2015), pp. 39–53.

[25] P. Schwaller, D. Probst, A. C. Vaucher, V. H. Nair, D. Kreutter, T. Laino, and J.-L. Reymond. "Mapping the space of chemical reactions using attention-based neural networks". In: *Nature Machine Intelligence* 3.2 (2021), pp. 144–152.

[26] M. Hartenfeller, M. Eberle, P. Meier, C. Nieto-Oberhuber, K.-H. Altmann, G. Schneider, E. Jacoby, and S. Renner. "A Collection of Robust Organic Synthesis Reactions for In Silico Molecule Design". In: *Journal of Chemical Information and Modeling* 51.12 (2011), pp. 3093–3098.

[27] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, and M. Rarey. "On the art of compiling and using 'drug-like'chemical fragment spaces". In: *ChemMedChem* 3.10 (2008), p. 1503.

[28] N. Software. *Pistachio.* Accessed January 31, 2024. URL: `http://www.nextmovesoftware.com/pistachio.html`.

[29] Elsevier. *Reaxys database.* Accessed January 31, 2024. URL: `https://www.reaxys.com`.

[30] A. C. Society. *CAS SciFinder database.* Accessed January 31, 2024. URL: `https://scifinder.cas.org`.

[31] S. Jiang, Z. Zhang, H. Zhao, J. Li, Y. Yang, B.-L. Lu, and N. Xia. "When SMILES smiles, practicality judgment and yield prediction of chemical reaction via deep chemical language processing". In: *IEEE Access* 9 (2021), pp. 85071–85083.

[32] S. M. Kearnes, M. R. Maser, M. Wleklinski, A. Kast, A. G. Doyle, S. D. Dreher, J. M. Hawkins, K. F. Jensen, and C. W. Coley. "The open reaction database". In: *Journal of the American Chemical Society* 143.45 (2021), pp. 18820–18826.

[33] P. Schwaller, A. C. Vaucher, R. Laplaza, C. Bunne, A. Krause, C. Corminboeuf, and T. Laino. "Machine intelligence for chemical reaction space". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12.5 (2022), e1604.

[34] W. L. Todsen. “ChemDoodle 6.0”. In: *Journal of Chemical Information and Modeling* 54.8 (2014), pp. 2391–2393.

[35] A. Fombona-Pascual, J. Fombona, and E. Vázquez-Cano. “VR in chemistry, a review of scientific research on advanced atomic/molecular visualization”. In: *Chemistry Education Research and Practice* 23.2 (2022), pp. 300–312.

[36] W. D. Ihlenfeldt, E. E. Bolton, and S. H. Bryant. “The PubChem chemical structure sketcher”. In: *Journal of Cheminformatics* 1 (2009), pp. 1–9.

[37] H. E. Pence and A. Williams. “ChemSpider: an online chemical information resource”. In: (2010).

[38] H. Bergwerf. *MolView*. Accessed May 31, 2024. URL: `https://molview.org/`.

[39] Chemaxon. *Marvin*. Accessed May 31, 2024. URL: `https://chemaxon.com/marvin`.

[40] G. Landrum. “RDKit documentation”. In: *Release 2013* 1.1-79 (), p. 4.

[41] RdKit. *RDKit - Drawing Molecules*. Accessed May 31, 2024. URL: `https://www.rdkit.org/docs/GettingStartedInPython.html#drawing-molecules`.

[42] P. Csizmadia. “MarvinSketch and MarvinView: molecule applets for the World Wide Web”. In: (1999).

[43] K. Schomburg, H.-C. Ehrlich, K. Stierand, and M. Rarey. “From Structure Diagrams to Visual Chemical Patterns”. In: *Journal of Chemical Information and Modeling* 50.9 (2010), pp. 1529–1535.

[44] C. A. James. *Daylight theory manual*. Accessed June 02, 2024. 2004. URL: `https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html`.

[45] J. B. Baell and G. A. Holloway. “New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays”. In: *Journal of Medicinal Chemistry* 53.7 (2010), pp. 2719–2740.

[46] Chemaxon. *Marvin*. Accessed May 31, 2024. URL: `https://marvinjs-demo.chemaxon.com/latest/`.

[47] C. Ehrt, B. Krause, R. Schmidt, E. S. Ehmki, and M. Rarey. “SMARTS.plus – a toolbox for chemical pattern design”. In: *Molecular Informatics* 39.12 (2020), p. 2000216.

[48] *Smarts.plus*. Accessed June 19, 2024. URL: `https://smarts.plus/`.

[49] A. Düfert. “Organische Synthesemethoden: Grundlagen, Mechanismen und Anwendungen”. In: Chapter 11: [Prinzipien der Syntheseplanung]. Berlin, Germany: Springer, 2020. Chap. 11. ISBN: 978-3-662-60483-2.

[50] G. Vleduts. “Concerning one system of classification and codification of organic reactions”. In: *Information Storage and Retrieval* 1.2-3 (1963), pp. 117–146.

[51] E. J. Corey and W. T. Wipke. “Computer-Assisted Design of Complex Organic Syntheses: Pathways for molecular synthesis can be devised with a computer and equipment for graphical communication.” In: *Science* 166.3902 (1969), pp. 178–192.

[52] I. W. Davies. “The digitization of organic synthesis”. In: *Nature* 570.7760 (2019), pp. 175–181.

[53] Z. Tu, I. Levin, and C. W. Coley. “Computer-Assisted Synthesis Planning”. In: *Enabling Tools and Techniques for Organic Synthesis: A Practical Guide to Experimentation, Automation, and Computation* (2023), pp. 423–459.

[54] Z. Zhong, J. Song, Z. Feng, T. Liu, L. Jia, S. Yao, T. Hou, and M. Song. “Recent advances in artificial intelligence for retrosynthesis”. In: *arXiv preprint arXiv:2301.05864* (2023).

[55] Y. Jiang, Y. Yu, M. Kong, Y. Mei, L. Yuan, Z. Huang, K. Kuang, Z. Wang, H. Yao, J. Zou, C. W. Coley, and Y. Wei. “Artificial intelligence for retrosynthesis prediction”. In: *Engineering* (2022).

[56] G. Skoraczyński, M. Kitlas, B. Miasojedow, and A. Gambin. “Critical assessment of synthetic accessibility scores in computer-assisted synthesis planning”. In: *Journal of Cheminformatics* 15.1 (2023), p. 6.

[57] P. Ertl and A. Schuffenhauer. “Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions”. In: *Journal of Cheminformatics* 1 (2009), pp. 1–11.

[58] M. Voršilák, M. Kolář, I. Čmelo, and D. Svozil. “SYBA: Bayesian estimation of synthetic accessibility of organic compounds”. In: *Journal of Cheminformatics* 12.1 (2020), pp. 1–13.

[59] J. Yu, J. Wang, H. Zhao, J. Gao, Y. Kang, D. Cao, Z. Wang, and T. Hou. “Organic compound synthetic accessibility prediction based on the graph attention mechanism”. In: *Journal of Chemical Information and Modeling* 62.12 (2022), pp. 2973–2986.

[60] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen. "SCScore: synthetic complexity learned from a reaction corpus". In: *Journal of Chemical Information and Modeling* 58.2 (2018), pp. 252–261.

[61] A. Thakkar, V. Chadimová, E. J. Bjerrum, O. Engkvist, and J.-L. Reymond. "Retrosynthetic accessibility score (RAscore)–rapid machine learned synthesizability classification from AI driven retrosynthetic planning". In: *Chemical Science* 12.9 (2021), pp. 3339–3349.

[62] C.-H. Liu, M. Korablyov, S. Jastrzebski, P. Włodarczyk-Pruszynski, Y. Bengio, and M. Segler. "RetroGNN: fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software". In: *Journal of Chemical Information and Modeling* 62.10 (2022), pp. 2293–2300.

[63] M. H. Segler and M. P. Waller. "Neural-symbolic machine learning for retrosynthesis and reaction prediction". In: *Chemistry–A European Journal* 23.25 (2017), pp. 5966–5971.

[64] J. N. Wei, D. Duvenaud, and A. Aspuru-Guzik. "Neural networks for the prediction of organic chemistry reactions". In: *ACS Central Science* 2.10 (2016), pp. 725–732.

[65] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay, and K. F. Jensen. "A graph-convolutional neural network model for the prediction of chemical reactivity". In: *Chemical Science* 10.2 (2019), pp. 370–377.

[66] K. Do, T. Tran, and S. Venkatesh. "Graph transformation policy network for chemical reaction prediction". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 750–760.

[67] J. Nam and J. Kim. "Linking the neural machine translation and the prediction of organic chemistry reactions". In: *arXiv preprint arXiv:1612.09529* (2016).

[68] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee. "Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction". In: *ACS Central Science* 5.9 (2019), pp. 1572–1583.

[69] D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher, and A. G. Doyle. "Predicting reaction performance in C–N cross-coupling using machine learning". In: *Science* 360.6385 (2018), pp. 186–190.

[70] F. Sandfort, F. Strieth-Kalthoff, M. Kühnemund, C. Beecks, and F. Glorius. “A structure-based platform for predicting chemical reactivity”. In: *Chem* 6.6 (2020), pp. 1379–1390.

[71] P. Schwaller, A. C. Vaucher, T. Laino, and J.-L. Reymond. “Prediction of chemical reaction yields using deep learning”. In: *Machine Learning: Science and Technology* 2.1 (2021), p. 015016.

[72] E. J. Corey, A. K. Long, and S. D. Rubenstein. “Computer-assisted analysis in organic synthesis”. In: *Science* 228.4698 (1985), pp. 408–418.

[73] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, and B. A. Grzybowski. “Computer-assisted synthetic planning: the end of the beginning”. In: *Angewandte Chemie International Edition* 55.20 (2016), pp. 5904–5937.

[74] M. H. Segler, M. Preuss, and M. P. Waller. “Planning chemical syntheses with deep neural networks and symbolic AI”. In: *Nature* 555.7698 (2018), pp. 604–610.

[75] M. E. Fortunato, C. W. Coley, B. C. Barnes, and K. F. Jensen. “Data augmentation and pretraining for template-based retrosynthetic prediction in computer-aided synthesis planning”. In: *Journal of Chemical Information and Modeling* 60.7 (2020), pp. 3398–3407.

[76] P. Seidl, P. Renz, N. Dyubankova, P. Neves, J. Verhoeven, J. K. Wegner, M. Segler, S. Hochreiter, and G. Klambauer. “Improving few-and zero-shot reaction template prediction using modern hopfield networks”. In: *Journal of Chemical Information and Modeling* 62.9 (2022), pp. 2111–2120.

[77] S. Zheng, J. Rao, Z. Zhang, J. Xu, and Y. Yang. “Predicting retrosynthetic reactions using self-corrected transformer neural networks”. In: *Journal of Chemical Information and Modeling* 60.1 (2019), pp. 47–55.

[78] V. R. Somnath, C. Bunne, C. Coley, A. Krause, and R. Barzilay. “Learning graph models for retrosynthesis prediction”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 9405–9415.

[79] M. Sacha, M. Błaz, P. Byrski, P. Dabrowski-Tumanski, M. Chrominski, R. Loska, P. Włodarczyk-Pruszynski, and S. Jastrzebski. “Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits”. In: *Journal of Chemical Information and Modeling* 61.7 (2021), pp. 3273–3284.

[80] S.-W. Seo, Y. Y. Song, J. Y. Yang, S. Bae, H. Lee, J. Shin, S. J. Hwang, and E. Yang. "GTA: Graph truncated attention for retrosynthesis". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 35. 1. 2021, pp. 531–539.

[81] C. W. Coley, D. A. Thomas III, J. A. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, et al. "A robotic platform for flow synthesis of organic compounds informed by AI planning". In: *Science* 365.6453 (2019), eaax1566.

[82] eMolecules. Accessed January 09, 2024. URL: `https://www.emolecules.com/`.

[83] S. Aldrich. Accessed January 09, 2024. URL: `https://www.sigmaaldrich.com`.

[84] T. Sterling and J. J. Irwin. "ZINC 15–ligand discovery for everyone". In: *Journal of Chemical Information and Modeling* 55.11 (2015), pp. 2324–2337.

[85] L. Saigiridharan, A. K. Hassen, H. Lai, P. Torren-Peraire, O. Engkvist, and S. Genheden. "AiZynthFinder 4.0: developments based on learnings from 3 years of industrial application". In: *Journal of Cheminformatics* 16.1 (2024), p. 57.

[86] A. Bøgevig, H.-J. Federsel, F. Huerta, M. G. Hutchings, H. Kraut, T. Langer, P. Low, C. Oppawsky, T. Rein, and H. Saller. "Route design in the 21st century: the IC SYNTH software tool as an idea generator for synthesis prediction". In: *Organic Process Research & Development* 19.2 (2015), pp. 357–368.

[87] RXN. Accessed January 09, 2024. URL: `https://rxn.res.ibm.com/`.

[88] Spaya. Accessed January 09, 2024. URL: `https://iktos.ai/spaya/`.

[89] ChemAIRS. Accessed January 09, 2024. URL: `https://www.chemical.ai/`.

[90] K. Maziarz, A. Tripp, G. Liu, M. Stanley, S. Xie, P. Gaiński, P. Seidl, and M. Segler. "Re-evaluating Retrosynthesis Algorithms with Syntheseus". In: *arXiv preprint arXiv:2310.19796* (2023).

[91] S. Genheden and E. Bjerrum. "PaRoutes: towards a framework for benchmarking retrosynthesis route predictions". In: *Digital Discovery* 1.4 (2022), pp. 527–539.

[92] B. Mikulak-Klucznik, P. Gołębiowska, A. A. Bayly, O. Popik, T. Klucznik, S. Szymkuć, E. P. Gajewska, P. Dittwald, O. Staszewska-Krajewska, W. Beker, et al. "Computational planning of the synthesis of complex natural products". In: *Nature* 588.7836 (2020), pp. 83–88.

[93] E. J. Corey and X.-M. Cheng. *The Logic of Chemical Synthesis*. New York: Wiley, 1995. ISBN: 978-0-471-11594-8.

[94] M. Pasquini and M. Stenta. “LinChemIn: SynGraph - a data model and a toolkit to analyze and compare synthetic routes”. In: *Journal of Cheminformatics* 15.1 (2023), p. 41.

[95] M. Pasquini and M. Stenta. “LinChemIn: Route Arithmetic-Operations on Digital Synthetic Routes”. In: *Journal of Chemical Information and Modeling* 64.6 (2024), pp. 1765–1771.

[96] I. Levin, M. E. Fortunato, K. L. Tan, and C. W. Coley. “Computer-aided evaluation and exploration of chemical spaces constrained by reaction pathways”. In: *AIChE journal* 69.12 (2023), e18234.

[97] M. A. Gallop, R. W. Barrett, W. J. Dower, S. P. Fodor, and E. M. Gordon. “Applications of combinatorial technologies to drug discovery. 1. Background and peptide combinatorial libraries”. In: *Journal of Medicinal Chemistry* 37.9 (1994), pp. 1233–1251.

[98] T. Hoffmann and M. Gastreich. “The next level in chemical space navigation: going far beyond enumerable compound libraries”. In: *Drug Discovery Today* 24.5 (2019), pp. 1148–1156.

[99] L. Bellmann. “Algorithmische Methoden für kombinatorische chemische Bibliotheken”. PhD thesis. Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, 2022.

[100] R. Schmidt, R. Klein, and M. Rarey. “Maximum common substructure searching in combinatorial make-on-demand compound spaces”. In: *Journal of Chemical Information and Modeling* 62.9 (2021), pp. 2133–2150.

[101] L. Bellmann, P. Penner, and M. Rarey. “Topological similarity search in large combinatorial fragment spaces”. In: *Journal of Chemical Information and Modeling* 61.1 (2020), pp. 238–251.

[102] M. Rarey and M. Stahl. “Similarity searching in large combinatorial chemistry spaces”. In: *Journal of Computer-Aided Molecular Design* 15 (2001), pp. 497–520.

[103] Enamine. *REAL Space*. Accessed June 08, 2024. URL: `https://enamine.net/compound-collections/real-compounds/real-space-navigator`.

[104] W. AppTec. *GalaXi Space.* Accessed June 08, 2024. URL: `https://wuxibiology.com/drug-discovery-services/hit-finding-and-screening-services/virtual-screening/`.

[105] X. Q. Lewell, D. B. Judd, S. P. Watson, and M. M. Hann. “Recap retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry”. In: *Journal of Chemical Information and Computer Sciences* 38.3 (1998), pp. 511–522.

[106] K. Kawai, N. Nagata, and Y. Takahashi. “De novo design of drug-like molecules by a fragment-based molecular evolutionary approach”. In: *Journal of Chemical Information and Modeling* 54.1 (2014), pp. 49–56.

[107] P. Polishchuk. “CReM: chemically reasonable mutations framework for structure generation”. In: *Journal of Cheminformatics* 12.1 (2020), p. 28.

[108] G. M. Ghiandoni, M. J. Bodkin, B. Chen, D. Hristozov, J. E. Wallace, J. Webster, and V. J. Gillet. “RENATE: a pseudo-retrosynthetic tool for synthetically accessible de Novo design”. In: *Molecular Informatics* 41.4 (2022), p. 2100207.

[109] F. Urbina, C. T. Lowden, J. C. Culberson, and S. Ekins. “MegaSyn: integrating generative molecular design, automated analog designer, and synthetic viability prediction”. In: *ACS omega* 7.22 (2022), pp. 18699–18713.

[110] BIOVIA, Dassault Systèmes. *Pipeline Pilot.* Accessed August 09, 2024. 2024. URL: `https://www.3ds.com/products/biovia/pipeline-pilot`.

[111] K. D. Konze, P. H. Bos, M. K. Dahlgren, K. Leswing, I. Tubert-Brohman, A. Bortolato, B. Robbason, R. Abel, and S. Bhat. “Reaction-based enumeration, active learning, and free energy calculations to rapidly explore synthetically tractable chemical space and optimize potency of cyclin-dependent kinase 2 inhibitors”. In: *Journal of Chemical Information and Modeling* 59.9 (2019), pp. 3782–3793.

[112] W. Gao, R. Mercado, and C. W. Coley. “Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design”. In: *arXiv preprint arXiv:2110.06389* (2021).

[113] R. Pophale, F. Daeyaert, and M. W. Deem. “Computational prediction of chemically synthesizable organic structure directing agents for zeolites”. In: *Journal of Materials Chemistry A* 1.23 (2013), pp. 6750–6760.

[114] J. O. Spiegel and J. D. Durrant. “AutoGrow4: an open-source genetic algorithm for de novo drug design and lead optimization”. In: *Journal of Cheminformatics* 12 (2020), pp. 1–16.

[115] M. Hartenfeller, H. Zettl, M. Walter, M. Rupp, F. Reisen, E. Proschak, S. Weggen, H. Stark, and G. Schneider. “DOGS: reaction-driven de novo design of bioactive compounds”. In: *PLoS Computational Biology* 8.2 (2012), e1002380.

[116] H. M. Vinkers, M. R. de Jonge, F. F. Daeyaert, J. Heeres, L. M. Koymans, J. H. van Lenthe, P. J. Lewi, H. Timmerman, K. Van Aken, and P. A. Janssen. “Synopsis: synthesize and optimize system in silico”. In: *Journal of Medicinal Chemistry* 46.13 (2003), pp. 2765–2773.

[117] J. Bradshaw, B. Paige, M. J. Kusner, M. Segler, and J. M. Hernández-Lobato. “Barking up the right tree: an approach to search over molecule synthesis dags”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6852–6866.

[118] J. Guo, A. S. Ibanez-Lopez, H. Gao, V. Quach, C. W. Coley, K. F. Jensen, and R. Barzilay. “Automated chemical reaction extraction from scientific literature”. In: *Journal of Chemical Information and Modeling* 62.9 (2021), pp. 2035–2045.

[119] D. M. Lowe. “Extraction of chemical structures and reactions from the literature”. PhD thesis. 2012.

[120] P. C. Fricker, M. Gastreich, and M. Rarey. “Automated Drawing of Structural Molecular Formulas under Constraints”. In: *Journal of Chemical Information and Modeling* 44.3 (2004), pp. 1065–1078.

[121] A. D. McNaught, A. Wilkinson, et al. *Compendium of chemical terminology.* Vol. 1669. Blackwell Science Oxford, 1997.

[122] R. Schmidt, E. S. Ehmki, F. Ohm, H.-C. Ehrlich, A. Mashychev, and M. Rarey. “Comparing Molecular Patterns Using the Example of SMARTS: Theory and Algorithms”. In: *Journal of Chemical Information and Modeling* 59.6 (2019), pp. 2560–2571.

[123] Center for Bioinformatics, University of Hamburg. *Reaction Viewer Datasets.* Accessed: June 20, 2024. 2022. URL: `https://www.zbh.uni-hamburg.de/forschung/amd/datasets/reaction-viewer-datasets.html`.

[124] K. T. Schomburg, L. Wetzer, and M. Rarey. “Interactive design of generic chemical patterns”. In: *Drug Discovery Today* 18.13-14 (2013), pp. 651–658.

[125] Enamine. *Enamine Building Blocks Global Stock.* Accessed February 02, 2023. URL: https://enamine.net/building-blocks/building-blocks-catalog.

[126] M. J. Munchhof, L. A. Reiter, A. Shavnya, C. S. Jones, Q. Li, and R. G. Linde. *Benzimidazole derivatives.* US Patent US 20090005416 A1, 2009.

[127] J. Eickhoff, G. Zischinsky, and U. Koch. *Pyrazolo-triazine derivatives as selective cyclin-dependent kinase inhinitors.* WO Patent WO 2013128028 A1, 2012.

[128] Schrödinger. Accessed January 09, 2024. URL: https://www.schrodinger.com/platform/products/maestro/.

[129] N. Software. *NameRxn - Expert System for Named Reaction Identification and Classification.* Accessed March 25, 2023.

[130] J. S. Carey, D. Laffan, C. Thomson, and M. T. Williams. "Analysis of the reactions used for the preparation of drug candidate molecules". In: *Organic & Biomolecular Chemistry* 4.12 (2006), pp. 2337–2347.

[131] Y. Y. Syed. "Futibatinib: First Approval". In: *Drugs, Springer* (2022), pp. 1–7.

[132] E. D. Deeks and S. Duggan. "Abrocitinib: first approval". In: *Drugs* 81 (2021), pp. 2149–2157.

[133] OpenAI. *ChatGPT: A large language model based on GPT-4 architecture.* Accessed June 20, 2024. URL: https://www.openai.com/chatgpt.

[134] G. Inc. *Grammarly: AI-Powered Writing Assistant.* Accessed June 20, 2024. URL: https://www.grammarly.com.

[135] DeepL. *DeepL Translate: The world's most accurate translator.* Accessed June 20, 2024. 2024. URL: https://www.deepl.com.

[136] P. Brachet. *TeXmaker: A free, modern, and cross-platform LaTeX editor.* Version 5.1.4, Accessed August 05, 2024. 2024. URL: https://www.xm1math.net/texmaker/.

[137] P. Team. *PlantUML: Open-source tool that allows users to create UML diagrams from plain text descriptions.* Version 1.2024.0, Accessed August 05, 2024. 2024. URL: https://plantuml.com/.

[138] S. Urbaczek, A. Kolodzik, J. R. Fischer, T. Lippert, S. Heuser, I. Groth, T. Schulz-Gasch, and M. Rarey. "NAOMI: on the almost trivial task of reading molecules from different file formats". In: *Journal of Chemical Information and Modeling* 51.12 (2011), pp. 3199–3207.

[139] S. Bietz. “Methoden zur computergestützten Generierung und Aufbereitung von Strukturenensembles für Proteinbindetaschen”. Ph.D. Thesis. Universität Hamburg, 2016.

[140] F. Lauck. “A Computational Framework for Modeling Pharmaceutically Relevant Chemical Space”. Ph.D. Thesis. Universität Hamburg, 2016.

[141] A. V. Aho and J. D. Ullman. *The theory of parsing, translation, and compiling.* Vol. 1. Prentice-Hall Englewood Cliffs, NJ, 1972.

[142] J. Greene, S. Kahn, H. Savoj, P. Sprague, and S. Teig. “Chemical function queries for 3D database search”. In: *Journal of Chemical Information and Computer Sciences* 34.6 (1994), pp. 1297–1308.

Bibliography of this Cumulative Dissertation

[D1] **U. Dolfus**, H. Briem, and M. Rarey. “Visualizing Generic Reaction Patterns”. In: *Journal of Chemical Information and Modeling* 62.19 (2022), pp. 4680–4689.

[D2] **U. Dolfus**, H. Briem, and M. Rarey. “Synthesis-aware generation of structural analogues”. In: *Journal of Chemical Information and Modeling* 62.15 (2022), pp. 3565–3576.

[D3] **U. Dolfus**, H. Briem, T. Gutermuth, and M. Rarey. “Full modification control over retrosynthetic routes for guided optimization of lead structures”. In: *Journal of Chemical Information and Modeling* 63.21 (2023), pp. 6587–6597.

Additional Publications

[E1] C. Meyenburg, **U. Dolfus**, H. Briem, and M. Rarey. “Galileo: Three-dimensional searching in large combinatorial fragment spaces on the example of pharmacophores”. In: *Journal of Computer-Aided Molecular Design* 37.1 (2023), pp. 1–16.

Appendix A

Scientific Contributions

A.1 Contributions to Publications in Scientific Journals

This section lists the authors publications in scientific journals. The contribution of all authors are described. The first three publications are part of this cumulative dissertation [D1–D3]. The last publication [E1] was published during the term of this thesis, but does not contribute to the cumulative dissertation.

[D1] **U. Dolfus**, H. Briem, and M. Rarey. "Visualizing Generic Reaction Patterns". In: *Journal of Chemical Information and Modeling* 62.19 (2022), pp. 4680–4689.

U. Dolfus and M. Rarey developed the algorithm and U. Dolfus implemented the necessary functionalities in the NAOMI code base. The resulting method was integrated by U. Dolfus into an existing software application with a graphical user interface by K.Schomburg [43]. In addition, U. Dolfus integrated the work into the software server SMARTS.plus (`https://smarts.plus/`), performed the comparative analyses and wrote the manuscript. All authors reviewed and approved the final version of the manuscript. H. Briem and M. Rarey provided feedback and supervision during the project.

[D2] **U. Dolfus**, H. Briem, and M. Rarey. "Synthesis-aware generation of structural analogues". In: *Journal of Chemical Information and Modeling* 62.15 (2022), pp. 3565–3576.

All authors contributed to the concepts of the algorithm and the experimental design. U. Dolfus developed the resulting method, designed the required data structures and functionality, integrated the necessary software into the NAOMI code base and implemented the resulting command line tool. The validation of the method and the writing of the manuscript were done by U. Dolfus. All authors have reviewed and approved the final version of the manuscript. H. Briem and M. Rarey provided feedback and supervision during the project.

[D3] **U. Dolfus**, H. Briem, T. Gutermuth, and M. Rarey. “Full modification control over retrosynthetic routes for guided optimization of lead structures”. In: *Journal of Chemical Information and Modeling* 63.21 (2023), pp. 6587–6597.

The algorithms presented in the third publication of this cumulative dissertation were developed by H. Briem, M. Rarey and U. Dolfus. U. Dolfus designed and implemented all necessary data structures and functionalities in the NAOMI code base and created the resulting command line tool. All authors were responsible for the design of the validation. T. Gutermuth was instrumental in the development of the application example for synthetically accessible scaffold hopping and in the selection of the required chemical data. All authors have reviewed and approved the final version of the manuscript. of the manuscript.
H. Briem and M. Rarey provided feedback during the project and supervised the project.

[E1] C. Meyenburg, **U. Dolfus**, H. Briem, and M. Rarey. “Galileo: Three-dimensional searching in large combinatorial fragment spaces on the example of pharmacophores”. In: *Journal of Computer-Aided Molecular Design* 37.1 (2023), pp. 1–16.

This publication describes a method identifying compounds in fragment spaces with an arbitrary scoring function, which opens up the possibility of searching with 3D descriptors, whereas previously only 2D descriptors were available (see Section 1.3, Fragment spaces). A genetic algorithm is used to perform the search. The application of the algorithm is demonstrated using a pharmacophore-based search in a fragment space. C. Meyenburg and M. Rarey developed the genetic algorithm adapted to the search in fragment spaces. C. Meyenburg implemented all necessary data structures and functionalities in the NAOMI code base and created the resulting command line tool. U.

Dolfus developed and implemented a pharmacophore mapping algorithm together with a command line tool for experimental validation. U. Dolfus wrote the text about the pharmacophore mapping algorithm in the manuscript and provided the visualization of example hits in the experiments. C. Meyenburg wrote the rest of the manuscript. H. Briem provided the fragment space used for validation. All authors reviewed and approved the final version of the manuscript. H. Briem and M. Rarey provided feedback and oversight during the project. This publication does not contribute to this cumulative dissertation.

A.2 Conference Contributions

A.2.1 Oral Presentations

The following section lists the author's oral presentations presented at national and international conferences of work contributing to this cumulative dissertation.

A.2.2 Poster Presentations

This section lists poster presentations presented at a national conference of this author, but of work which does not contribute to this cumulative dissertation but was conducted during the term of this thesis.

A.3 Documentation of (AI-based) Tools Used

In the following, all tools used to write this thesis are documented. The use of artificial intelligence (AI) has been limited to research purposes and textual improvements without the addition of further content. The generative AI application ChatGPT [133] was used to summarize publications and create 'BibTex'-references for the bibliography of this thesis. Furthermore, the AI-based tools Grammarly [134] and DeepL Translator/ Write [135] were used to improve word choice and sentence structure. In addition, TexMaker [136], a *Latex* editor, was used to maintain the manuscript and to create the PDF file. PlantUML [137] has been used to create UML diagrams for the implemented data structures of the software applications in this thesis.

Appendix B

Methodical Details

In the following, additional information regarding methodical details of the work presented in Chapters 2 - 4 ([D1–D3]) is presented. This includes the descriptions of the implemented software libraries and algorithms.

B.1 NAOMI

The implementation of all approaches and algorithms described in this paper is based on the NAOMI software library [138]. NAOMI is written in *C++* and provides base classes, functions and algorithms for chem- and bioinformatics. The main functionality that is already available in the library and was used for this thesis is listed below:

1. Parsing and internal representation of molecules [138]

2. Calculation of physicochemical properties and interactions [139] of a molecule

3. Parsing and internal representation of SMILES and SMARTS patterns and their visualization [43]

4. Matching of SMARTS pattern on molecules [122]

5. Calculation of circular and topological fingerprints [101]

6. Parsing and internal representation of Fragment Spaces [140]

The following is a description of additional libraries written to implement the software applications described in this thesis. Background information on the implementation is also given.

B.2 ReactionViewer

This section provides further information on the implementation of the visualization strategy of the SMARTSviewer [43], as well as implementation details of the integration of the ReactionViewer algorithm. In addition, the Section B.2.2 discusses how the SMIRKS [13] rules could be integrated into the current implementation.

B.2.1 Implementation Details

The SMARTSviewer translates the functionality of the SMARTS and SMILES languages into intuitive visualisations using structure diagrams. Atoms are shown as circles with different attributes represented by colours, line types and short statements. At the same time, bonds and configurations are represented by distinct line styles, and an optional legend helps users interpret the patterns. The implementation starts by parsing the SMARTS string into a tree-like structure to extract semantic information . Each atom is represented by a node, which stores all corresponding information (e.g. element information, number of explicit connections, logical expressions). Edges represent bonds. Additional information, given as bond queries, is stored with the edge. By modelling the language as a context-free grammar [141], the parsed information can be checked directly for correct syntax. To parse a generic reaction pattern, a separate graph structure is generated for each SMARTS pattern representing a different component (reactant, product) of the reaction pattern. Each graph contains information about its atoms and bonds, together with its predecessor and successor. In addition, the position of the SMARTS pattern representing the last reactant before the SMARTS pattern representing the first product in the reaction pattern is marked. [43, D1]

In the second step, the parsed SMARTS tree, consisting of potentially multiple SMARTS graphs, is used to check for semantic errors. This can involve, for example, removing impossible logical specifications; 'an atom must be hydrogen AND it must be oxygen'. Another example is the removal of redundant logical specifications, as 'an atom can be any atom and an oxygen', which is the same as the atom must be an oxygen. Finally, a legend with explanations of each unique atom or bond expression is generated, followed by the overall layout and visualization. [43, D1]

B.2.2 Potential Implementation of SMIRKS Rules

As discussed in Chapter 2, it could be useful to integrate the five SMIRKS [13] rules into the ReactionViewer software to additionally check the correctness of the pattern. The SMIRKS rules are cited below:

1. *'The reactant and product sides of the transformation are required to have the same numbers and types of mapped atoms and the atom maps must be pairwise. However, non-mapped atoms may be added or deleted during a transformation.'* [44]

2. *'Stoichiometry is defined to be 1-1 for all atoms in the reactant and product for a transformation. Hence, if non-unit stoichiometry is desired, reactants or products must be repeated.'* [44]

3. *'Explicit hydrogens that are used on one side of a transformation must appear explicitly on the other side of the transformation and must be mapped.'* [44]

4. *'Bond expressions must be valid SMILES (no bond queries allowed).'* [44]

5. *'Atomic expressions may be any valid atomic SMARTS expression for nodes where the bonding (connectivity & bond order) doesn't change. Otherwise, the atomic expressions must be valid SMILES.'* [44]

Rules one to four are already integrated into the Pattern Analyser class of the ReactionSMARTS library. The library was developed for Synthesia and is used to generate product structures based on a generic reaction pattern and reactant structures (see Section B.3.2 for more information). The fifth SMIRKS rule is similar to the second rule in the ReactionSMARTS library. However, the SMIRKS rule is more general and restricts the description of atom expressions not only in the SMARTS pattern describing the product structures, but in all SMARTS patterns of the generic reaction. This could easily be extended in the implementation. To integrate the additional correctness check of the SMIRKS pattern, the pattern analyzer simply needs to be called during the initial parsing phase of the ReactionViewer, where the patterns are checked for correct syntax (see Section B.2.1 and Chapter 2). The integration of all five rules into the current implementation of the ReactionViewer would further support the development of valid SMIRKS patterns.

B.3 Synthesia

The software libraries developed for the Synthesia application and integrated into the NAOMI software code base are described below. These libraries include the main data structures, functionalities and algorithms used in the software applications.

B.3.1 RetroSynTree Library

The RetroSynTree library is the fundamental library for all algorithms implemented in Synthesia. Its core is a hierarchical tree data structure with interconnected nodes. The tree structure represents a retrosynthetic pathway that is parsed and stored as an ordered set of chemical and reaction nodes. The root node represents the lead or target structure. Intermediate nodes alternate between chemical and reaction nodes, while leaf nodes and the root node are entirely of the type chemical. Each node contains information about its direct predecessor (except the root node) and all existing child nodes. The RetroSynTree library contains utilities and convenience functions for the retrosynthetic tree and its nodes. These functions include parsing or writing retrosynthetic routes to or from the JSON file format, which is compatible with common open-source retrosynthetic prediction software or expert input. The expected layout of the JSON file for a retrosynthetic route can be found in the Supporting Information of [D2]. Replacement building blocks can be provided in SMILES, SDF or MOL2 file format or as part of a fragment space in FSDB file format. Regardless of the input format, all building blocks are parsed and converted accordingly into SMILES strings and, if required, into the internal molecular data structures.

Retrosynthetic Tree Nodes

All nodes inherit from the base node data structure. This data structure stores the type of node (chemical, reaction, unknown), an ID value required to store child-parent relationships, and a SMILES [6] string. For chemical nodes, the SMILES string describes the compound in the retrosynthetic route, while for reaction nodes the SMILES string can be empty or a Reaction SMILES string [6]. Each node can optionally have a list of child IDs and a parent ID. The base node data structure contains additional helper and getter/setter functions. Reaction nodes must contain a SMIRKS string in addition to the base node parameters. The SMIRKS pattern must obey additional rules in order to allow unambiguous application and generation of product structures. Further details are described in Section B.3.2. The compound nodes also receive a molecule as a member variable, which is formed from parsed SMILES string. Both reaction and chemical nodes, as well as the retrosynthetic tree, have additional information and functions beyond the base node structure. Figure B.1 shows a UML diagram of the four described data classes of the RetroSynTree library together with their member variables.

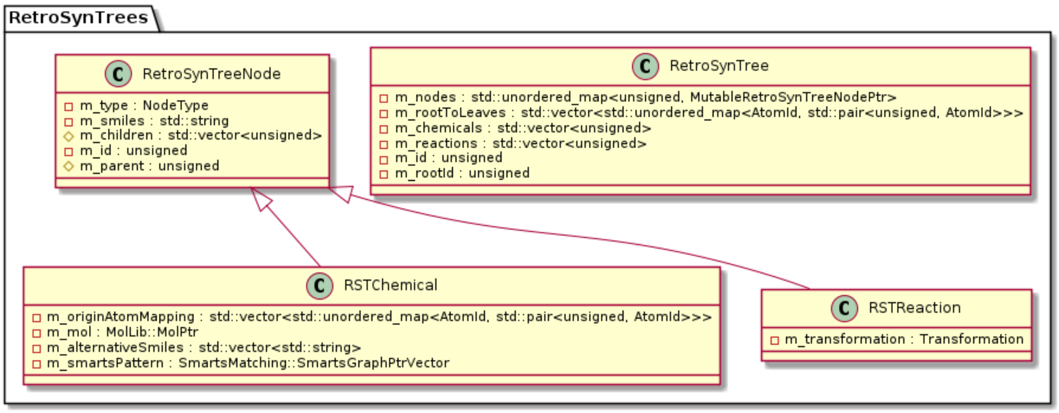

RetroSynTrees
C RetroSynTreeNode
□ m_type : NodeType
□ m_smiles : std::string
◇ m_children : std::vector<unsigned>
□ m_id : unsigned
◇ m_parent : unsigned
C RetroSynTree
□ m_nodes : std::unordered_map<unsigned, MutableRetroSynTreeNodePtr>
□ m_rootToLeaves : std::vector<std::unordered_map<AtomId, std::pair<unsigned, AtomId>>>
□ m_chemicals : std::vector<unsigned>
□ m_reactions : std::vector<unsigned>
□ m_id : unsigned
□ m_rootId : unsigned
C RSTChemical
□ m_originAtomMapping : std::vector<std::unordered_map<AtomId, std::pair<unsigned, AtomId>>>
□ m_mol : MolLib::MolPtr
□ m_alternativeSmiles : std::vector<std::string>
□ m_smartsPattern : SmartsMatching::SmartsGraphPtrVector
C RSTReaction
□ m_transformation : Transformation

Figure B.1: An UML diagram of the four data classes of the RetroSynTree library together with their member variables. Arrows implicate an inheritance relationship. The image was generated with PlantUML [137].

Validity of a Retrosynthetic Tree

The data structure of the retrosynthetic tree has a method for proving validity. This method is used during the initial parsing. For each reaction node, the SMARTS patterns describing the reactant structures in the given reaction pattern are compared with the corresponding compounds of the children of the reaction node. This assigns at least one SMARTS pattern with valid matches to each compound (or several if the assignment is not unique). This assigned information is used to test the validity of the parsed route, i.e. all compounds stored in the chemical nodes of the tree are recreated from leaves to root node, using the SMARTS pattern of the reaction nodes (for more details see Section B.3.2). For each chemical node, a comparison is made to see if the newly created compound is the same as the one stored in the node. Only if this is the case will the algorithm proceed to the next level. Otherwise it is aborted and the retrosynthetic tree is considered invalid. A retrosynthetic tree is valid if the compound in the root node can be reconstructed.

Root to Leaves Atom Mapping

During the validity check described above, an atom mapping is calculated for each chemical node. Each atom in the compound of the chemical node is mapped to an atom in the compounds of the children of the subordinate reaction node, i.e. the reactant compounds. In rare cases, atoms may also be assigned to the reaction node itself. This can happen when the SMIRKS pattern adds atoms to the product that are not part of the reactant structures. With the attached atom assignment of each chemical compound, a unique assignment of all atoms of the lead compound in the root to the responsible reactant atoms can be made. This information is used to automatically determine the location of the modification (see Section 4.1.5).

Exchanger Classes

The RetroSynTree library includes four exchanger classes, each designed to perform specific functions related to the modification of retrosynthetic trees. These classes are briefly described below:

1. The EductExchanger class implements the algorithms described in [D2] for exchanging a single reactant compound in the tree and calculating the resulting modifications. In addition, it includes functionality to skip reaction nodes during the tree traversing algorithm (compare Section 3.1.1 and 4.1.4 or [D2]).

2. The ExhaustiveExchanger class is able to exchange multiple reactant compounds at the same time and calculate the resulting modification to all tree components, especially the lead structure in the root node. The algorithm is outlined in 4.1.2 and described in detail in [D3].

3. The ReactionExchanger class allows the replacement of a reaction node in the retrosynthetic tree with alternative reactions that lead to modifications in a desired direction. Modifications to the tree structure are calculated, expressing the effect of reaction changes on intermediate and final products. The reaction exchange can call an instance of the EductExchanger and the ExhaustiveEductExchanger, depending on whether additional reactant structures are to be exchanged. The different options of reaction exchange together with reactant exchange and their influence on the possible substitution reactions are shown in figure B.2. The algorithm is outlined in 4.1.3 and described in detail in [D3].

4. The ProductExchanger class allows the modification of the compound in the root node, e.g. the lead compound, of the retrosynthetic tree without further specification of where in the tree the modification should take place. A substructure of the

lead structure is defined by a SMARTS pattern and specified as 'replace or modify' or 'keep'. The described atom mapping from root atoms to reactant atoms is used to calculate which nodes should be exchanged to modify the substructure of the lead compound in the root. For all identified atoms, the corresponding nodes are selected. If more than one node is identified, the percentage of the responsible atom per node is calculated and from the highest percentage to the lowest, the nodes are first exchanged individually and then simultaneously. Optionally, the reaction node above the identified responsible nodes can also be exchanged. The algorithm is outlined in 4.1.5 and described in detail in [D3].

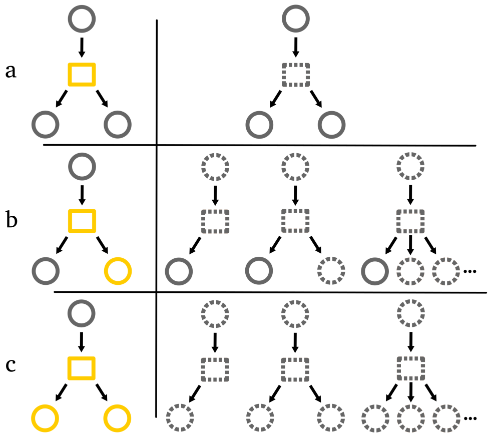


Figure B.2: The image shows an abstract visualization of a reaction node that should be exchanged. On the left side, the initial reaction is shown with the parts chosen for exchange highlighted in yellow. On the right side, the potential substitute reactions are displayed. The exchanged components are indicated with dotted lines. From top to bottom, the following scenarios are illustrated: (a) no reactants are chosen for exchange, (b) some reactants are chosen for exchange, and (c) all reactants are chosen for exchange. The image is extracted from [D3].

In Figure B.3 the four exchanger classes are displayed. The lines between the classes indicate the ability to call an instance of another class during the algorithm. All four classes employ the same fast filtering function to identify available substitute building blocks, utilizing SMARTS matching as the filtering technique. Additionally, each class uses the same function to manage the result queue: if space is available, the current result is added to the queue. When the queue is full, the function automatically removes the worst result, writes it to an output file or the console, dismisses it, and then adds the current result to the queue. The use of a queue enables efficient memory management, as only the most important results are stored and obsolete or less important results are systematically removed.

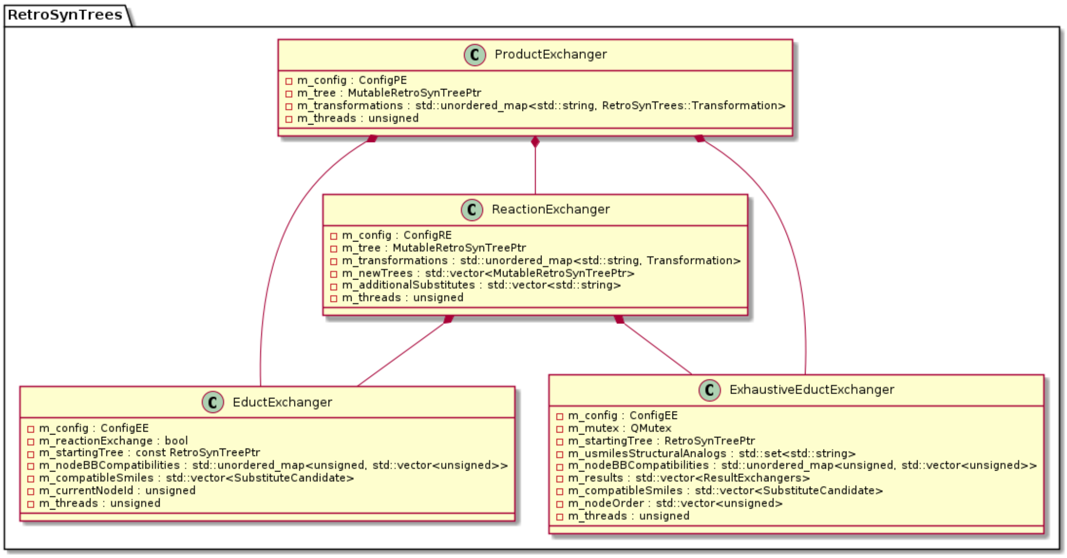

RetroSynTrees
C ProductExchanger
□ m_config : ConfigPE
□ m_tree : MutableRetroSynTreePtr
□ m_transformations : std::unordered_map<std::string, RetroSynTrees::Transformation>
□ m_threads : unsigned
C ReactionExchanger
□ m_config : ConfigRE
□ m_tree : MutableRetroSynTreePtr
□ m_transformations : std::unordered_map<std::string, Transformation>
□ m_newTrees : std::vector<MutableRetroSynTreePtr>
□ m_additionalSubstitutes : std::vector<std::string>
□ m_threads : unsigned
C EductExchanger
□ m_config : ConfigEE
□ m_reactionExchange : bool
□ m_startingTree : const RetroSynTreePtr
□ m_nodeBBCompatibilities : std::unordered_map<unsigned, std::vector<unsigned>>
□ m_compatibleSmiles : std::vector<SubstituteCandidate>
□ m_currentNodeId : unsigned
□ m_threads : unsigned
C ExhaustiveEductExchanger
□ m_config : ConfigEE
□ m_mutex : QMutex
□ m_startingTree : RetroSynTreePtr
□ m_usmilesStructuralAnalogs : std::set<std::string>
□ m_nodeBBCompatibilities : std::unordered_map<unsigned, std::vector<unsigned>>
□ m_results : std::vector<ResultExchangers>
□ m_compatibleSmiles : std::vector<SubstituteCandidate>
□ m_nodeOrder : std::vector<unsigned>
□ m_threads : unsigned

Figure B.3: An UML diagram of the four exchanger classes of the RetroSynTree library together with their member variables. Arrows implicate a 'calls instance during algorithm' relationship. The image was generated with PlantUML [137].

B.3.2 ReactionSMARTS Library

The ReactionSMARTS library is specifically designed to apply generic reaction patterns to reactant compounds to build resulting product compounds. These patterns consist of at least two SMARTS patterns: one that defines the structural requirements for the reactant compound and one that describes the requirements for the product compound. The atom and bond changes of the reaction are described by an atom mapping, which is given in the form of atom labels. The removal or addition of atoms during the transformation can be specifically defined. An example of the application of a generic reaction pattern, visualized as a graphical representation and pattern string, to two reactants, transforming them into a product structure, is shown in Figure B.4.

To ensure conflict-free and unambiguous generation of product compounds, the library expects a SMIRKS pattern (a restricted version of Reaction SMARTS) that follows not only the first four SMIRKS rules (see B.2.2), but also these additional rules:

1. SMARTS patterns describing a product structure must not contain logical operations concerning the element of the atom. Each atom must either have a specific element or be a wildcard.

2. Implicit bonds in the product pattern are treated as single bonds; non-single bonds must be specified explicitly.

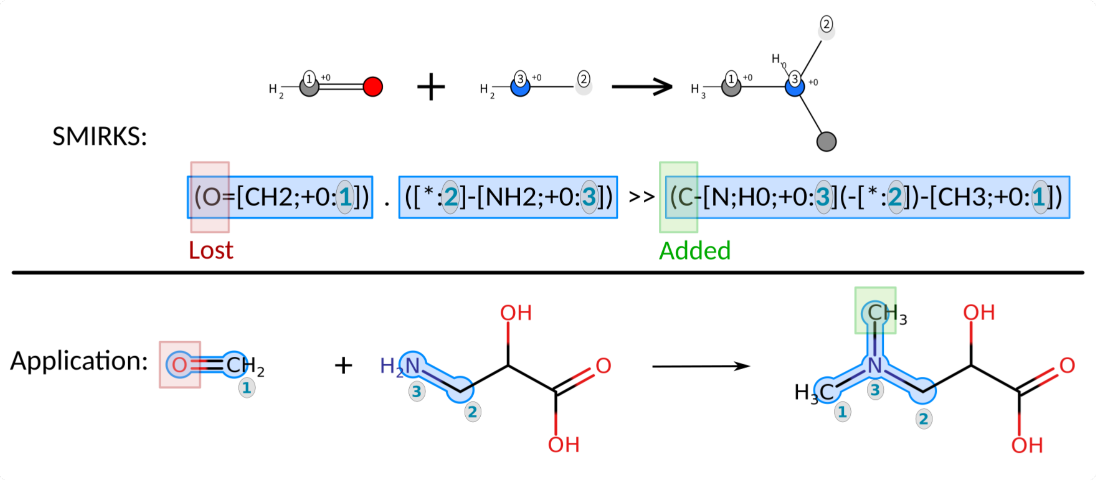

SMIRKS:
(O=[CH2;+0:1]) . ([*:2]-[NH2;+0:3]) >> (C-[N;H0;+0:3](-[*:2])-[CH3;+0:1])
Lost
Added
Application:
OH
O
OH
CH_3
H_3C
N

Figure B.4: A visualization of a SMIRKS pattern as a graphical representation and a pattern string. In the pattern string atoms that are added or removed during the reaction are marked green and respectively red. The application of the SMIRKS pattern onto two reactant structures, transforming them into a product structure, is visualized at the bottom. The image is taken from [D2]

Note that stereo information cannot yet be handled. However, syntheses that produce only one enantiomer are very challenging anyway, so purification methods are often used instead. In addition, stereoselective reaction steps require very specific conditions, often involving special catalysts that are not included in the generic reaction patterns. Therefore, in individual cases where stereochemistry is crucial, chemists need to investigate these steps in more detail.

To create product structures from a generic reaction pattern and reactant structures, the library uses a data structure called MolGraph, which is already included in NAOMI. The difference between the internal representation of a molecule and a MolGraph is that the latter contains only basic information about the given structure, especially about its arrangement, but is adaptable. This means that atoms and bonds can be added, removed and modified, but no additional information, such as implicitly defined hydrogens or aromaticity, is calculated and stored. To include chemically relevant information, the modification process on the MolGraph must be completed, and the MolGraph is converted back into a molecule. During this process, the additional information is calculated and the chemical validity, such as the correct valency, is checked and annotated. To generate product structures from given reactant structures and a generic reaction pattern the following steps are performed:

- Test all SMARTS patterns of the generic reaction (which describe the reactant and product structures) to see whether they fulfill the described rules. If at least one pattern or the combination does not meet all rules, the creation of product structures is aborted.

- Create a MolGraph from the reactant structures. Do not include atoms that match the corresponding SMARTS graph but do not have a label (see Figure B.4 'Lost' atoms). Create bonds between atoms that do not have a label and are not matched by the corresponding SMARTS pattern based on the reactant structure.
- Create a MolGraph from the SMARTS pattern describing the product structure, but only include atoms that do not have a label (see Figure B.4 'Added' atoms). Create bonds between the added atoms based on the SMARTS pattern. If multiple product SMARTS patterns are specified, create multiple MolGraphs.
- Merge the MolGraphs from the reactant structures into the MolGraph from the product structure. Form bonds between the atoms according to the SMARTS pattern of the product structure.
- Create all possible and chemically valid molecules based on all resulting MolGraphs from the product structure and calculate all chemical properties.

The algorithm is simple if only one reactant is to be converted into a product or if the SMARTS patterns describing the reactant structures match only one structure at a time. However, if multiple reactant or product structures are involved, and the SMARTS patterns are written more generically and therefore fit multiple structures, all possible combinations of SMARTS patterns and reactant or product structures have to be tested. This can quickly become combinatorially expensive, and is often the reason why more than one product structure is created although only one product SMARTS pattern is given. Additionally, this can result in product structures that are structurally very different from each other structurally.

B.3.3 Pharmacophore Library

The Pharmacophore Library provides data structures and functionality for mapping and aligning pharmacophore queries to molecules. It employs a graph structure to represent pharmacophores, which enables efficient organization and retrieval of pharmacophore features. The implemented mapping algorithm and data structures are described in [E1]. The library contains a calculation of hydrophobic points based on an algorithm by Greene et al [142]. This algorithm is included as an optional search query constraint in Synthesia to influence the hydrophobicity of the generated structural analogues or the building blocks used.

Appendix C

Software Architecture and Usage

C.1 Software Usage

In the following, the usage of the software applications Synthesia (compare Chapters 3, 4 and [D2, D3]) and ReactionViewer (Chapter 2 and [D3]) are described.

C.1.1 Synthesia Software User Guide

For the use of Synthesia, a command line program was created that allows the creation of synthetically accessible structural analogues based on a synthetic route. The software user guide is listed below after an example call of the software. The user guide includes the description and parameterization options of the command line parameters, information on licensing and error notification.

Example call of Synthesia for the calculation of structural analogues based on a synthetic route in which all start materials are exchanged:

```
./synthesia -r synthesisRoute.json -i buildingBlocks.sdf -o output.json
-allLeaves
```

# Synthesia

Synthesia: a novel approach for synthesis-aware lead optimization. Synthesia preserves a synthetic pathway to the virtual product while providing a variety of computable changes to the compound properties. By exchanging precursor structures in the retrosynthetic pathway, followed by forward synthetic reconstruction optimized analogs are generated. Potential substitutes must fulfill two criteria: they have to be compatible with the retrosynthetic route and must also have the ability to optimize the specified molecular properties in the desired direction. Users can either specify exactly where their retrosynthetic route should be modified and are presented with suitable alternatives, or they specify only the substructure of the target molecule to be modified and let the method automatically determine the responsible subtree, proposing modification options. Furthermore, users can exchange or skip reactions, exchange multiple reactant structures simultaneously, and create a target function that defines wanted or unwanted substructures in the target molecule. Synthesia has an easy to use interface that makes it simple to define your own optimization goals of your lead structure.

## License

Synthesia requires a license. Licenses are free for academic use. You can get a license at: https://software.zbh.uni-hamburg.de/

### Activation

After acquiring a license, you will have to activate Synthesia with that license. To do so, open the license file, copy the content and execute Synthesia as follows:

```
$ ./Synthesia --license <your_license_here>
```

## Retrosynthetic Routes

Synthesia requires the retrosynthetic route(s) of the lead structure to be in JSON format. Each node requires a SMILES object, a specification if it is a reaction or chemical, and a children object (if the node is a leaf this can be empty). An example retrosynthetic route file is bundled with Synthesia.

## Substitute Candidates

Synthesia requires a list with possible substitute candidates. This list can be parsed as an "sdf", "mol", "mol2", "smiles", "smi" or "fsdb" file. In case of a given fragment space (.fsdb), all link-atoms will be terminated before they are considered as substitutes.

## Optimization Goals

Synthesia provides a collection of 29 (structural) properties that can be used to define desired optimization goals. A list of all possible constraint settings is provided below.

## Output

Synthesia will generate a number of optimized compounds together with the modified retrosynthetic route. Note, the modifications of the route are only on the structural level of the chemical nodes, introduced by the substitute, and only accepted if they do not harm the integrity of the route. A list with the basic results will be printed to the console. More detailed results can be set with the parameter --printFullResults 1. All results can be printed to a JSON output file. If more than 1000 hits are generated, the best 1000 hits are printed last. All other hits will be printed before in any order.

## Configuration file

All additional settings of Synthesia can be specified in a configuration file. This file is optional and the user does not have to use it. If both the configuration file as well as command line parameters are used to define parameters, the settings parsed via command line overwrite settings defined in the configuration file. The configuration file has to be in valid standard JSON format. An example configuration file is bundled with Synthesia.

### Possible Configurations

```
/* -- General Options -- */
| Configuration               | Value Type | Explanation
|-----------------------------|------------|-------------------------------------
|`-h [--help]`                |            | Print help message.
|`-t [--threads]`             | Integer    | Number of threads used for
|                             |            | parallelization.
|`-v [--verbosity]`           | Integer    | Verbosity level.
|`--visualizeTrees`           | Boolean    | Print given retrosynthetic tree to
|                             |            | command line.
|                             |            | Routine won't start.
|                             |            | Expected: --visualizeTrees
|`--printFullResults`         | Boolean    | Printed results will contain
|                             |            | representations of all new
|                             |            | trees. Expected: --printFullResults

/* -- Required Options -- */
| Configuration               | Value Type | Explanation
|-----------------------------|------------|-------------------------------------
|`-i [--inputStructures]`     | String     | Path to a file with possible
|                             |            | substitute candidates.
|                             |            | Allowed file extensions are ".sdf",
|                             |            | ".mol", ".mol2", ".smiles",
|                             |            | ".smi", ".fsdb". If a
|                             |            | fragment space (.fsdb)
|                             |            | is specified, all fragments are
|                             |            | terminated before being
|                             |            | considered as substitutes.
|`-r [--retroSynTree(s)]`     | String     | Path to a file with the
|                             |            | retrosynthetic route. Expected
|                             |            | file extension: ".json". An
|                             |            | example tree file is
|                             |            | bundled with Synthesia.

/* -- Configuration Options -- */
| Configuration               | Value Type | Explanation
|-----------------------------|------------|-------------------------------------
|`-c [--config]`              | String     | Path to a configuration file,
|                             |            | where all following configuration
|                             |            | options can be set. All values
|                             |            | from the configuration file can be
|                              |            | overwritten by parameters set
|                              |            | during the program call.
|`-o [--output]`               | String     | Path to an output file.
|                              |            | Expected file extension: ".json"
|`--transformations`           | String     | Path to a transformation file. If
|                              |            | the retrosynthetic tree does not
|                              |            | contain SMIRKS patterns for the
|                              |            | reaction nodes, these can be
|                              |            | parsed in an additional
|                              |            | transformation file. The
|                              |            | file must contain a SMIRKS pattern
|                              |            | along with a numeric identifier
|                              |            | (tf-id). The identifier must be
|                              |            | parsed with the corresponding
|                              |            | reaction node so that a unique
|                              |            | assignment is possible.
|                              |            | Expected file extension: ".csv" or
|                              |            | ".txt".
|`--treeId`                    | Integer    | If more than one retrosynthetic
|                              |            | route is stored in the specified
|                              |            | input file (--retroSynTree(s)), you
|                              |            | can use this parameter to specify
|                              |            | which tree to use for the
|                              |            | routine. The first tree in the
|                              |            | file has id 1. If no id is
|                              |            | specified but more than one
|                              |            | tree is given, the first
|                              |            | tree will be chosen automatically.
|                              |            | Multiple ids allowed, need to be
|                              |            | parsed seperated with a space.

/* -- Reaction Exchanger configuration options */
| Configuration                | Value Type | Explanation
|------------------------------|------------|------------------------------------
|`--reactionId`                | Unsigned   | This parameter can be used to
|                              |            | specify which reaction node is open
|                              |            | for exchange.
|`--rLevel`                    | String     | This parameter specifies on which
|                              |            | level the reaction exchange should
|                              |            | occur. Options are 0 =
|                              |            | nameExchange, 1 = superClass, 2 =
|                              |            | commonClass, 3 = specificClass,
|                              |            | 4 = None.


/* -- Exhaustive Exchanger configuration options */
| Configuration                | Value Type | Explanation
|------------------------------|------------|------------------------------------
|`--exchangeSim`               | Bool       | Defines if all specified nodes
|                              |            | should be exchanged simultaneously.


/* -- Product Exchanger configuration options */
| Configuration                | Value Type | Explanation
|------------------------------|------------|------------------------------------
|`--smartsProduct`             | String     | This parameter can be used to
|                              |            | specify which substructure
|                              |            | of the product structure should be
|                              |            | either kept or exchanged. The
|                              |            | string has to be a smarts pattern,
|                             |            | which must match uniquely on a
|                             |            | substructure of the product. If
|                             |            | this parameter is set, the PE
|                             |            | routine is started otherwise the EE
|                             |            | routine is used.
|`--productExchangeType`      | String     | This parameter specifies if the
|                             |            | substructure specified with the
|                             |            | matching parsed smartsProduct
|                             |            | pattern should be excluded
|                             |            | (exchanged) or included (kept,
|                             |            | rest of structure open for
|                             |            | exclusion). Options are 1 =
|                             |            | inclusion, 2 = exclusion.

/* -- Educt Exchanger configuration options -- */
| Configuration               | Value Type | Explanation
|-----------------------------|------------|------------------------------------
|`--nodeId`                   | Integer(s) | This parameter can be used to
|                             |            | specify which chemical node should
|                             |            | be open for exchange to introduce
|                             |            | structural modifications. Either
|                             |            | this parameter must be specified or
|                             |            | the option --allLeaves or
|                             |            | --allChemicals must be set. To get
|                             |            | all nodeIds of the given
|                             |            | retrosynthetic route, use --
|                             |            | visualizeTrees 1. Multiple
|                             |            | ids allowed, need to be parsed
|                             |            | seperated with a space.
|`--allLeaves`                | Boolen     | Set this parameter if all
|                             |            | chemical leaf nodes should be open
|                             |            | for exchange. Either this
|                             |            | parameter or the option
|                             |            | allChemicals must be set or the
|                             |            | nodeId parameter must be specified.
|                             |            | Expected: --allLeaves
|`--allChemicals`             | Boolean    | Set this parameter if all
|                             |            | chemical nodes should be open for
|                             |            | exchange. Note, for intermediate
|                             |            | structures the retrosynthetic route
|                             |            | compatibility is only guaranteed in
|                             |            | the direction of the root up the
|                             |            | tree. Either this parameter or the
|                             |            | option allChemicals must be set or
|                             |            | the nodeId parameter must be
|                             |            | specified. Expected: --allChemicals
|`--nofMinMatchs`             | Integer    | Can be used to specify the number
|                             |            | of additional search query
|                             |            | constraints which must be
|                             |            | fulfilled. By default, this number
|                             |            | is equal to the number of given
|                             |            | search query constraints.
|`--searchQueryApplication`   | Integer    | Specifies whether only the
|                             |            | substitute structure (0,
|                             |            | default) or only the modified
|                             |            | product structure (1) or
|                             |            | both (2) have to fulfill the
|                             |            | defined search query
|                             |            | constraints. Value has to be in
|                             |            | range [0,2].
```

`--deviationOptimization`	String	Specifies whether the calculated deviation from the reference structure value should be maximized (maximum = 0), or minimized = 1 (minimum = 1) for the sorting of the results.
`--useECFP`	Multitoken	This parameter can be used to add the Extended-Connectivity Fingerprint (ECFP) as a additional search query constraint. 4 parameter values are expected: **<Integer> <String> <Integer> <Integer>** The first number equals the appended number of the ECFP and thereby is the effective diameter of the largest feature. It is equal to twice the number of iterations performed. The second string parameters specifies the similarity measure method for a fingerprint comparison. Options are 'tanimoto', 'cosine', ' hamming', 'euclidean', 'dice.' The third number specifies the minimum threshold value for the similarity fingerprint comparison and the fourth number specifies the maximum threshold value. The following example parametrization: `--useECFP 4 tanimoto 0.6 1.0` equals a ECFP_4 constraint with a tanimoto coefficient comparison and an allowed range between 0.6 and 1.
`--useFCFP`	Multitoken	This parameter can be used to add the Functional-Class Fingerprint (FCFP) as a additional search query constraint. 4 parameter values are expected: **<Integer> <String> <Integer> <Integer>** The first number equals the appended number of the FCFP.The second string parameters specifies the similarity measure method for a fingerprint comparison. Options are 'tanimoto', 'cosine', ' hamming', 'euclidean', 'dice.' The third number specifies the minimum threshold value for the similarity fingerprint comparison and the fourth number specifies the maximum threshold value. The following example parametrization: `--useFCFP 4 tanimoto 0.6 1.0` equals a FCFP_4 constraint with a tanimoto coefficient comparison and

```
|                              |            | an allowed range between 0.6 and 1.
|`--useCSFP`                   | Multitoken | This parameter can be used to add
|                              |            | the Connected-Subgraph Fingerprint
|                              |            | (CSFP) as a additional search
|                              |            | query constraint. 6 parameter
|                              |            | values are expected:
|                              |            | <String> <String> <Integer>
|                              |            | <Integer> <Integer> <Integer>
|                              |            | The first string defines which
|                              |            | CSFP type should be used.
|                              |            | Options are 'csfp', 'icsfp', '
|                              |            | gcsfp', 'tcsfp', 'fcsfp'.
|                              |            | The second string parameters
|                              |            | specifies the similarity
|                              |            | measure method for a fingerprint
|                              |            | comparison. Options are
|                              |            | 'tanimoto', 'cosine', ' hamming',
|                              |            | 'euclidean', 'dice.' The third
|                              |            | number specifies the minimum
|                              |            | threshold value for the similarity
|                              |            | fingerprint comparison and the
|                              |            | fourthnumber specifies the maximum
|                              |            | threshold value. The fifth integer
|                              |            | sets the lower bound for the csfp
|                              |            | subgraph size and the sixth the
|                              |            | upper bound. The following example
|                              |            | parametrizati on:
|                              |            | `--useCSFP icsfp tanimoto 0.6 1.0
|                              |            | 2 5` equals a icsfp constraint
|                              |            | with a tanimoto coefficient
|                              |            | comparison, an allowed range
|                              |            | between 0.6 and 1 and a
|                              |            | subgraph size between 2 and 5.
|`--useSmartsFilter`           | Multitoken | This parameter can be used to add
|                              |            | SMARTS pattern as additional search
|                              |            | query constraint. It can be defined
|                              |            | if the SMARTS pattern either have
|                              |            | to be included in the structural
|                              |            | modifications or excluded. 2
|                              |            | parameter values are expected:
|                              |            | <string> <string>
|                              |            | The first string is either a valid
|                              |            | SMARTS pattern or a path to a
|                              |            | ".smi" file, which contains
|                              |            | multiple SMARTS pattern. The second
|                              |            | string has to set the type of
|                              |            | matching, options are 'exclusion'
|                              |            | or 'inclusion'. Example
|                              |            | parametrization:
|                              |            | `--useSmartsFilter '[#7;!R]=[#7]'
|                              |            | exclusion
|`--useLargestRing`            | Multitoken | This parameter can be used to add
|                              |            | the number of atoms of the largest
|                              |            | ring as additional search query
|                              |            | constraint.
|                              |            | 2 parameter values are expected:
|                              |            | <string> <integer>
|                              |            | The first string defines how the
|                              |            | filter should be applied. The
|                              |            | following options are available:
```

```
|                             |            | - `Exact_Value` Calculated value
|                             |            |   of the substitute candidate has
|                             |            |   to be equal to a numeric value.
|                             |            |   Example parametrization:
|                             |            |   `--useLargestRing
|                             |            |   Exact_Value 6`
|                             |            |   The largest ring must have
|                             |            |   exactly 6 heavy atoms.
|                             |            | - `UpperBound_Value` Calculated
|                             |            |   value of the substitute
|                             |            |   candidate has to be equal or
|                             |            |   smaller than a numeric value.
|                             |            |   Example parametrization:
|                             |            |   `--useLargestRing
|                             |            |   UpperBound_Value 6`
|                             |            |   The largest ring must not
|                             |            |   exceed 6 heavy atoms.
|                             |            | - `LowerBound_Value` Calculated
|                             |            |   value of the substitute
|                             |            |   candidate has to be equal or
|                             |            |   larger than a numeric value.
|                             |            |   Example parametrization:
|                             |            |   `--useLargestRing LowerBound_
|                             |            |   Value 6`
|                             |            |   The largest ring must have at
|                             |            |   least 6 heavy atoms.
|                             |            | - `Exact_RefMolecule` Calculated
|                             |            |   value of the substitute
|                             |            |   candidate has to be equal to
|                             |            |   the value of the original
|                             |            |   structure in the node.
|                             |            |   Note, the numeric value has no
|                             |            |   effect in this setting.
|                             |            |   Example parametrization:
|                             |            |   `--useLargestRing
|                             |            |   Exact_RefMolecule 0`
|                             |            |   The largest ring must have
|                             |            |   exactly the same number
|                             |            |   of heavy atoms as the largest
|                             |            |   ring in the original structure.
|                             |            | - `Threshold_RefMolecule`
|                             |            |   Calculated value of the
|                             |            |   substitute candidate must be
|                             |            |   above or below the
|                             |            |   value of the original structure
|                             |            |   in the node plus or
|                             |            |   minus a numeric value.
|                             |            |   Example parametrization:
|                             |            |   `--useLargestRing Threshold_
|                             |            |   RefMolecule -2`
|                             |            |   The largest ring must have at
|                             |            |   least two heavy atoms less than
|                             |            |   the largest ring of the
|                             |            |   original structure.
|                             |            | - `Range_RefMolecule` Calculated
|                             |            |   value of the substitute
|                             |            |   candidate must be in a range of
|                             |            |   a numeric value around the value
|                             |            |   of the original structure node.
|                             |            |   Example parametrization:
```

		`--useLargestRing Range_RefMolecule 3` The number of the heavy atoms of the largest ring of the substitute candidate must be in a range of [-3,+3] around the number of heavy atoms of the largest ring of the original structure. The second integer value sets the numeric value for the comparison.
`--useLargestRingsystem`	Multitoken	This parameter can be used to add the number of atoms of the largest Ringsystem as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useMolecularWeight`	Multitoken	This parameter can be used to add the MolecularWeight as additional search query constraint (g/mol). For detailed about usage information see `--useLargestRing`.
`--useNofAcceptors`	Multitoken	This parameter can be used to add the number of acceptors as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofAnions`	Multitoken	This parameter can be used to add the number of anions as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofAromaticAtoms`	Multitoken	This parameter can be used to add the number of aromatic atoms as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofAromaticRings`	Multitoken	This parameter can be used to add the number of aromatic rings as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofAromaticRingsystems`	Multitoken	This parameter can be used to add the number of aromatic ringsystems as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofCations`	Multitoken	This parameter can be used to add the number of cations as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofDonors`	Multitoken	This parameter can be used to add the number of donors as additional search query constraint. For detailed about usage information

		see `--useLargestRing`.
`--useNofHalogens`	Multitoken	This parameter can be used to add the number of halogens as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofHeavyAtoms`	Multitoken	This parameter can be used to add the number of heavy atoms (non-hydrogen) as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofHydrophobicPoints`	Multitoken	This parameter can be used to add the number of hydrophobic points as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofInorganicAtoms`	Multitoken	This parameter can be used to add the number of inorganic atoms as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofLipinskiDonors`	Multitoken	This parameter can be used to add the number of lipinski donors as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofNitrogensAndOxygens`	Multitoken	This parameter can be used to add the number of nitrogens and oxygens as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofNonHydrogenBonds`	Multitoken	This parameter can be used to add the number of non-hydrogen bonds as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofRings`	Multitoken	This parameter can be used to add the number of rings as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofRingsystems`	Multitoken	This parameter can be used to add the number of ringsystems as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useNofRotatableBonds`	Multitoken	This parameter can be used to add the number of rotatable bonds as additional search query constraint. For detailed about usage information see `--useLargestRing`.
`--useLogP`	Multitoken	This parameter can be used to add

```
|                              |            | the LogP value as additional search
|                              |            | query constraint. For detailed
|                              |            | about usage information see
|                              |            | `--useLargestRing`.
|`--useTotalCharge`            | Multitoken | This parameter can be used to add
|                              |            | the total charge as additional
|                              |            | search query constraint. For
|                              |            | detailed about usage information
|                              |            | see `--useLargestRing`.
|`--useTpsa`                   | Multitoken | This parameter can be used to add
|                              |            | the Topological polar surface area
|                              |            | (Tpsa) as additional search query
|                              |            | constraint. For detailed
|                              |            | about usage information see
|                              |            | `--useLargestRing`.
|`--useVolume`                 | Multitoken | This parameter can be used to add
|                              |            | the volume as additional search
|                              |            | query constraint. For detailed
|                              |            | about usage information see
|                              |            | `--useLargestRing`.

If you want to specify a negative value as property, this needs to be parsed
with a '%' character. For a -1.0 value the input needs to be %-1.0.

/* -- License -- */
| Configuration                | Value Type | Explanation
|------------------------------|------------|------------------------------------
|`--license`                   | String     | License key for Synthesia. To
|                              |            | reactivate the executable, please
|                              |            | provide a new license key.


## Error Reporting

If you want to report a problem with Synthesia, please provide as much
information as possible. An error report should at the very least contain a
short description of the problem, detailed reproduction steps and your
configuration file/ command line parameters.
```

C.1.2 ReactionViewer User Guide

A command line program was created for the use of ReactionViewer and the algorithm was integrated into an interactive graphical user interface and a software server. In the following, the user guidance for the command line program is explained after a sample call of the software has been shown. Snapshots of the graphical user interfaces and the software server are given in the following chapters. Example call of ReactionViewer for the visualization of an example reaction pattern:

```
./reactionviewer -s '[C;H1&$(C([#6])[#6]),H2&$(C[#6]):1][OH1].[NH1;$(N([#6
])S(=O) =O):2]»[C:1][N:2]'
```

```
# ReactionViewer

ReactionViewer is a tool for the automatic generation of visualizations of
generic reaction patterns. It supports the visualization of Reaction SMILES,
Reaction SMARTS and SMIRKS, which makes it versatile for different types of
chemical reaction data. Simple SMILES or SMARTS pattern can be visualized, too.
In addition, ReactionViewer offers the possibility to display explanations of
individual components and provides various customizable parameters to adapt the
visualizations to your specific needs. Detailed descriptions of these options
and settings can be found below.

## License

ReactionViewer requires a license. Licenses are free for academic use. You can
get a license at: https://software.zbh.uni-hamburg.de/


### Activation

After acquiring a license, you will have to activate ReactionViewer with that
license. To do so, open the license file, copy the content and execute
ReactionViewer as follows:

    $ ./ReactionViewer --license <your_license_here>


### Possible Configurations

| Configuration               | Value Type | Explanation
|-----------------------------|------------|------------------------------------
|`-h`                         |            | Print help message.
|`-s` <smarts>                | String     | The input smarts for
|                             |            | visualization. Can be either a
|                             |            | SMILES, SMARTS, Reaction SMILES,
|                             |            | Reaction SMARTS or SMIRKS pattern.
|                             |            | Either -s or -f have to be given.
|`-f` <file>                  | String     | A file containing the smarts for
|                             |            | visualization. Can be multiple
|                             |            | patterns, but have to be a SMILES,
|                             |            | SMARTS, Reaction SMILES, Reaction
|                             |            | SMARTS or SMIRKS patterns. Either
|                             |            | -s or -f have to be given.
|`-o` <outfile>               | String     | Prints the diagram to
|                             |            | <outfile> possible file formats:
|                             |            | .pdf, .ps, .svg
|`-d` <w> <h>                 | Multitoken | Dimension of the  .svg output file.
|                             |            | (100 <= w|h <= 1000)
|`-p`                         | Multitoken | Eight values have to be given,
|                             |            | range and defaults:
|                             |            | 1. Display options: 0-3 <0>
|                             |            |    (0=Complete Visualization, 1=
|                             |            |    IDs, 2= Element symbols,
|                             |            |    3=Structure Diagram-like)
|                             |            | 2. Default bond options: 0-1
|                             |            |    <0> (0=Single bond, 1=Single or
|                             |            |    aromatic bond
|                             |            | 3. Show Userlabels?: 0-1 <0>
|                             |            |    (0=No, 1=Yes)
|                             |            | 4. Trim-errorcheck?: 0-1 <0>
|                             |            |    (0=Yes, 1=No)
```

		5. Trim-simplification?: 0-1 <0> (0=Yes, 1=No) 6. Trim-interpretation?: 0-1 <0> (0=Yes, 1=No) 7. Show Legend?: 0-3 <0> (0=No, 1=Dynamic legend, 2=Static Legend 3=Both) 8. Print SMARTS string into picture?: 0-1 <0> (0=YES, 1=NO)
`--license`	String	License key for ReactionViewer. To reactivate the executable, please provide a new license key.

## Error Reporting

If you want to report a problem with ReactionViewer, please provide as much information as possible. An error report should at the very least contain a short description of the problem, detailed reproduction steps and your command line parameters.

Appendix D

Publications of the Cumulative Dissertation

D.1 Visualizing Generic Reaction Patterns

[D1] **U. Dolfus**, H. Briem, and M. Rarey. “Visualizing Generic Reaction Patterns”. In: *Journal of Chemical Information and Modeling* 62.19 (2022), pp. 4680–4689.

The following pages include the published manuscript. Due to the length of the Supporting Information, the corresponding pages are not included in this document. They can be found here `https://doi.org/10.1021/acs.jcim.2c00992`. The Supporting Information includes the visualization of the complete data set provided by Hartenfeller et al. [26] generated by the ReactionViewer. The visualization of the AiZynthFinder [19] reaction template data set, consisting of 46696 reaction schemes, is available here `https://www.zbh.uni-hamburg.de/forschung/amd/datasets/reaction-viewer-datasets.html`.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

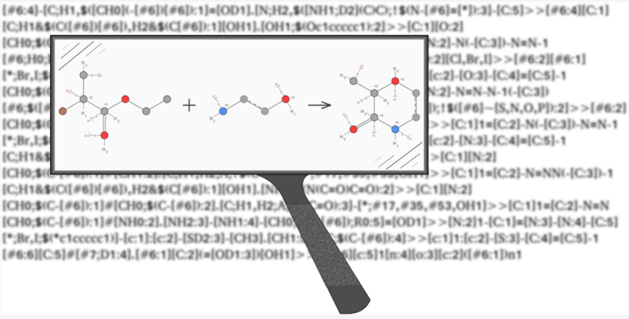
Visualizing Generic Reaction Patterns

Uschi Dolfus, Hans Briem, and Matthias Rarey*

Cite This: *J. Chem. Inf. Model.* 2022, 62, 4680−4689

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information



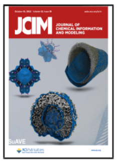
ABSTRACT: Reaction schemes for organic molecules play a crucial role in modern in silico drug design processes. In contrast to the classical drawn reaction diagrams, computational chemists prefer SMARTS based line notations due to a substantially increased expressiveness and precision. They are used to search databases, calculate synthesizability, generate new molecules, or simulate novel reactions. Working with computer-readable representations of reaction schemes can be challenging due to the complexity of the features to be represented. Line representations of reaction schemes can often be cryptic, even to experienced users. To simplify the work with Reaction SMARTS for synthetic, computational, and medicinal chemists, we introduce a visualization technique for reaction schemes and provide a respective tool, called ReactionViewer. ReactionViewer is able to convert reaction schemes encoded as Reaction SMILES, Reaction SMARTS, or SMIRKS into a visual representation. The visualization technique is based on the concept of structure diagrams and follows IUPAC's "Compendium of Chemical Terminology" definition of chemical reaction equations for the reaction symbols. We demonstrate the applicability of the method using two data sets of organic synthesis reaction schemes taken from recent publications. We discuss various properties of the visualization and highlight its readability and interpretability.

■ INTRODUCTION

Chemical pattern languages belong to the most central foundations of cheminformatics. In a time when these methods are playing an increasingly important role in the cost-effective development of therapeutically relevant molecules, they should be easily understandable and accessible to a wide community of chemists involved. Chemical patterns are used to organize and analyze chemical data. For example, they are used to query databases, to apply filters, and to describe specific structural features or even whole reaction schemes. The latter is increasingly coming into focus due to the growing number of computer-assisted synthesis planning (CASP) techniques.[1−6] Developed with the rise of machine learning in chemistry, they rely on computer-interpretable representations of generic reaction schemes. These techniques are essential to guaranteeing the synthetic accessibility of in silico designed molecules. However, generic reaction schemes are still rarely found in organic synthesis literature even though they are extremely valuable. They are not only used for synthesizability prediction but also used as models to create new reactions or are applied to modify or create novel molecules.[7] To gain generic reaction schemes, experimentally validated reactions are often extracted and translated from the literature by semiautomatic procedures. This process usually requires a manual control by a synthetic chemist to guarantee a correct and meaningful translation. All important details regarding the reaction center must be included, while disposable or overly specific information must be disregarded. Synthetic chemists have few tools at their disposal to assist them in this process.

The community standard for expressing chemical patterns is the Daylight Information Systems' SMARTS (SMiles ARbitrary Target Specification) language.[8] SMARTS is an extension of the SMILES (Simplified Molecular Input Line Entry System) language for molecules widely used in computational chemistry.[9] SMILES represent chemical structures in a highly compact, still human readable text form. SMARTS adds a level of abstraction and is used to describe general chemical patterns. Both languages are written as line notation. SMARTS generalizes the concept of SMILES but retains all of its original elements. Extensions introduced with SMARTS are, for example, logical expressions or the possibility to specify atomic or bond properties. Thereby, SMARTS allows chemists to design highly detailed descriptions of their pattern of choice without needing to list all specific structures explicitly. Both languages can be used to express reaction information. Reaction SMILES and Reaction SMARTS are, like their underlying languages, able to describe either explicit reactions or more

Received: August 3, 2022
Published: September 28, 2022



ACS Publications

https://doi.org/10.1021/acs.jcim.2c00992

abstract reaction patterns. SMIRKS is the third language provided by Daylight Information Systems especially designed to describe generic reactions.[7] It is a restricted version of the Reaction SMARTS pattern. SMIRKS combines the power of the SMILES and SMARTS language to express not only the reaction graph but also indirect effects of a reaction in the structures. It allows a detailed description of the reaction center but is still, like all the Daylight languages, editable with a simple text editor. SMIRKS is one of the few human and computer readable languages for reactions. All three languages are established in the in silico drug development process and are supported by various software suites and programming toolkits for molecular modeling. The development of these languages focused on effective computational processing rather than human readability and interpretability. SMARTS or SMIRKS patterns are a combination of regular expressions, element abbreviation, and further symbols, which results in a complex and hardly readable expression. Even scientists familiar with the languages need time and skill to interpret these patterns or spot errors, let alone create them. This handicaps the well needed development of generic reaction schemes.

A logical step to build a bridge between the abstract and hard-to-read patterns and the human reader without losing the computer readability of the patterns is their automated visualization. ChemAxon[10] provides a molecule visualizer, MarvinSketch,[11,12] which is able to parse and handle reaction patterns. However, the visualization is not specialized on the SMARTS language. The reaction pattern are displayed as structural diagrams, but additional information regarding for example atom and bond properties or recursive patterns are only added in textual form. This way, the full power of the SMARTS language is not adequately represented, which results in compromised readability. Here, we introduce a graphical language and a respective tool named ReactionViewer, an extension of the SMARTSviewer[13] software, a handy tool for visualizing reaction schemes written in Reaction SMARTS or SMIRKS. ReactionViewer automatically converts complex reaction schemes into easy-to-understand images, following the general model of structure diagrams. It uses colors and shapes to present an intuitive representation of all given information, as well as an optional natural language explanation in a legend. We present the utility of ReactionViewer by visualizing two different data sets of reaction schemes used in recent CADD methods.[2,14] In addition, we compare the visualization technique of ReactionViewer with a common classical visualization technique used in the publication by Hartenfeller et al.[14]

■ METHODS

In the following, we describe the algorithm used for the conversion process of chemical reaction patterns given in textual form into a graphical representation of these patterns. The concept was derived from an existing approach named SMARTSviewer.[13] SMARTSviewer converts SMARTS or SMILES expressions into visual representations. We give a brief summary of the existing SMARTSviewer methodology and describe in detail the adjustments that have been made to allow not only single SMARTS patterns but also reaction scheme conversions. The structure in which the reaction schemes must be given to be convertible is described. Additional supported features of the methodology are highlighted as well as its current limitations.

Concept and Implementation of the SMARTSviewer. SMARTSviewer provides a visual representation of a SMARTS pattern and is able to covert the complete range of features of the language. To provide chemists with an easy-to-understand and intuitive visualization of chemical structures, the visualization concept is based on structure diagrams. Atoms are drawn as circles. Elements are represented either by color or by element letters in the circle. Additional atomic properties are visualized, e.g., by the type of line (aliphaticity, aromaticity), by visual representation in the atom circle (atomic mass, valence), or by short indications near the corresponding atom (charge, explicit hydrogen atoms). Bonds are represented by either one, two, or three lines for single, double, or triple bonds. The "any" SMARTS specification of a bond is visualized by color. Cis/trans configurations are taken into account when calculating coordinates of atoms. Logical operators (AND, OR, NOT) are all visualized by color codes. Recursive specifications are shown as independent graphs next to the corresponding atom. The SMARTSviewer provides the user with an (optional) legend with explanations of all displayed features to facilitate understanding and overview of the visualized pattern.[13]

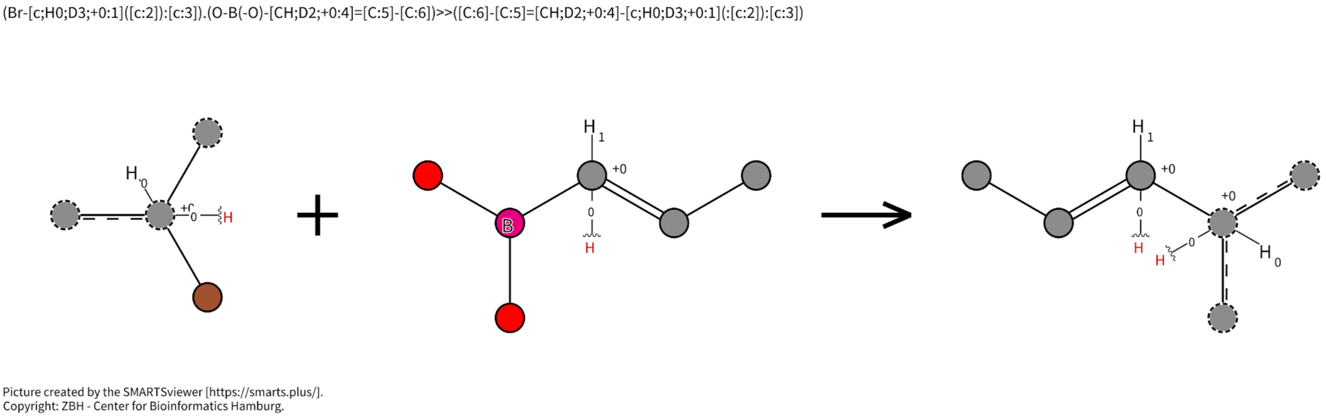
The implementation consists of three steps: parsing the SMARTS string to get all the semantic information, processing the information and computing the corresponding internal objects, and finally drawing the actual image. During the first two phases, the SMARTS string is checked for semantic errors, and redundant information is removed. The calculation of the coordinates follows the same principle as with a complete molecule and thereby can be solved with structure diagram generation methods.[15−17] All additional information is placed around the calculated structure avoiding clashes. A more detailed description can be found in the respective publication.[13]

Reaction SMILES, Reaction SMARTS, and the SMIRKS Language. Reaction schemes can be written as SMIRKS, Reaction SMARTS, or Reaction SMILES. Reaction SMILES represent explicit specified molecules in a reaction and are therefore closest to the traditional representation of reactions. Reaction SMARTS and SMIRKS patterns represent generic reaction schemes. The SMIRKS language is a restricted version of Reaction SMARTS. It adds explicit atom and bond change descriptions to the pattern.[7] Reaction patterns written in any of the three languages can be parsed by ReactionViewer if the pattern has one of the following formats:

1. Forward-synthetic direction:
   $reactant_1 \,.\, \ldots \,.\, reactant_i >> product_1 \,.\, \ldots \,.\, product_j$
2. Retrosynthetic direction:
   $product_1 \,.\, \ldots \,.\, product_j >> reactant_1 \,.\, \ldots \,.\, reactant_i$

with $1 \leq i, j \in \mathbb{N}$

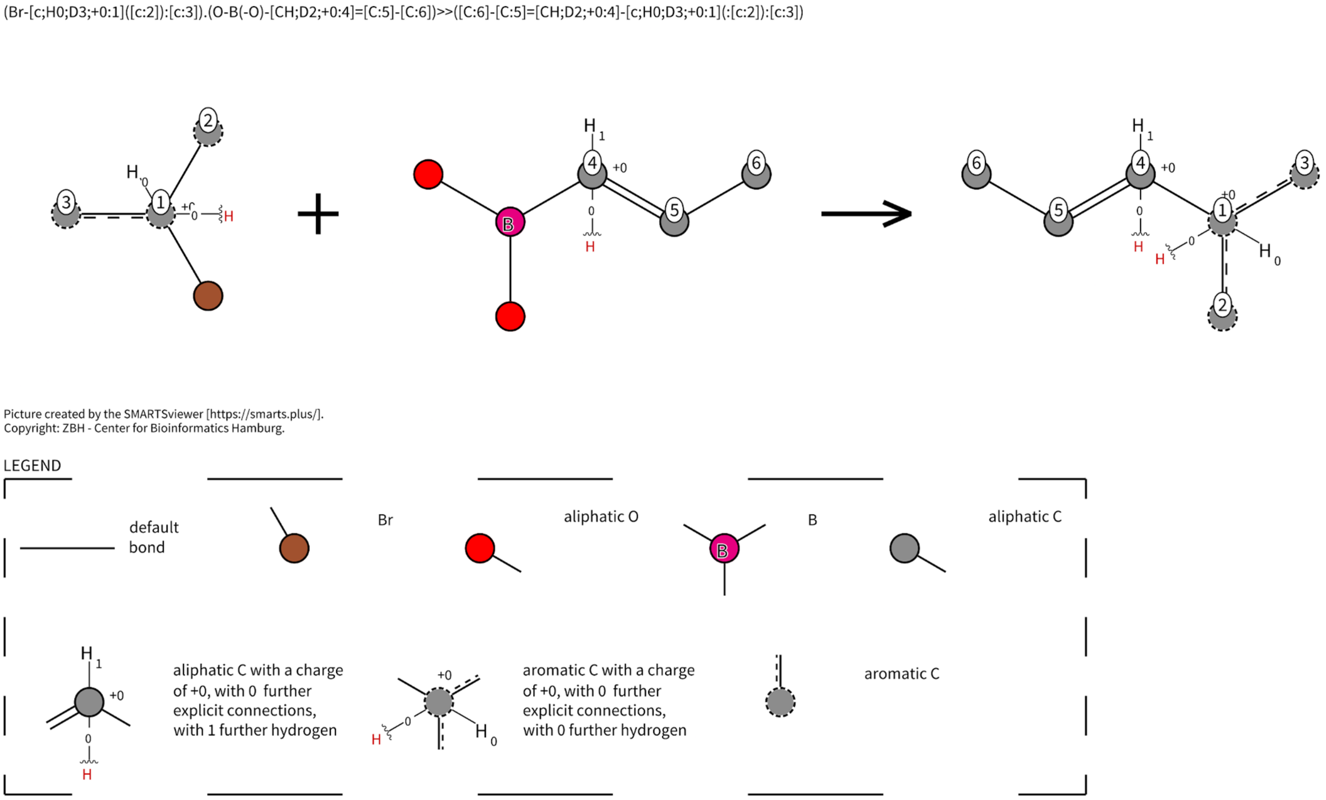
Each pattern consists of a reactant and a product part, separated by the characters ">>". Both parts can contain any number of disconnected patterns describing different components of the reaction, each separated by a dot. For generic reaction schemes, each component, either reactant or product, is described as a SMARTS pattern. In this way, for any given reactant, all of the necessary structural properties required for conversion to the specified products are encoded, using the full power of the SMARTS language. Structural changes are represented by pairwise atom mapping between the reactant and product parts of the pattern using atom labels. The SMIRKS language adds additional restraining rules on top of the Reaction SMARTS format to ensure that the reaction can be converted



(Br-[c;H0;D3;+0:1]([c:2]):[c:3]).(O-B(-O)-[CH;D2;+0:4]=[C:5]-[C:6])>>([C:6]-[C:5]=[CH;D2;+0:4]-[c;H0;D3;+0:1](:[c:2]):[c:3])

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.

Figure 1. Visualization of a Bromo−Suzuki-type coupling reaction scheme generated with ReactionViewer. The underlying SMIRKS pattern was adapted from the transformation data of the AiZynthFinder tool.[2]



(Br-[c;H0;D3;+0:1]([c:2]):[c:3]).(O-B(-O)-[CH;D2;+0:4]=[C:5]-[C:6])>>([C:6]-[C:5]=[CH;D2;+0:4]-[c;H0;D3;+0:1](:[c:2]):[c:3])

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.

Figure 2. Visualization of a Bromo−Suzuki-type coupling reaction scheme generated with ReactionViewer, including the visualization of the atom mapping and a legend. The underlying SMIRKS pattern was adapted from the transformation data of AiZynthFinder.[2]

into a reaction graph for which all structural changes can be derived. For example, to be a valid SMIRKS pattern, each atom label must occur exactly twice in the whole pattern (once to the left and once to the right of the ">>" characters). Another constraint requires that all explicit hydrogens must appear explicitly on both sides of the ">>" symbol (see ref 7 for the full description). The ReactionViewer does not restrict its parsing capabilities to the SMIRKS language rules. The method is able to parse any reaction pattern following the described format where each component is described as a valid SMARTS expression. Invalid SMARTS expressions in any part of the reaction pattern will be marked red and result in an error. Therefore, ReactionViewer is very useful for visual inspection and control of SMIRKS rules during their development. Note that disconnected patterns indicated by a space in the line-notation are not supported. In the following, all reaction patterns referred to as such are meant to be written as either a Reaction SMILES, a Reaction SMARTS, or a SMIRKS pattern. The explicit specification of an agent structure in the reaction is not supported. Therefore, patterns in the format *reactant* > *agent* > *product* will produce an error.

Converting Reaction Patterns into Graphic Representations. Reaction SMARTS patterns are syntactically very similar to disconnected SMARTS patterns. Disconnected SMARTS patterns are composed of two or more distinct SMARTS patterns separated by a dot, but written in one line. The dot indicates that there is no bond between the adjacent atoms such that the subpatterns are disconnected. Note that the dot has highest priority. For example the pattern `1CCC.CC1` will result in an error, because the number indicating a ring opening and closure has a lower priority than the dot. The only syntactic difference between disconnected and reaction patterns is therefore the ">>" symbol.

As already described in a previous section Schomburg et al.[13] already described a visualization concept and an algorithm to convert single SMARTS pattern into images, including disconnected SMARTS pattern. The following describes the simple adaptations of the method to also support reaction patterns. During the initial parsing phase, the ">>" symbol is replaced with a dot transforming the reaction pattern into a disconnected SMARTS pattern. Note, this is only possible if the correct syntax is used. Incorrect patterns, including, for example, multiple ">>" symbols, result in an error message. Next, every SMARTS expression of the disconnected SMARTS pattern is transformed into a tree-like data structure named SMARTS graph that represents the semantic of the pattern. In the case of an underlying reaction pattern, each SMARTS expression describing a reactant or product structure is converted into its own SMARTS graph. Schomburg et al. model the SMARTS language as context-free grammar[18] to achieve an easy extraction of all relevant information.[13] In addition, the SMARTS graph corresponding to the SMARTS pattern before the ">>" symbol is marked as the last reactant, and all SMARTS graphs are marked as reaction graphs. As a next step, each SMARTS graph is checked for its validity. This includes a syntax test, the removal of semantic errors, and a simplification step including the removal of redundant information. For all "purified" SMARTS graphs, an overall legend is generated. In addition to the steps performed by Schomburg et al., all redundant information in the legend resulting from multiple SMARTS graphs is removed. To generate a layout for each SMARTS graph, Schomburg et al. utilized the similarity to the structure diagram generation problem for which Fricker et al.[19] proposed a solution in 2004. All generated layouts are arranged in a row; the generated legend is placed below. In a final step, reaction symbols get placed between the layouts of the different SMARTS graphs.

Visualization of Reaction Schemes. By introducing reaction patterns to the described visualization process, only two additional graphic elements need to be designed. The dot between components of the reaction is visualized as a plus and the ">>" characters as an arrow. We followed IUPAC's "Compendium of Chemical Terminology" definition[20] of a chemical reaction equation.

The implementation of the described algorithm for parsing and visualizing reaction patterns resulted in the adaption of the SMARTSviewer, called ReactionViewer. Figure 1 shows an example visualization generated by ReactionViewer of a reaction pattern of a Bromo−Suzuki-type coupling.

The pairwise atom assignment between the reaction and product patterns is indicated by means of atom labels in the reaction pattern. These labels can optionally be visualized. Each labeled atom receives an additional white circle with a black number corresponding to the set label assigned to the corresponding atom circle. This provides an easy way to visually control all set labels. In addition, the option to show a legend containing an explanation of all displayed SMARTS features is provided. Both visualization options are highly useful for an easy interpretation of reaction patterns. Figure 2 shows the already displayed reaction pattern of the Bromo−Suzuki-type coupling with the additional visualization options.

ReactionViewer is able to visualize not only single reaction patterns but also multiple patterns given in a file. These can be written as Reaction SMILES, Reaction SMARTS, or SMIRKS. Thereby, complete reaction data sets can be automatically visualized. Multiple patterns can be exported in one PDF file, and single patterns can be saved to PDF documents as well as SVG and PNG images.

With the described workflow, the ReactionViewer adaption, SMARTSviewer, is able to display not only SMILES and SMARTS but Reaction SMILES, Reaction SMARTS, and SMIRKS in the same interface on the command line level. The reaction visualization can be used either via the web interface of the SMARTS.plus server[21] at https://smarts.plus/ or with the graphical user interface of the SMARTSviewer tool. Visual comparison between two reaction patterns is possible via the "Compare" mode of the SMARTS.plus server. However, the underlying comparison algorithm[22] does not support reaction patterns so far.

RESULTS AND DISCUSSION

In the following, we present two experiments to investigate the performance and utility of automatically generating the visualization of reaction patterns. First, we visualize a large data set, containing 46 695 reaction schemes, to verify the applicability of the visualization routine. Then, we compare the results of ReactionViewer with given schematic representations for a smaller Reaction SMARTS data set, containing 58 reaction schemes.

Data Sets. To test the automatic generation of visualizations of a reaction pattern, we extracted two data sets of synthesis reaction patterns of two real-world applications. The first data set comes from open-source retrosynthetic planning software called AiZynthFinder.[2] The tool uses a Monte Carlo tree search guided by an artificial neural network policy to find a retrosynthetic pathway for a given target molecule, based only on commercially available starting materials. The tool is available with a set of reaction schemes used for the neural network policy training, among others. The reaction data set is based on the publicly available US patent office data. The 46 695 reaction schemes are written in a retrosynthetic manner, meaning that the product structure is on the left side of the ">>" symbol and the reactant structures are on the right side. This corresponds with the concept of breaking down the structures into precursor structures.[2] The largest string consists of 1081, the smallest of 41 characters. The largest number of reactants across all reaction schemes is seven, as is the largest number of products. The smallest number is one in each case. In the following, we will refer to this set of reaction schemes as the AiZynthFinder data set.

The second data set is extracted from Hartenfeller et al.,[14] who provided a set of robust organic reaction schemes available for in silico molecule design. The 58 reaction schemes are provided as Reaction SMARTS. They were codeveloped by medicinal chemists to be highly practical and applicable in real-world chemistry.[14] The largest string consists of 244, the smallest of 47 characters. All reaction schemes specify one or two reactants and one product. In the following, we will refer to this set of reaction schemes as the Hartenfeller data set.

Visualization and Comparison. Despite the wide range of reaction types, the described visualization routine was successfully applied to all reaction schemes of both data sets. The complete visualization of both data sets can be downloaded from https://www.zbh.uni-hamburg.de/forschung/amd/

([C:8]-[C;H0;D3;+0:7](-[c:9])=[CH;D2;+0:1]-[C:2](=[O;D1;H0:3])-[#8:4]-[C:5]-[C;D1;H3:6])>>(Br-[CH2;D2;+0:1]-[C:2](=[O;D1;H0:3])-[#8:4]-[C:5]-[C;D1;H3:6]).(O=[C;H0;D3;+0:7](-[C:8])-[c:9])

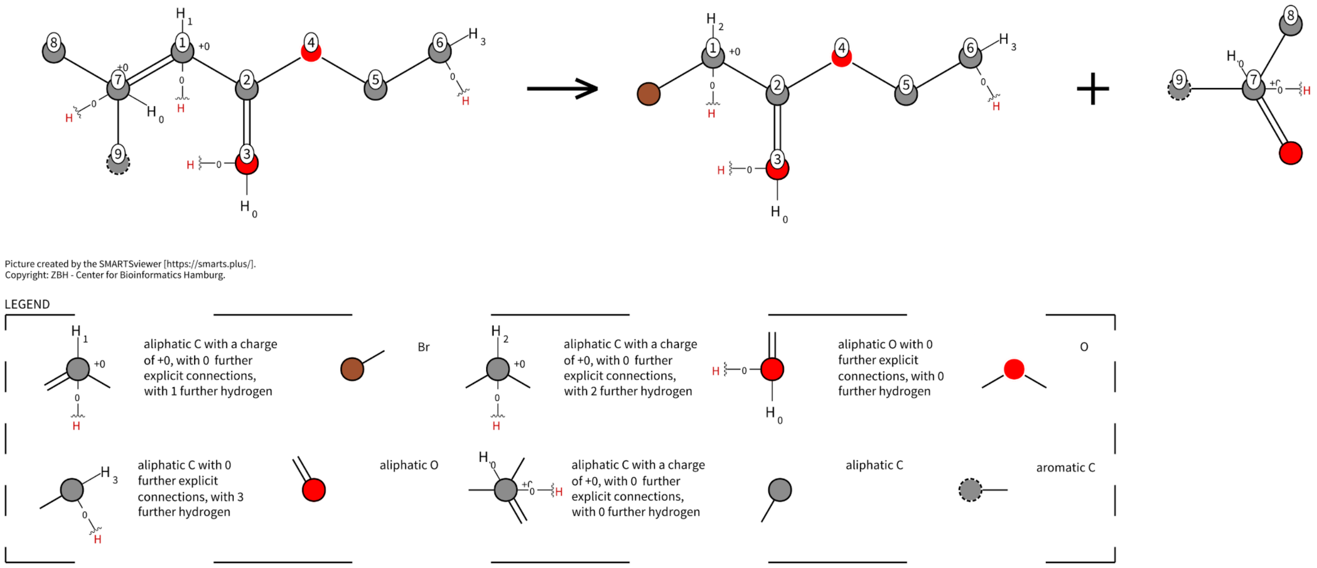

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
aliphatic C with a charge of +0, with 0 further explicit connections, with 1 further hydrogen
Br
aliphatic C with a charge of +0, with 0 further explicit connections, with 2 further hydrogen
aliphatic O with 0 further explicit connections, with 0 further hydrogen
O
aliphatic C with 0 further explicit connections, with 3 further hydrogen
aliphatic O
aliphatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic C
aromatic C

Figure 3. A reaction scheme of the AiZynthFinder data set visualized with the SMARTSviewer tool, classified as a Wittig olefination. Above, the corresponding reaction pattern.

([#7;a:6]:[c:7]:[c;H0;D3;+0:8](:[c:9])-[n;H0;D3;+0:10]1:[cH;D2;+0:1]:[c;H0;D3;+0:2](-[c;H0;D3;+0:3](:[#7;a:4]):[c:5]):[n;H0;D2;+0:12]:[c;H0;D3;+0:11]:1-[c;H0;D3;+0:13]1:[c:14]:[c:15]:[c:16]:[c:17]:[c:18]:1)>>(Br-[CH2;D2;+0:1]-[C;H0;D3;+0:2](=O)-[c;H0;D3;+0:3](:[#7;a:4]):[c:5]).([#7;a:6]:[c:7]:[c;H0;D3;+0:8](:[c:9])-[N;H0;D2;+0:10]=[C;H0;D3;+0:11](-[NH2;D1;+0:12])-[c;H0;D3;+0:13]1:[c:14]:[c:15]:[c:16]:[c:17]:[c:18]:1)

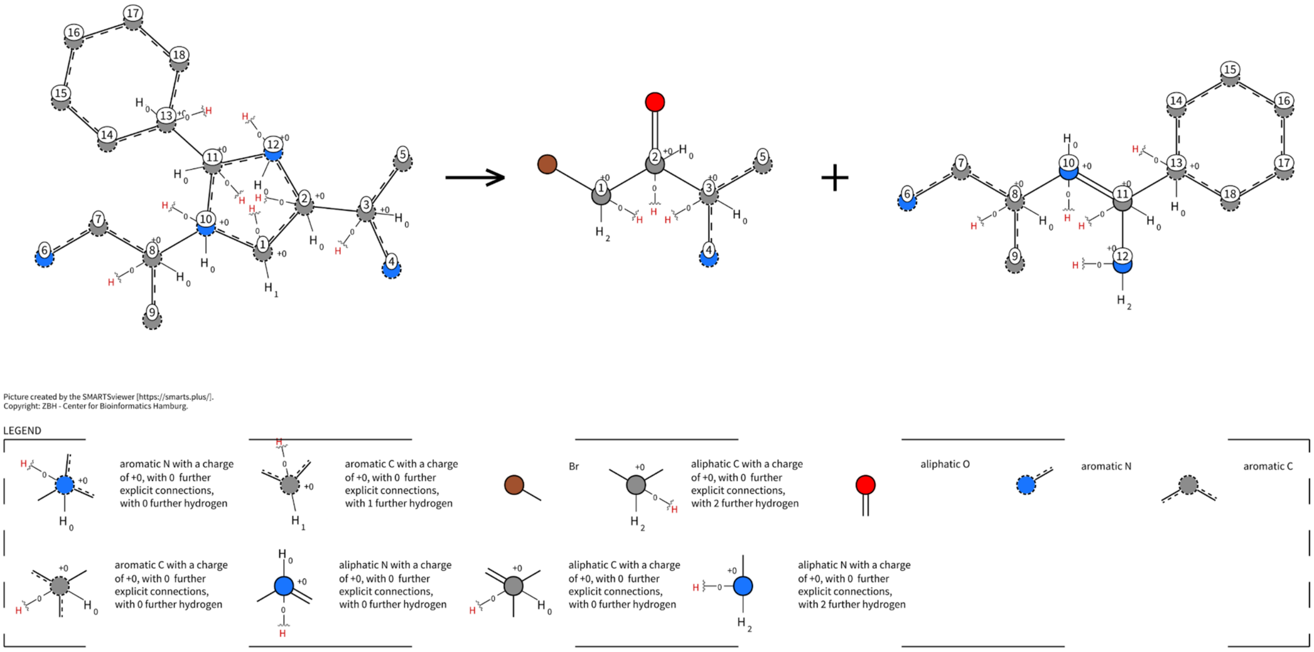

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
aromatic N with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aromatic C with a charge of +0, with 0 further explicit connections, with 1 further hydrogen
Br
aliphatic C with a charge of +0, with 0 further explicit connections, with 2 further hydrogen
aliphatic O
aromatic N
aromatic C
aromatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic N with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic N with a charge of +0, with 0 further explicit connections, with 2 further hydrogen

Figure 4. A reaction scheme of the AiZynthFinder data set visualized with the SMARTSviewer tool, classified as an organometallic C−C bond formation. Above, the corresponding reaction pattern.

datasets/reaction-viewer-datasets.html. Figures 3−5 show the visualization of three out of the 46 695 reaction schemes of the AiZynthFinder data set together with the corresponding reaction pattern.

Comparing the pattern strings with the visualization, one can easily see the advantages in terms of interpretability of the patterns. At a glance, the images provide clarity about general information such as the number of reactants and products and

([#7:7]-[C:6]-[c:5]1:[c:4]:[c:3]:[c:2]:[c;H0;D3;+0:1](-[C;H0;D3;+0:9](-[C;D1;H3:10])=[O;D1;H0:11]):[c:8]:1)>>(Br-[c;H0;D3;+0:1]1:[c:2]:[c:3]:[c:4]:[c:5](-[C:6]-[#7:7]):[c:8]:1).(C-O-N(-C)-[C;H0;D3;+0:9](-[C;D1;H3:10])=[O;D1;H0:11])

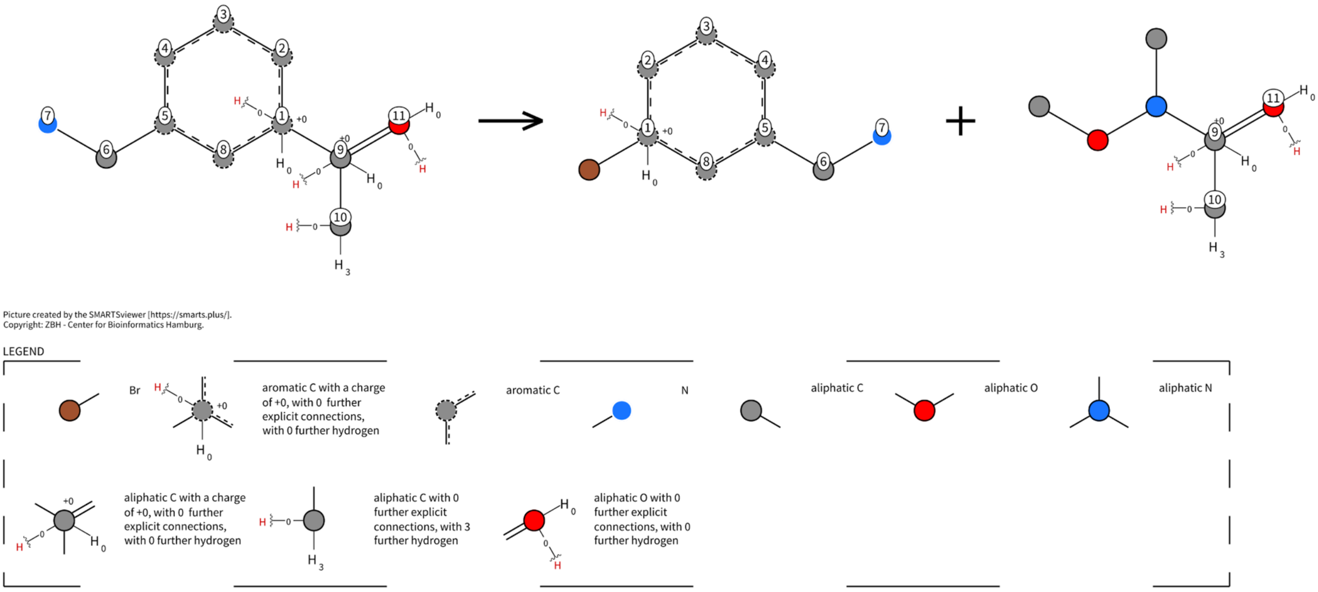
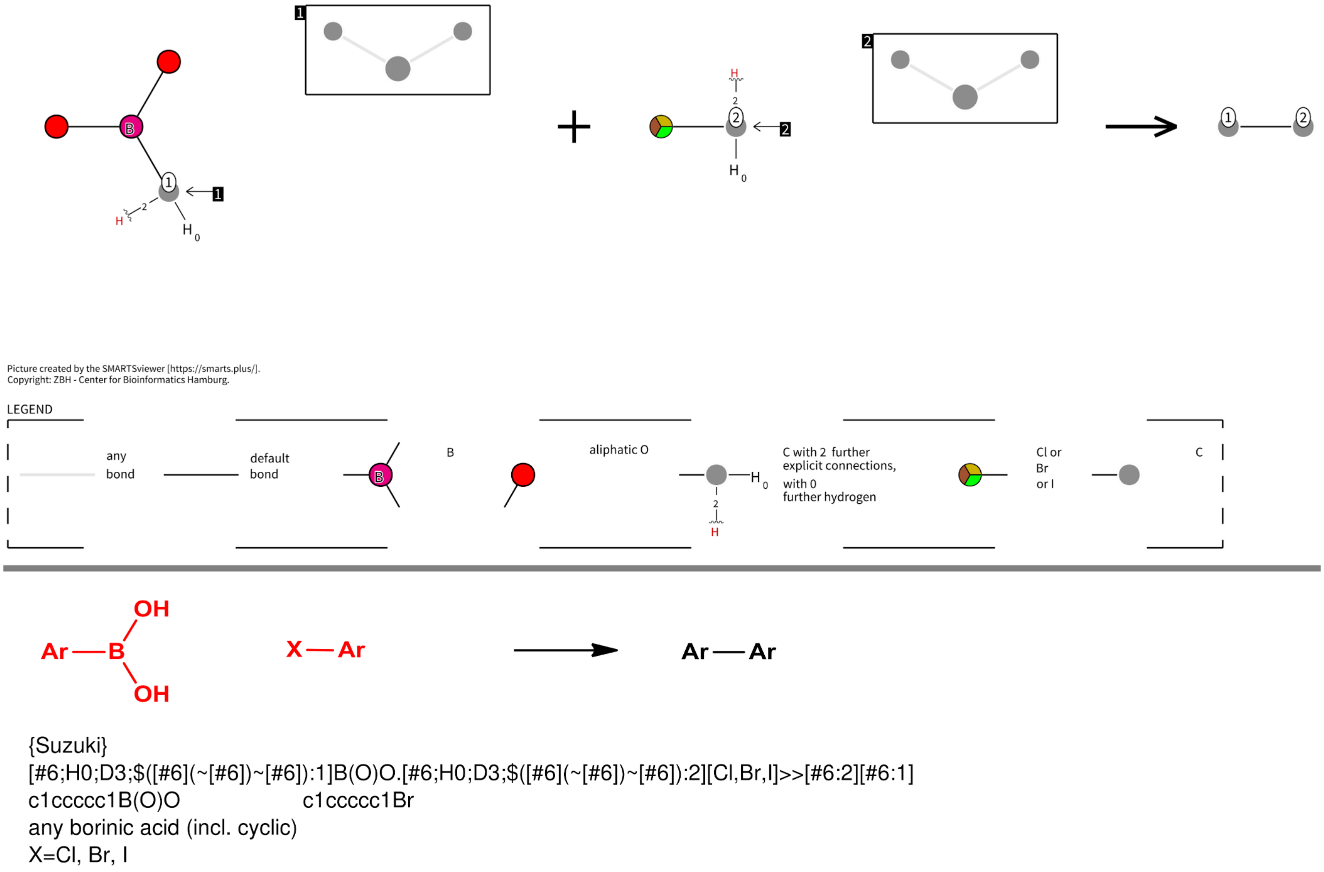

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
Br
aromatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aromatic C
N
aliphatic C
aliphatic O
aliphatic N
aliphatic C with a charge of +0, with 0 further explicit connections, with 0 further hydrogen
aliphatic C with 0 further explicit connections, with 3 further hydrogen
aliphatic O with 0 further explicit connections, with 0 further hydrogen

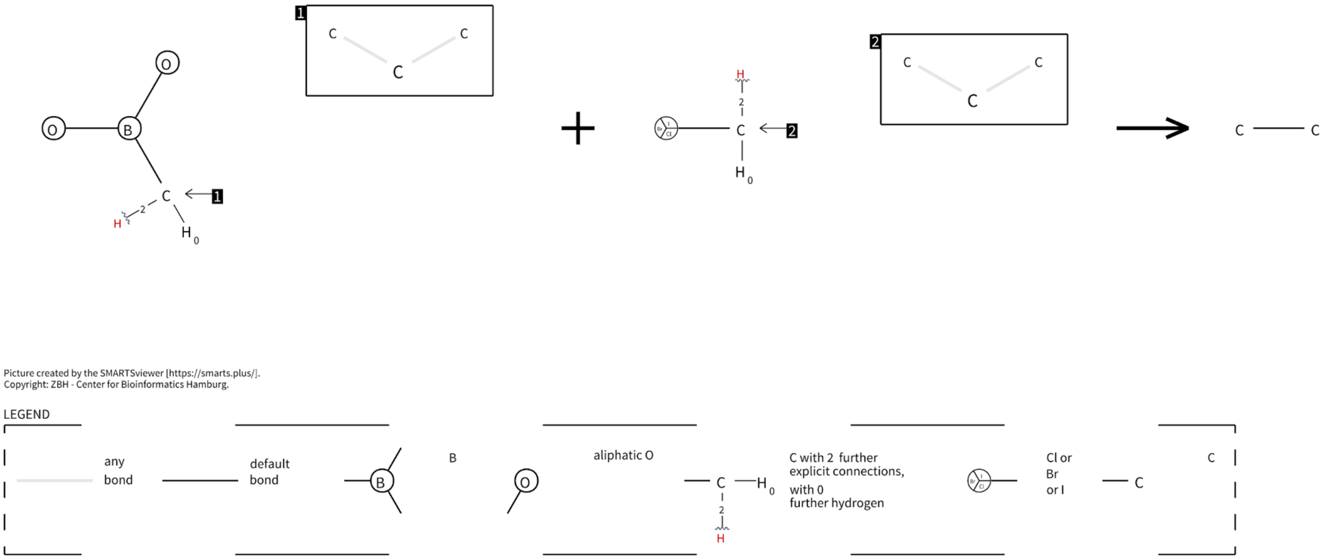
Figure 5. A reaction scheme of the AiZynthFinder data set visualized with the SMARTSviewer tool, classified as a N-containing heterocycle formation. Above, the corresponding reaction pattern.


Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
any bond
default bond
B
aliphatic O
C with 2 further explicit connections, with 0 further hydrogen
Cl or Br or I
C
Ar—B(OH)(OH)
X—Ar
Ar—Ar

{Suzuki}
[#6;H0;D3;$([#6](~[#6])~[#6]):1]B(O)O.[#6;H0;D3;$([#6](~[#6])~[#6]):2][Cl,Br,I]>>[#6:2][#6:1]
c1ccccc1B(O)O c1ccccc1Br
any borinic acid (incl. cyclic)
X=Cl, Br, I

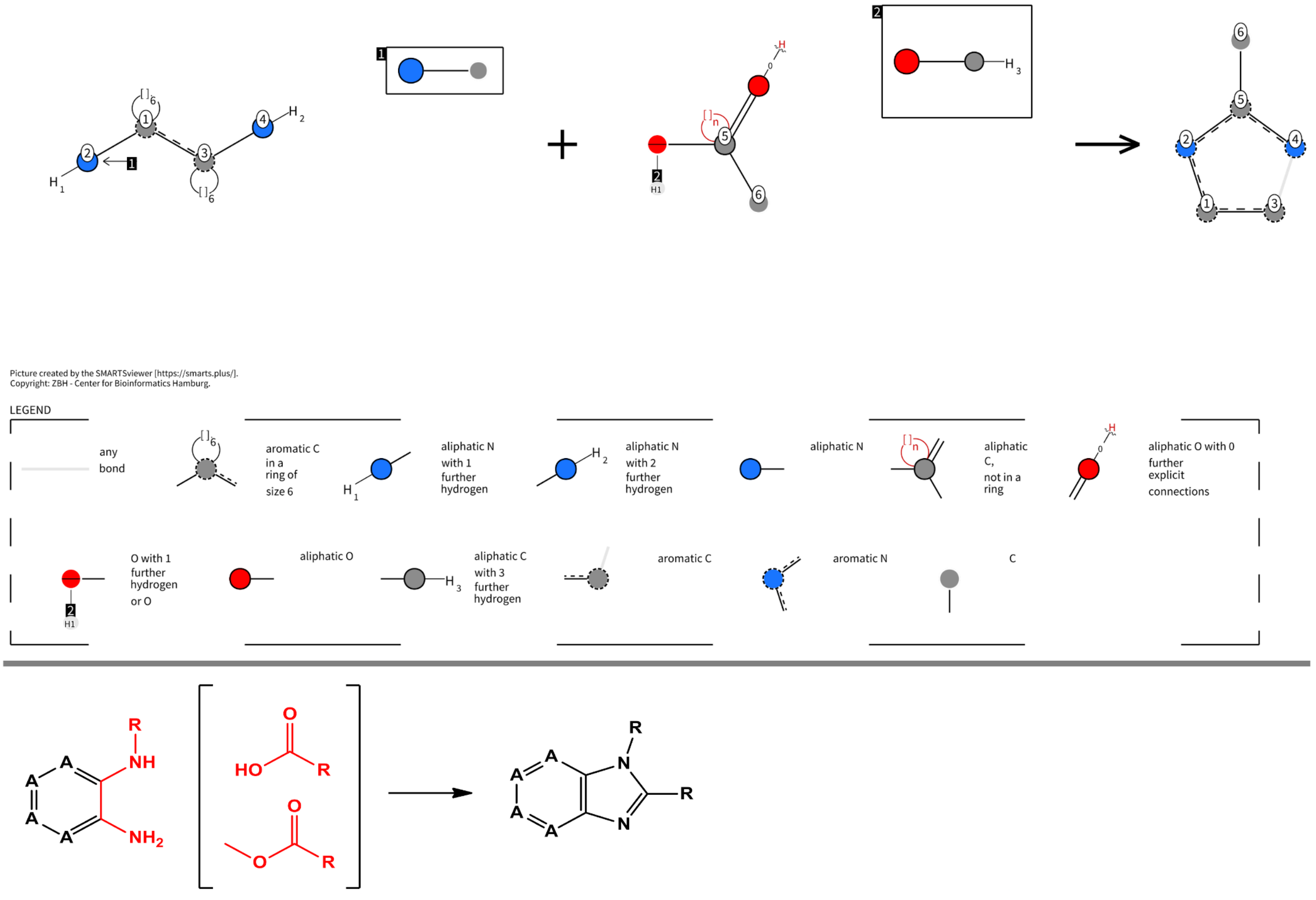
Figure 6. Visualization of the same reaction scheme, a Suzuki coupling. The upper image shows the visualization generated by ReactionViewer. Below is the visualization provided by Hartenfeller et al.[14]

their overall structure. Closer inspection then reveals the different atom and bond types and properties, all visualized with easily interpretable and intuitive color and shape schemes, including a detailed explanation in the legend. Furthermore, the


1
2
LEGEND
any bond
default bond
B
aliphatic O
C with 2 further explicit connections, with 0 further hydrogen
Cl or Br or I
C

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.

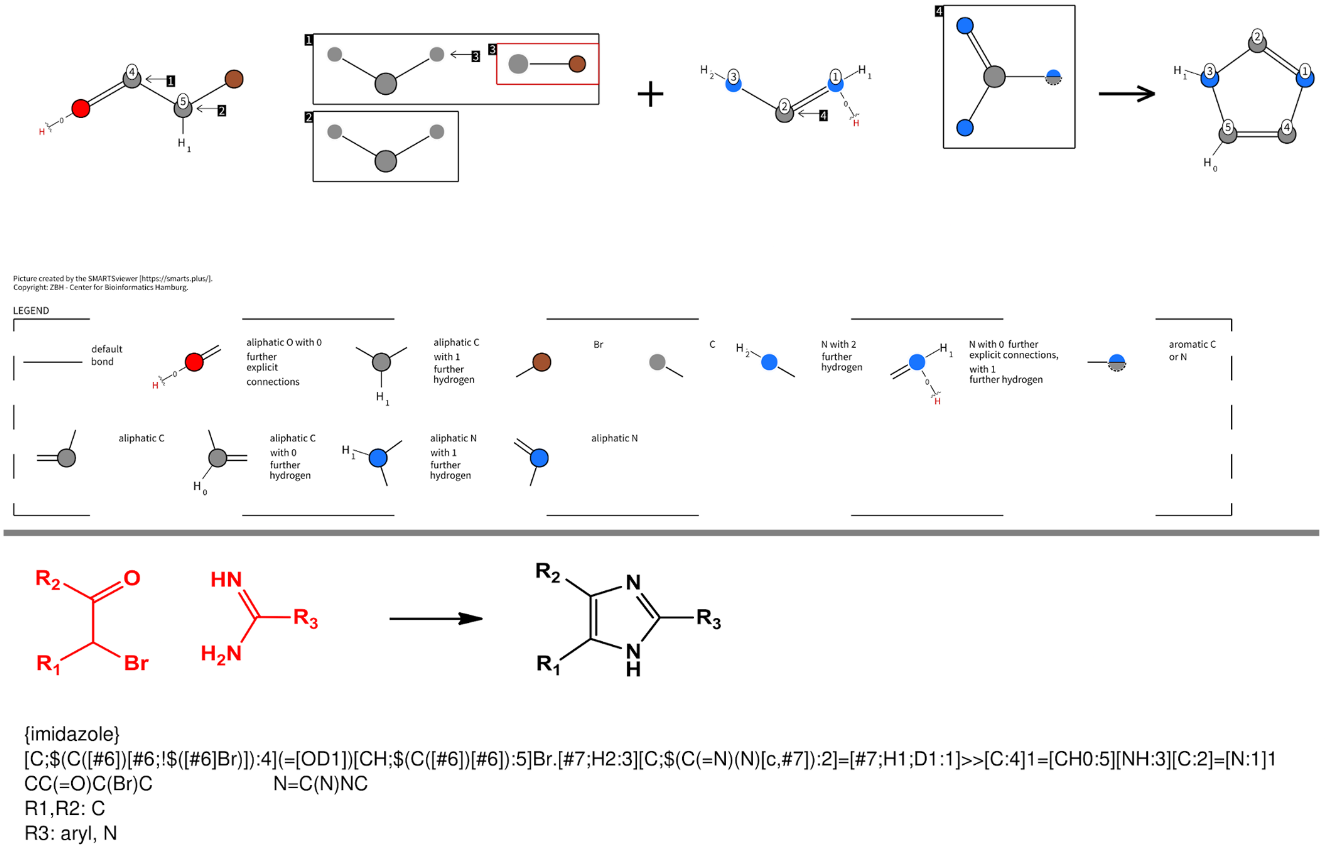
Figure 7. Visualization of a Suzuki coupling generated by ReactionViewer with the visualization modus: element symbols. The underlying reaction pattern is the same as in Figure 6.


1
2
LEGEND
any bond
aromatic C in a ring of size 6
aliphatic N with 1 further hydrogen
aliphatic N with 2 further hydrogen
aliphatic N
aliphatic C, not in a ring
aliphatic O with 0 further explicit connections
O with 1 further hydrogen or O
aliphatic O
aliphatic C with 3 further hydrogen
aromatic C
aromatic N
C
R
NH
NH_2
A
HO
O
N

Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.

{benzimidazole_derivatives_carboxylic-acid/ester}
[c;r6:1](-[NH1;$(N-[#6]):2]):[c;r6:3](-[NH2:4]).[#6:6]-[C;R0:5](=[OD1])-[#8;H1,$(O-[CH3])]>>[c:3]2:[c:1]:[n:2]:[c:5](-[#6:6]):[n:4]@2
c1c(NC)c(N)ccc1 CC(=O)O
Any sixmembered aromatic heterocycle

Figure 8. Visualization of the same reaction scheme, named benzimidazole derivatives carboxylic-acidester. The upper image shows the visualization generated by ReactionViewer. Below is the visualization provided by Hartenfeller et al.[14]


4
1
5
2
H
0
H
1
1
3
3
2
+
H
2
3
2
1
H
1
4
0
H
4
→
H
1
3
2
1
5
4
H
0
Picture created by the SMARTSviewer [https://smarts.plus/].
Copyright: ZBH - Center for Bioinformatics Hamburg.
LEGEND
default bond
aliphatic O with 0 further explicit connections
aliphatic C with 1 further hydrogen
Br
C
N with 2 further hydrogen
N with 0 further explicit connections, with 1 further hydrogen
aromatic C or N
aliphatic C
aliphatic C with 0 further hydrogen
aliphatic N with 1 further hydrogen
aliphatic N
R2
O
R1
Br
HN
R3
H2N
R2
N
R3
R1
N
H

```
{imidazole}
[C;$(C([#6])[#6;!$([#6]Br)]):4](=[OD1])[CH;$(C([#6])[#6]):5]Br.[#7;H2:3][C;$(C(=N)(N)[c,#7]):2]=[#7;H1;D1:1]>>[C:4]1=[CH0:5][NH:3][C:2]=[N:1]1
CC(=O)C(Br)C                    N=C(N)NC
R1,R2: C
R3: aryl, N
```

Figure 9. Visualization of the same reaction scheme, an imidazole synthesis. The upper image shows the visualization generated by ReactionViewer. Below is the visualization provided by Hartenfeller et al.[14]

atomic mapping between reaction and product atoms can be easily read and controlled. If one looks at the pattern strings, it takes much longer to read and map all of the described information. Even experienced users will have their difficulties visualizing the reaction scaffold with a glance at the pattern string. Extracting more detailed information, such as the specified number of attached hydrogens for a specific atom, requires time and skill, both of which can be saved by the visualization routine.

Hartenfeller et al. provide a visual representation of their reaction schemes. In the following, we compare the two visualization techniques and discuss differences and similarities. We would like to emphasize that the focus of the Hartenfeller publication is not on the visualization of the provided Reaction SMARTS but rather on the generation and evaluation. Their provided visualization of the reaction schemes is only used to highlight the features of the ReactionViewer visualization in comparison to a common visualization technique. Figures 6−9 show three exemplary reaction schemes of the Hartenfeller data set both visualized by ReactionViewer and the original authors.

Comparing the two visualization techniques, there are a few general differences. First, the visualization of Hartenfeller et al. uses explicit element symbols or letter abbreviations to describe a single or multiple possible element types of atoms, whereas ReactionViewer relies on a color scheme and only uses explicit element types if no established color exists for the element. However, ReactionViewer has the option to change the visualization to element symbols instead of being color based. Figure 7 shows an example: the Suzuki coupling reaction scheme of Figure 9 with the different visualization modus. In addition, the Hartenfeller visualization uses only two colors to distinguish reactant atoms involved with the reaction center. This information is not provided by the SMARTS pattern and thereby must be the result of human interpretation. Furthermore, the Hartenfeller visualization, in contrast to the ReactionViewer visualization, does not feature atom labels that correspond to the atom assignment given in the SMARTS reaction.

The Suzuki coupling reaction scheme, visualized in Figure 6, is a rather simple pattern, describing the coupling of a boronic acid and an organohalide to form a carbon−carbon single bond. Both visualization techniques show this structural framework. However, they differ in minor components. ReactionViewer shows a higher level of detail of the given Reaction SMARTS pattern and, unlike the Hartenfeller visualization, translates the pattern one to one. This can be easily seen in the first atom of the first reactant described in the Reaction SMARTS pattern, which looks like this: `[#6;H0;D3;$([#6](~[#6])~[#6]):1]`. Translating this description into spoken language, the atom labeled one is a carbon atom with two other explicit bonds and no hydrogens attached. The two bonded atoms are described by a recursive expression that states that the atoms are also carbon atoms bonded with arbitrary bonds. All of this information is included in the ReactionViewer visualization, and the details of the bonds are shown right next to the corresponding atom and are further explained in the legend. The recursively described atomic environment is explicitly visualized as an independent molecular graph next to the reactant structure. The corresponding atom is described in the Hartenfeller visualization with the letter abbreviation “Ar”, which stands for “Aryl”.

Figure 8 shows a reaction scheme classified as benzimidazole derivative carboxylic-acid or -ester. Here, the visualized structural framework differs between the two visualization techniques. The SMARTSviewer visualization again follows the given Reaction SMARTS pattern one-to-one. Atoms with the labels 1 and 3 in the first reactant are both par in any six-membered ring, indicated by the following part `r6`. This ring is explicitly visualized in the Hartenfeller visualization together with the element specification “A”, which stands for either a nitrogen or a carbon. ReactionViewer only displays this ring implicitly, by annotating its existence on the corresponding atoms, due to its implicit specification in the pattern string. The ring is displayed explicitly in the Hartenfeller visualization but not at all in the ReactionViewer visualization. Again, ReactionViewer only follows the Reaction SMARTS pattern one-to-one, in which the ring is not mentioned in the product part anymore. Further details, like the arbitrary bond type of the bond between the product atoms with labels 3 and 4 (`[c:3]..[n:4]@2`) and the fact that the reactant atom with label 5 is not allowed to be in a ring (`[C;R0:5]`) are only visualized by ReactionViewer.

In Figure 9, a reaction scheme of an imidazole synthesis is displayed. Comparing the two visualization techniques, a large difference again lies in the display of the recursion part of the SMARTS pattern. The Hartenfeller scheme visualizes it in form of an R_i atom identifier, where each i is explained under the image. ReactionViewer displays the recursion part more explicitly and precisely. In this example, one can see a display of a logical NOT in the SMARTS pattern, which is separated in its corresponding recursive box as an additional box with red borders. This information is not directly displayed in the Hartenfeller visualization. The additional R_i atom identifier in the product structure of the Hartenfeller visualization are not specified in the Reaction SMARTS pattern and are added due to a human contextual interpretation.

All three examples have shown that ReactionViewer differs from the visualization technique of the Hartenfeller data set mainly in the point that it visualizes the given pattern one-to-one. The Hartenfeller visualization shows its origin as a human product where an interpretation of the pattern has already taken place. The ReactionViewer visualization is generated fully automatically and can only display as much information as is given in the pattern. However, it can display this in full detail and with a satisfactory explanation.

CONCLUSION

Working with chemical patterns in computer-readable languages can be a challenging task. In particular, reaction schemes, which are usually the largest and most complex type of chemical patterns, are difficult to formulate without error. At least two different patterns, one for a reactant and one for a product structure, and often more, must be translated to fit the specified reaction center exactly. A missing hydrogen atom or a mismatched atom assignment can result in a nonfunctioning pattern, or worse, in a erroneous transformation.

In this work, we presented ReactionViewer, a method for automatic visualization of generic reaction patterns. The visualization concept follows the model of structure diagrams together with IUPAC’s definition of chemical reaction equations. We showed the applicability of ReactionViewer with two different, real-world data sets. The visualization of the 46 695 reaction schemes of the AiZynthFinder data set showed the advantages of the graphical form of reaction schemes in contrast to their linear form. The visualization in contrast to the textual reaction schemes, which consists of several lines of text, results in a substantially improved readability and interpretability. A comparison between the ReactionViewer visualization and the visualization concept used in Hartenfeller et al. highlighted the feature-richness of the ReactionViewer visualization technique: ReactionViewer provides an explicit one-to-one translation with a detailed explanation of the visualized structural components. In addition, examples of the easy-to-interpret visualizations of syntactic elements representing recursion or logical operators are provided. However, due to its fully automated generation approach, ReactionViewer lacks additional information on the Hartenfeller visualization concept which is conveyed by human interpretation. This should be seen as a strength, since the patterns are visualized exactly in the way they are interpreted by downstream applications.

With the results of this work, we have presented an automatic visualization technique for reaction schemes written as Reaction SMILES, as Reaction SMARTS, or in SMIRKS. Due to its intuitive visualization design, the method can directly support the medicinal chemists during the interpretation, generation, or correction of these patterns. While a graphical editor for SMARTS exists,[23] the corresponding development of a Reaction SMARTS editor remains a task for the future.

ASSOCIATED CONTENT

Data Availability Statement

The complete visualization of the AiZynthFinder[2] data set can be downloaded from https://www.zbh.uni-hamburg.de/forschung/amd/datasets/reaction-viewer-datasets.html. ReactionViewer is available as command line tool 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, ReactionViewer is integrated at our web frontend https://smarts.plus.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c00992.

Complete visualization of the Hartenfeller[14] data set (PDF)

AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany;* orcid.org/0000-0002-9553-6531; Phone: +49 (40) 428387351; Email: matthias.rarey@uni-hamburg.de

Authors

Uschi Dolfus − *Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany;* orcid.org/0000-0002-2720-1086

Hans Briem − *Bayer AG, Research and Development, Pharmaceuticals, Computational Molecular Design Berlin, 13342 Berlin, Germany*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jcim.2c00992

Author Contributions

U.D., H.B., and M.R. developed the ReactionViewer concept. U.D. implemented the ReactionViewer approach. M.R. supervised the project. All authors participated in manuscript writing.

Notes

The authors declare the following competing financial interest(s): M.R., as a shareholder of BioSolveIT GmbH, declares a potential financial interest in the event that the ReactionViewer software is licensed for a fee to nonacademic institutions in the future.

■ REFERENCES

(1) Segler, M. H.; Preuss, M.; Waller, M. P. Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI. *Nature* **2018**, *555*, 604−610.

(2) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: A Fast, Robust and Flexible Open-Source Software for Retrosynthetic Planning. *J. Cheminf.* **2020**, *12*, 1−9.

(3) Szymkuć, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.

(4) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation. *J. Chem. Inf. Model.* **2009**, *49*, 593−602.

(5) Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehiers, P. P.; Byington, J.; Piotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, *365*, eaax1566.

(6) Baylon, J. L.; Cilfone, N. A.; Gulcher, J. R.; Chittenden, T. W. Enhancing Retrosynthetic Reaction Prediction with Deep Learning Using Multiscale Reaction Classification. *J. Chem. Inf. Model.* **2019**, *59*, 673−688.

(7) Daylight, SMIRKS - A Reaction Transform Language. https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html (accessed March 7, 2022).

(8) Daylight, SMARTS - A Language for Describing Molecular Patterns. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (accessed May 25, 2021).

(9) Daylight, SMILES - A Simplified Chemical Language. https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html (accessed May 25, 2021).

(10) ChemAxon, ChemAxon−Software solutions and services for chemistry and biology, 2016. https://chemaxon.com/products/marvin, (accessed September 1, 2022).

(11) Csizmadia, P. *MarvinSketch and MarvinView: Molecule Applets for the World Wide Web*; The 3rd International Electronic Conference on Synthetic Organic Chemistry (ECSOC-3), 1999.

(12) ChemAxon, MarvinJs. https://marvinjs-demo.chemaxon.com/latest/index.html (accessed September 1, 2022).

(13) Schomburg, K.; Ehrlich, H.-C.; Stierand, K.; Rarey, M. From Structure Diagrams to Visual Chemical Patterns. *J. Chem. Inf. Model.* **2010**, *50*, 1529−1535.

(14) Hartenfeller, M.; Eberle, M.; Meier, P.; Nieto-Oberhuber, C.; Altmann, K.-H.; Schneider, G.; Jacoby, E.; Renner, S. A Collection of Robust Organic Synthesis Reactions for In Silico Molecule Design. *J. Chem. Inf. Model.* **2011**, *51*, 3093−3098.

(15) Helson, H. E. Structure Diagram Generation. *Rev. Comput. Chem.* **1999**, *13*, 313−398.

(16) Hilbig, M.; Urbaczek, S.; Groth, I.; Heuser, S.; Rarey, M. MONA−Interactive Manipulation of Molecule Collections. *J. Cheminf.* **2013**, *5*, 38.

(17) Hilbig, M.; Rarey, M. MONA 2: A Light Cheminformatics Platform for Interactive Compound Library Processing. *J. Chem. Inf. Model.* **2015**, *55*, 2071.

(18) Alfred, A.; Ullman, J. *The Theory of Parsing, Translation and Compiling*; Prentice Hall, 1972.

(19) Fricker, P. C.; Gastreich, M.; Rarey, M. Automated Drawing of Structural Molecular Formulas under Constraints. *J. Chem. Inf. Model.* **2004**, *44*, 1065−1078.

(20) McNaught, A. D.; Wilkinson, A. *Compendium of Chemical Terminology*, 2nd ed. (the "Gold Book"); Blackwell Science Oxford, 1997; Vol. *1669*.

(21) Ehrt, C.; Krause, B.; Schmidt, R.; Ehmki, E. S.; Rarey, M. SMARTS.plus − A Toolbox for Chemical Pattern Design. *Mol. Inf.* **2020**, *39*, 2000216.

(22) Schmidt, R.; Ehmki, E. S.; Ohm, F.; Ehrlich, H.-C.; Mashychev, A.; Rarey, M. Comparing Molecular Patterns Using the Example of SMARTS: Theory and Algorithms. *J. Chem. Inf. Model.* **2019**, *59*, 2560−2571.

(23) Schomburg, K. T.; Wetzer, L.; Rarey, M. Interactive Design of Generic Chemical Patterns. *Drug Discovery Today* **2013**, *18*, 651−658.

D.2 Synthesis-Aware Generation of Structural Analogues

[D2] **U. Dolfus**, H. Briem, and M. Rarey. “Synthesis-aware generation of structural analogues”. In: *Journal of Chemical Information and Modeling* 62.15 (2022), pp. 3565–3576.

The following pages contain the published manuscript. Due to the length of the Supporting Information, the corresponding pages are not included in this document. They can be found here `https://doi.org/10.1021/acs.jcim.2c00246`. The Supporting Information provides an overview and detailed information on the different query constraints that can be used. It also includes a template for the expected file format of retrosynthetic trees and additional information on experimental results. The list of compounds used in the proof-of-concept experiment and the retrosynthetic routes for the generalised filter and cluster experiments are also included. In addition, the data can also be downloaded as JSON and SMI files from here `https://doi.org/10.1021/acs.jcim.2c00246`.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

Synthesis-Aware Generation of Structural Analogues

Uschi Dolfus, Hans Briem, and Matthias Rarey*

Cite This: *J. Chem. Inf. Model.* 2022, 62, 3565−3576

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: In modern drug design, one of the main issues is the optimization of an initial lead structure toward a drug candidate by modifying specific properties in the desired direction. The synthetic feasibility of the target structure is often neglected during this process, resulting in structures with low or suboptimal synthetic accessibility. In this work, we present a novel approach for synthesis-aware lead optimization called Synthesia. In contrast to the traditional approaches, Synthesia integrates the preservation of the synthesizability of the target structure into the lead structure modification process. Synthesia is able to create structural diversity for a lead structure that matches user-defined molecular properties without losing the applicability of a particular synthetic pathway. The methodology is validated by demonstrating that Synthesia is capable of providing structural analogues of DrugBank compounds that meet generic modification goals and maintain their synthetic pathways. In addition, Synthesia is used to cluster compounds from two different patent structure series (CDK7, Daurismo) according to their compatibility with the same synthetic pathways, maximizing the synthetic efficiency and providing an initial estimation of the effort of synthesizing the entire series. Altogether, we demonstrate Synthesia's ability to modify compound properties while maintaining in silico synthesizability.

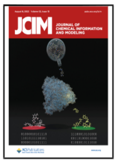
■ INTRODUCTION

Molecular optimization of lead compounds toward drug candidates with desired properties is one of the main tasks in modern drug development. An established approach is the "analogue design"[1] of experimentally determined lead structures. The challenge is to find and apply structural modifications which increase biological or pharmacological activity and minimize undesirable properties such as toxicological behavior or poor solubility. Due to its intricacy, this process is often modeled virtually. Nowadays, computational drug design includes methods from virtual screening of lead optimization libraries over scaffold hopping to molecular dynamics simulations, all capable of helping the medical chemist in their task of lead structure optimization. However, virtually generated or modified structures often have low or suboptimal synthetic accessibility, as synthesizability is often disregarded in the design process due to its complexity. There are numerous possible synthesis steps, which are almost impossible to oversee manually during the design and modification process. In addition, the evaluation of interaction motifs may be biased by the opinion of the researcher. Possible chemical reactions or the availability of thousands of precursor molecules may be overlooked. Computer-assisted synthesis planning (CASP) provides techniques to assist medical chemists in designing synthetic routes for the current structure of interest. So far, however, the optimization of compound properties and synthetic accessibility are mostly considered separate.

Retrosynthetic analysis is a fundamental technique for synthetic route design and the basis for many state-of-the-art CASP tools. Vlěduts introduced the idea of reaction codification and CASP in 1963.[2] Corey and Wipke followed with the first logical retrosynthetic route planning based on a computer program in 1969.[3] The core idea of retrosynthetic analysis is to recursively partition and thereby simplify lead structures into smaller precursors by applying formally reversed chemical reactions as structural transformations. This process is finished when each starting structure is either commercially available or trivial in its synthesis. The product structure can then be created by forward synthesis, following the generated pathway. In theory, the described process can result in huge numbers of synthetic routes for a target molecule, for which the possibilities are only limited by the individual knowledge of chemical reactions and the endurance of the analyzing chemist.

There are two frequently implemented strategies to provide a user with a retrosynthetic pathway for a lead structure: Searching a reaction database and performing an exhaustive retrosynthetic analysis of the target structure, or analyzing the topology of the target structure by applying rule-based expert systems including

Received: March 1, 2022
Published: July 22, 2022



https://doi.org/10.1021/acs.jcim.2c00246

heuristics to propose a synthetic pathway.[4] One of the first CASP programs is Corey's Logic and Heuristics Applied to Synthetic Analysis (LHASA), an interactive method for synthetic route design, which is mainly based on the application of reaction rules.[5,6] With the (re)introduction of machine learning (ML) into synthesis prediction and planning problems, a new type of CASP programs is introduced, which shows substantial improvements in handling the complexity of exhaustive retrosynthetic analysis.[7] A variety of ML techniques have already been assigned to assist in the task of retrosynthetic route generation. Examples are the Monte Carlo tree search algorithm combined with neural networks used to discover retrosynthetic routes for given target molecules[8] or the prediction of chemical reactivity performed by graph neural networks and classifiers successfully used for the selection of reaction conditions.[9,10] Furthermore, there exist complete software packages giving the user a variety of tools covering the different tasks of computer-assisted synthesis planning. Popular examples are ASKCOS,[11,12] Chemical.AI[13] and Molecule.One.[14]

In comparison to their predecessors, ML-based synthetic route generation methods are significantly faster. An example for an open-source implementation is the AiZynthFinder[15] method. Given their lead structures the medicinal chemist receives a collection of suggestions to start synthesis planing. However, after settling on one or a few specific routes, subsequent changes in the structure can no longer be taken into account. In the context of the lead structure optimization process this is often not ideal. Relying solely on synthetic pathway prediction software the modification of a lead structure will most often result in the need to generate completely new pathways. However, there is no guarantee that the software is able to find suitable synthetic pathways that meet the individual requirements of medical chemists. Even without the help of synthetic prediction tools the synthetic chemist has to test and adapt the applicability of their chosen synthetic pathway after each modification of the lead structure. Settling for an "ideal" route, regarding properties such as its simple feasibility in the responsible laboratory, is only practical at the end of the lead optimization process, given such a synthetic pathway can be found for the structure at this point. Consequently, earlier choices of synthetic routes have to be adapted during the lead optimization process. Automated selection of molecular modifications and their effects based on their compatibility with the ideal route is not yet possible.

There are methods, which already combine de novo design techniques together with synthetic route prediction. Usually they follow a generative design and employ reaction-based retrosynthetic rules to fragment query molecules and reassemble new products.[16−20] To date, these approaches are restricted by the size of their cutting rules sets, resulting in limited consideration of the chemical environment or the entire molecule, which limits the predictability of the accessibility of the synthesis. There are also reaction-driven de novo design methods which utilize ML techniques to learn forward enumeration to design synthesizable molecules.[21,22] To our knowledge, there is no method which utilizes the entire retrosynthetic tree of a molecule as a guide to find user-defined structural modifications that result in an optimized lead compound without compromising the synthesizability of the structure.

Here, we introduce Synthesia, a new approach for synthesis-aware lead structure modification. Synthesia utilizes a specified retrosynthetic route of a lead compound to generate modified analogues without losing the feasibility of the route. The structural analogues are constructed by exchanging precursor compounds in the retrosynthetic pathway, followed by forward synthetic reconstruction. Possible substitutes not only have to be compatible with the retrosynthetic route, but must also have the ability to modify specified molecular properties in the desired direction. Thereby, the synthetic route is preserved, meanwhile calculable compound properties related to ADMET profiles, are adapted. We show the successful application of Synthesia with diverse generic modification goals: for example, "Given a synthetic pathway for a lead structure and a set of building blocks, show me all possible structures that are more hydrophilic than my lead structure." Synthesia is able to generate a set of modified structures along with applicable retrosynthetic routes for all specified property values or distributions. The generated retrosynthetic routes are based on the initial given synthetic pathway. The sequence and types of the reactions specified in the pathway are preserved, while structural changes are introduced to generate the analogues. In addition, we use Synthesia to perform a cluster analysis of two patent structure series according to their compatibility with retrosynthetic routes. The application of the method as a basis for maximizing the synthetic efficiency of multiple structures and the possible estimation of the expected effort for their synthesis is demonstrated.

■ METHODS

In the following, the exchange procedure used to modify and optimize the target structure while preserving the given retrosynthetic pathway is discussed in detail. As a basis of Synthesia, the underlying structure used for the representation of the retrosynthetic pathways, the internal application of the reaction schemes onto reactants, and the generation of new products are introduced.

The method expects as input the target structure together with at least one fully formulated retrosynthetic pathway and a set of suitable building blocks. It is advisable to use building blocks, which are commercially or in-house available or trivial in their synthesis. With the choice of the retrosynthetic pathway the synthetic chemists has the opportunity to further integrate his expertise into the process of structural modification. Pathways that have been tested in practice and are feasible in one's own laboratories or routes with a high success rate are good starting points. If no route is available, publicly available tools such as AiZynthFinder[15] can be used to attempt to create routes.

Retrosynthetic Route Representation. Retrosynthetic pathways are represented using a retrosynthetic tree structure. The rooted, bipartite tree contains compound and reaction nodes, starting with the target compound in the root node. Internal nodes represent intermediate compounds or reaction schemes, whereas leaf nodes contain only reactants. Edges between the nodes represent parent−child or reaction−reactant relationships. The tree has a hierarchical structure, where each compound is followed by a reaction scheme (except for the root) and each reaction scheme is followed by a compound. The parent node of a reaction is always its resulting product and the children of a reaction are always its reactants. Figure 1 shows the representation of an abstract retrosynthetic tree. The expected input file format for a retrosynthetic route can be found in the Supporting Information (see S2). The expected format for the generic reaction schemes is the SMIRKS[23] language, and that for

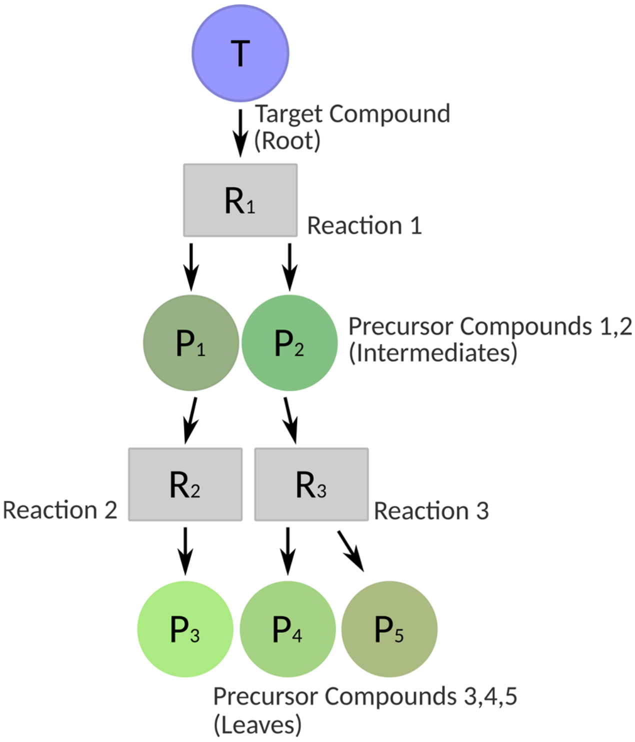

T
Target Compound (Root)
R1
Reaction 1
P1
P2
Precursor Compounds 1,2 (Intermediates)
R2
R3
Reaction 2
Reaction 3
P3
P4
P5
Precursor Compounds 3,4,5 (Leaves)

Figure 1. Example of a retrosynthetic tree representation. Circles represent compounds, gray rectangles represent reaction schemes. The blue circle illustrates the target compound in the root of the tree. Green circles are intermediate or reactant compounds.

the chemical nodes is the SMILES[24] language. Both will be discussed in more detail in the next section.

Reaction Parsing and Application. The application of reaction schemes onto reactant compounds and the generation of the resulting product compounds(s) is a central component of this work. This functionality is required for the modification of the target compound with constant verification of the validity of the given retrosynthetic pathway. In order to provide an easily readable and established interface for the parsing of reaction schemes and a compatible representation of the reactant compounds, we have chosen two of the most popular line notations: SMIRKS[23] and SMILES.[24] Both pattern languages have minimal space requirements without losing the capability to display the most important information about their underlying base data. They are widely used and established in the in silico drug development.

In the following, we briefly describe the two pattern languages and are only going into detail, where our internal definition and implementation differ from the original one provided by Daylight.[23,24] SMILES is a language to provide line representations of chemical compounds. SMIRKS is a specification of the SMARTS language,[25] which itself is an extension of SMILES to describe molecular patterns. SMIRKS is used to express generic reaction schemes. A SMIRKS pattern consists of at least two restricted SMARTS patterns describing the structural requirements for a reactant compound and the resulting product compound. In the following we will refer to these as “ReactantSMARTS” pattern and respectively “ProductSMARTS” pattern. SMIRKS describes generic reactions as a set of atom and bond changes, which are decoupled from any specific molecular structure. Any additional information, for example “electron-donating groups” are included in the atom/bond queries using the SMARTS pattern language. All changes are displayed by using an atom mapping with indices. An index of an atom in the ProductSMARTS pattern indicates that this atom originated from one of the reactant compounds. All atoms described in the ProductSMARTS pattern without an index originate in the reaction itself. These atoms will be created and added to the product compound during an application of the SMIRKS pattern onto reactant compounds. Corresponding new bonds from atoms with no index to other atoms will be created as well. All atoms in the ReactantSMARTS pattern without an index are cut off during the reaction and are not included in the resulting product compound. Bonds adjacent to atoms without indices are removed from the resulting product compound as well.[23] In Figure 2 the application of a SMIRKS pattern onto reactant compounds and the construction of a new product compound is shown.

SMIRKS patterns follow specific rules[23] to define generic reactions. For this work, we extended the list of basic SMIRKS rules to guarantee conflict-free parsing and compatibility with the internal data representations of the underlying cheminformatics engine (NAOMI[26]), as well as compatibility with the described structure of a retrosynthetic tree. In the following we list all additional rules:

- Each node in the ProductSMARTS pattern must describe exactly one element. Alternatively, only a wildcard (*) is accepted. Additionally, information about charge and number of attached hydrogens can be included.
- Labeled nodes must have the same element in the ReactantSMARTS and ProductSMARTS pattern. The only exception allowed is a ProductSMARTS node with a wildcard as element specifier.

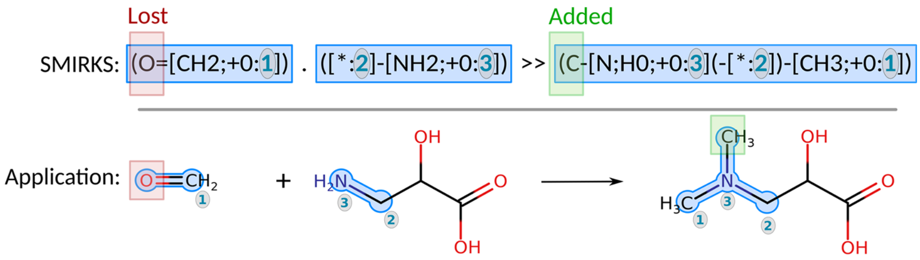

Lost
Added
SMIRKS: (O=[CH2;+0:1]) . ([*:2]-[NH2;+0:3]) >> (C-[N;H0;+0:3](-[*:2])-[CH3;+0:1])
Application:
O
CH2
1
+
H2N
3
2
OH
O
OH
CH3
OH
H3C
N
1
3
2
O
OH

Figure 2. Example application of a SMIRKS pattern onto reactant compounds resulting in a new product compound. Illustrated are a SMIRKS pattern above three compounds on which the pattern matches. The matching substructures are highlighted in blue. Atoms which will be cut during the reaction are highlighted in red, atoms which will be added during the reaction are highlighted in green. The labeling of the atoms in the SMIRKS pattern and the corresponding atoms in the compounds is highlighted with gray circles.

https://doi.org/10.1021/acs.jcim.2c00246

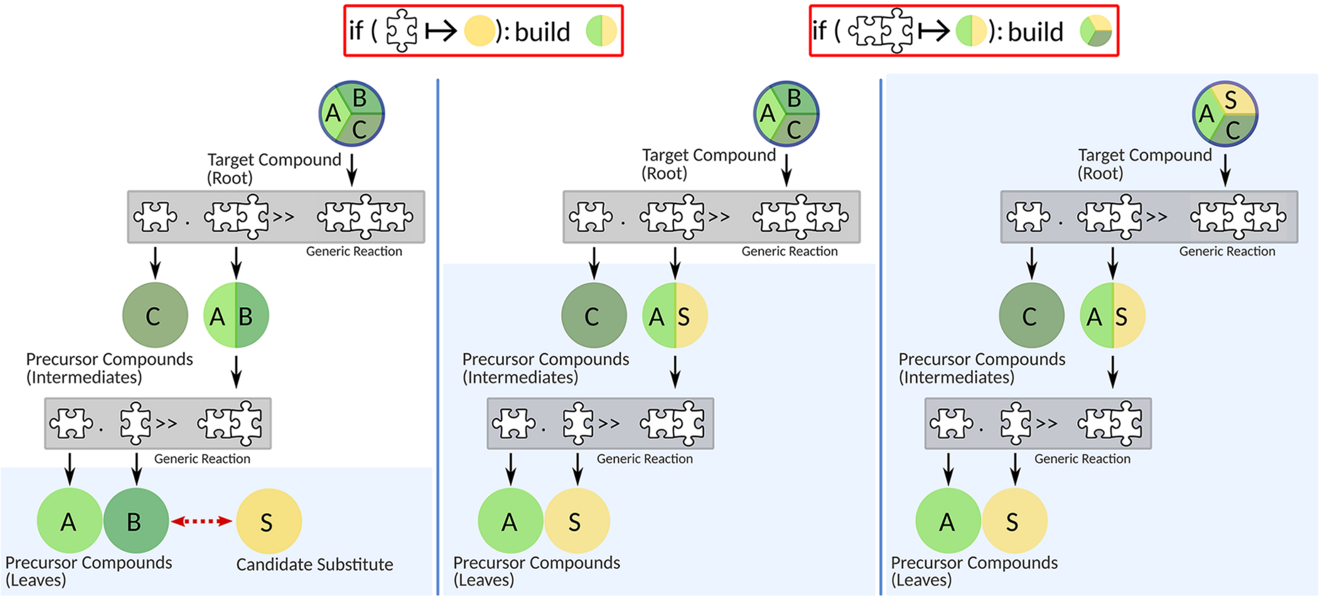


Figure 3. Simplified visualization of the modifications in a retrosynthetic tree due to the exchange of a reactant (leaf) compound. Three different states of the procedure, illustrating the structural impact of the substitute compound traversing the tree upward (blue rectangles), are shown. Circles represent compound nodes, gray rectangles illustrate reaction schemes. The potential substitute is colored yellow. All remaining compounds are green. The red outlined rectangles between the three states visualize step 1 and 2 of the described procedure.

- Regarding the product pattern only: Implicit bonds are considered as single bond; that is, all non-single bonds must be set explicitly.

Note, at the time of publishing stereo information handling is not yet supported.

Structural Optimization with Retrosynthetic Trees. To optimize the target compound situated in the root of a retrosynthetic tree without losing the described pathway, modifications at deeper levels of the tree are needed. In theory, modifications of all nodes are possible. However, any modifications will have an immediate structural influence on all following nodes traversing upward in the tree and will eventually change the root compound. The challenge is to control these structural influences and guide the resulting modifications toward the desired optimization goal. For now, we solely focus on the modification of chemical compounds within the tree. Modifications of reactions are possible too, but will be the topic of future work. During every modification step the validity of the retrosynthetic tree has to be guaranteed. A tree is considered valid, if all compounds associated with the children and parent nodes of a reaction node can be matched with the ReactantSMARTS and ProductSMARTS, respectively. Hence, for every modification at a deeper level of the tree all resulting changes traversing the tree upward have to be calculated and verified. To achieve optimization of the target compound in the root of the tree, reactant or intermediate compounds are exchanged and the influence of this exchange on the remaining compounds of the tree is calculated. This procedure will be described in detail below.

Generate Modifications of Compounds and Verify Tree Compatibility. In general, an arbitrary chemical structure can be used as a potential substitute for an exchange of a compound node of the tree. Usually, commercially available building blocks are a reasonable choice. With a list of potential substitutes, the selection of suitable reactants and the calculation of the impact of their exchange with a single compound node starts. The following steps have to be performed for each potential substitute:

1. Verify that the ReactantSMARTS pattern corresponding to the compound node that is open for replacement can be matched with a potential substitute. If not, dismiss the candidate and move on to the next one.
2. Use the whole SMIRKS pattern stored in the parent node of the specified compound together with the remaining reactant compounds and the candidate substitute to generate the modified product compound. This step includes the implicit verification that the newly generated modified product compound is compatible with the ProductSMARTS pattern.
3. Replace the compound of the parent node of the current reaction with the newly generated product compound and start over with step one, where the substitute is the newly generated compound and the specified compound node is the old parent node.
4. Continue until a modified root compound is constructed or the substitute has to be dismissed due to incompatibility with a reaction scheme of the tree.

In Figure 3 an example of the described process on a simplified retrosynthetic tree is shown.

Specifying Desired Modifications. In order to direct the modifications of the target compound toward an optimization goal the desired properties of the modified compound can be defined as search query constraints. So far, 29 possible constraints are integrated. A full description can be found in the Supporting Information (see Table S1). Since constraints are tested explicitly on specified compounds in the tree, further constraints can be added easily. All constraints are optional and can be logically combined. In addition, it can be specified that only a subset, for example, “at least 4 out of 5 constraints”, have to be fulfilled. Search query constraints can be applied at two stages. First, they can be used as filter for the potential substitute candidates, so that for example only substitutes which are more hydrophilic than the original reactant compound are considered. Second, they can be applied to the modified product compound so that for example only new product compounds are returned, which are more hydrophilic than the original one. In a lead optimization process, constraint testing of the product reflects the application scenario in most cases. However, the first version is quite reasonable as well and can restrain the set of potential reactant compounds from the perspective of the reaction.

Generation of Optimized Trees and Scoring. All structures successfully exchanged result in a valid retrosynthetic tree, which undergoes scoring and ranking. The modified tree contains all changes in the pathway resulting from the new substitute. This includes not only the modified target compound, but also all modified intermediate and reactant compounds. Note that the general structure of the tree and the reaction schemes remain unchanged.

There are three options for the verification of the additional search query constraints. First, only the potential substitute must satisfy the specified search query constraints. Second, only the modified target compound in the root of the tree has to fulfill the specified search query constraints. Finally, a combination of both is also possible. The user has the ability to choose from all three options.

A score is calculated whenever a numerical comparison is specified as an additional search query constraint, rather than just a pass or no-pass filter. For example, if a similarity constraint is specified, the calculated similarity score will be returned. The user can choose from a variety of similarity measures. If only property-based filter constraints are specified, all matching substitutes are returned along with a 1, indicating that they passed successfully. If a combination of more than one search query constraint is used, both an average value and the individual scores of each search query constraint are returned.

Multithreading is used to search for suitable substitutes. In our experiments, we achieved a throughput in the order of roughly a thousand building blocks per second and thread. Note that the required computing time varies substantially with the frequency of pattern match, the size of the synthetic route, and several further parameters. The building blocks are parsed into chunks, and only the best 1000 results are stored permanently during the search. All other results found are printed on the fly, freeing up space and resulting in constant RAM consumption.

■ RESULTS AND DISCUSSION

In the following, we present three experiments designed to explore the performance and utility of Synthesia. First, a "proof of concept" validating the integrity of the presented methods is given. Second, we demonstrate the ability of Synthesia to create structural analogues within the defined molecular property ranges while maintaining their theoretical synthetic accessibility. Third, we present Synthesia as a means to maximize the synthetic efficiency by analyzing the synthetic compatibility of a series of structures with specific retrosynthetic pathways. Therefore, we performed the analysis with two structural series of patents that select the most suitable pathways from a set of retrosynthetic pathways to create all given compounds with as few pathways as possible.

Data Sets. For all following experiments and analyses we used the building blocks of the EnamineREAL Space,[27] which are in stock in Europe (214,557 structures, Enamine-Full-EU) as potential substitute candidates. In addition, we generated a "lead structure set" of almost 500 target compounds with their retrosynthetic routes. Half of this set is derived from all structures of the DrugBank clustered by dissimilarity. We chose a random but uniformly distributed set of 247 structures from these clusters. The second half of our target compound set consists of the most similar structure for each selected compound of the DrugBank found in the EnamineREAL Space. For this task we made use of the tool SpaceLight.[28] For the last experiment we used two series of structures extracted from two patents as test sets named Daurismo[29] and CDK7.[30] Daurismo is a benzimidazole derivative used for the treatment of acute myeloid leukemia. We extracted a total of 71 structures from the patent, including the approved drug molecule. The second patent includes pyrazolo-triazine derivatives described as cyclin-dependent kinase (CDK7) inhibitors. Their main field of application is infectious diseases. We extracted a total of 155 structures from the patent. All needed retrosynthetic pathways were generated by AiZynthFinder,[15] an open-source software for retrosynthetic planning. For the generation of the retrosynthetic pathways we used the default parameters together with the given AiZynthFinder pretrained model. For training they used a reaction template library based on the publicly available US patent office (USPTO) data set,[31] and as building blocks a set of compounds based on the ZINC[32] database. Note that since the quality of the generated routes is not important for the planned experiments, we did not perform further investigations. Moreover, we want to show the applicability of our method for more arbitrary retrosynthetic routes. However, we would like to point out that Synthesia is not able to evaluate given synthesis routes. The method relies on the user's ability to select suitable routes. All targets with their retrosynthetic routes can be found in the Supporting Information (see Tables S6–S110). Note that in all following experiments only leaf nodes were open for exchange; no intermediate nodes were considered.

Proof of Concept. As a first step we verified that Synthesia is able to recreate the original lead structure if presented with suitable substitute candidates. Therefore, we randomly chose 100 retrosynthetic routes from the described DrugBank data set and used all reactant compounds saved as leaves in the chosen trees as substitute candidates. Only structures from the substitute candidates set can be selected to construct the optimized or, in this case, the original lead compound in the root. Compounds stored in the node currently open for exchange are only accessible for the comparison with the possible substitute candidates. Synthesia was able to reconstruct all original lead compounds demonstrating that the method's search and forward synthetic reconstruction procedure did not compromise the integrity of the retrosynthetic route.

Evaluation of Generalized Modification Goals. In a second experiment we investigate the ability of the presented algorithm to accomplish generalized modification goals for different lead compounds while maintaining the synthetic accessibility. Therefore, we determined the number of successfully modified compounds whose molecular properties fall within predefined ranges and are compatible with given retrosynthetic routes. We randomly chose 14 out of the available 29 search query constraints with various application types for "single-filter" runs. In addition, we designed two advanced search queries by applying the Rule-of-Five[33] and Rule-of-Three[34] as constraints. All settings can be found in the Supporting Information in Tables S3, S4, and S5. As input we used 100 randomly chosen target compounds from our lead structure set together with their retrosynthetic routes and the Enamine-Full-EU data set as potential substitute list. All chosen

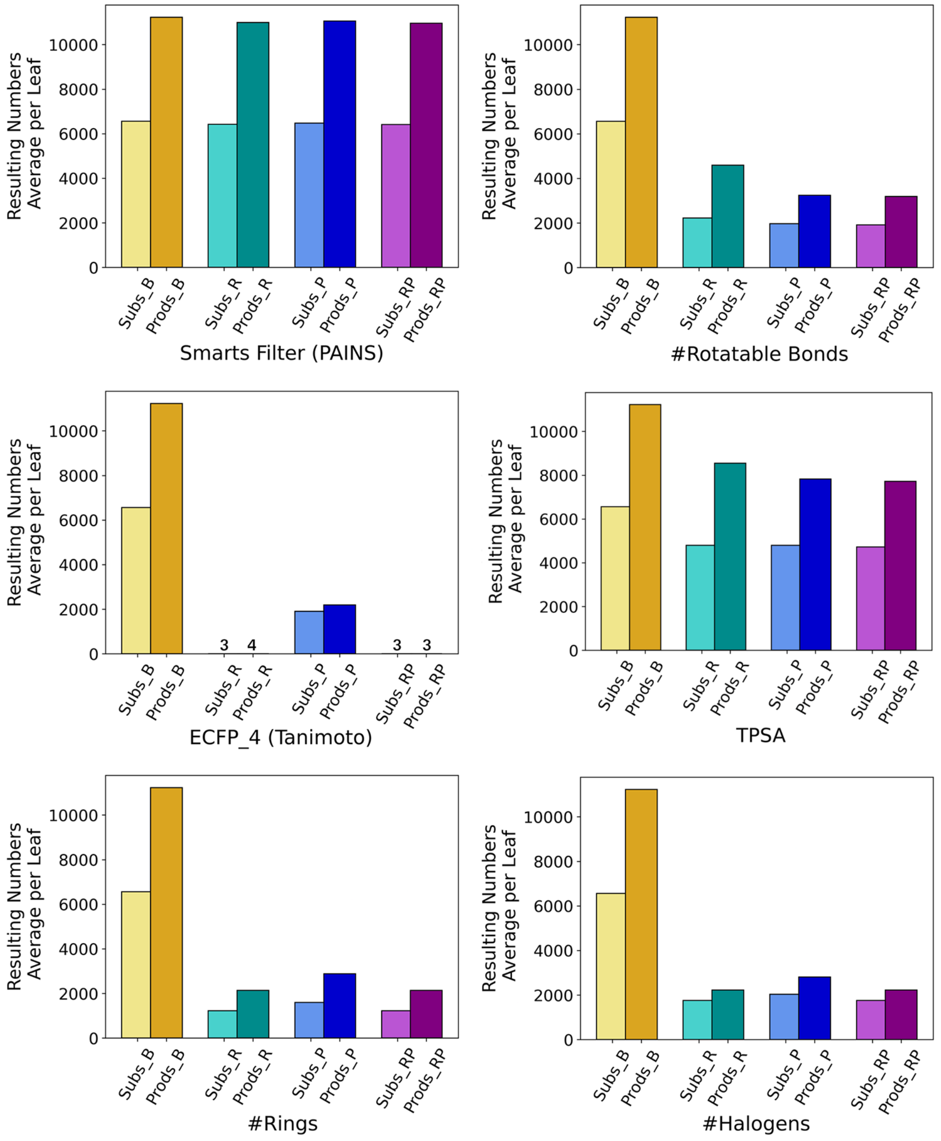


Figure 4. Results of six different single constraint runs, with the Enamine-Full-EU data set as possible substitutes and with 100 target structures of our lead-structure set. The applied search query constraint types are listed under the diagrams. All specific settings can be found in the Supporting Information. Subs_B, number of found substitutes of the baseline; Prods_B, number of resulting products of the baseline; Subs_R, number of found substitutes with search query constraints applied to reactants; Prods_R, number of resulting products with search query constraints applied to reactants. In the case for which the resulting numbers are too small to be displayed visually, we have added the numerical representation above the bar markers.

target compounds with their retrosynthetic routes can be found in the Supporting Information (Tables S6−S105) All reactant compounds, saved as leaves in the retrosynthetic trees, of all 100 routes were at the disposal for an exchange. To enable a more objective assessment of the resulting numbers we first started Synthesia without any search query constraints; thus, no modification goals were pursued. Thereby, we created a baseline (_B) to identify all compounds that can be used as substitutes without compromising the validity of the retrosynthetic route. Accordingly, these are the maximum achievable values and these can be considered as the upper limit for all further results. For each target, three different runs were started: First, the search query constraints were applied only to the potential substitutes (_R). Second, the constraints were only applied to the modified product compounds (_P). Third, the search query constraints were applied to both (_RP) the reactants as well as the products. Only the products resulting from the application methods _P and _RP are guaranteed to meet the defined molecular properties. With the _R application method, only the selected substitutes are guaranteed to meet the properties. This may or may not result in products that match the specified properties. The _R application method should be used when reactant compounds are to be replaced by substitutes with specific structural properties, with the resulting analogues being secondary. For each run both the number of found substitutes and the number of resulting products were saved. Note that the number of resulting products may be higher than the number of substitutes found, due to the ambiguous matching of the SMIRKS pattern. Figure 4 shows the results of 6 out of the 15 different runs, and Figure 5 shows the results of the two advanced search query constraints. All results can be found in Tables S106, S107, and S108 in the Supporting Information.

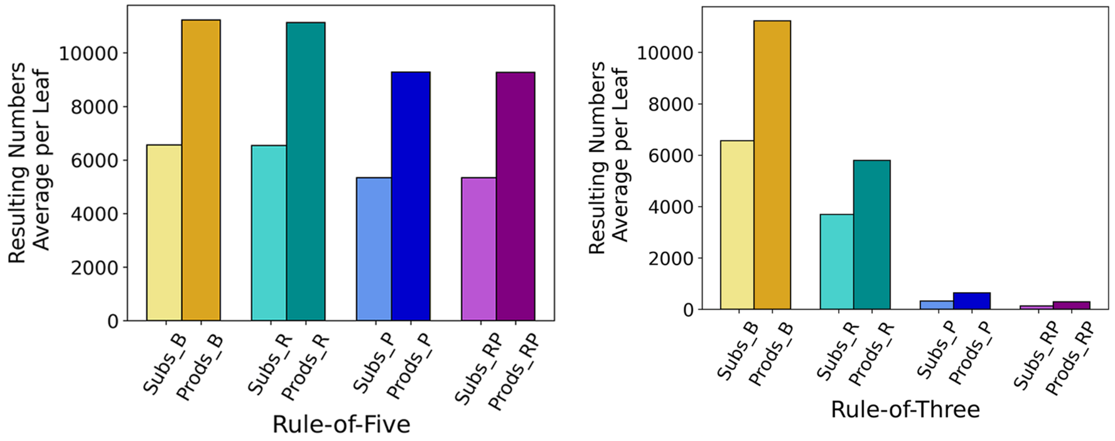


Figure 5. Results of the two advanced search query constraints with the Enamine-Full-EU data set as possible substitutes and with 100 target structures of our lead-structure set. The applied search query constraint types are listed under the diagrams. All specific settings can be found in the Supporting Information. Subs_B, number of found substitutes of the baseline; Prods_B, number of resulting products of the baseline; Subs_R, number of found substitutes with search query constraints applied to reactants; Prods_R, number of resulting products with search query constraints applied to reactants.

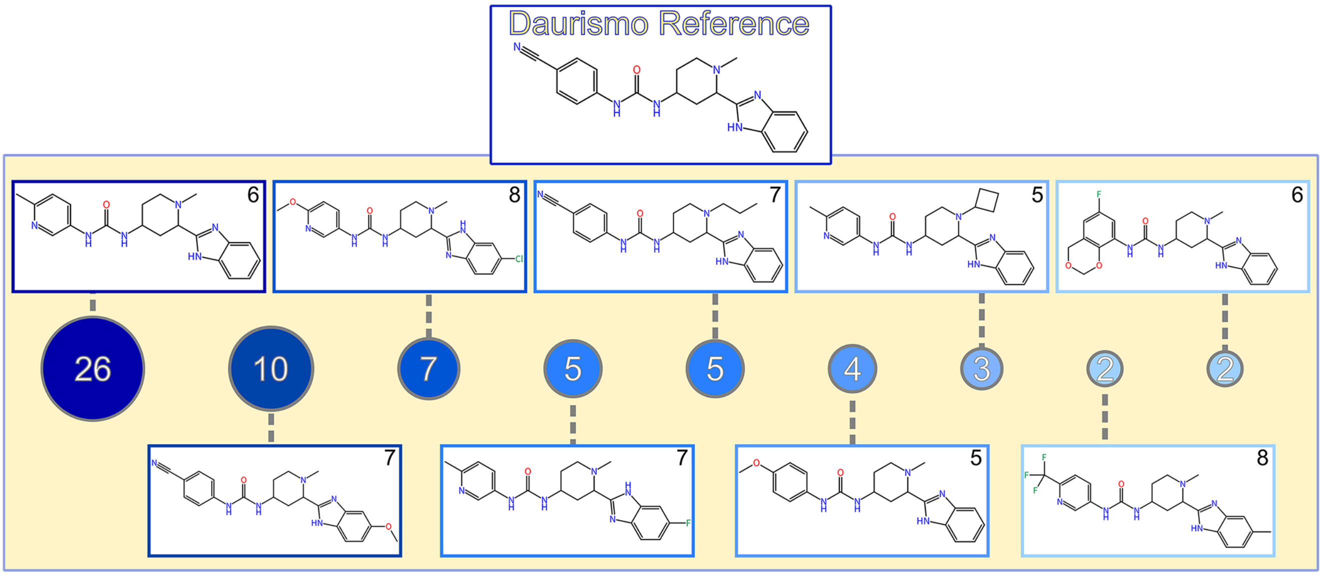


Figure 6. Minimum clustering of the retrosynthetic routes of the patent structures of Daurismo. Visualized are the size of the clusters together with the structure for which the common retrosynthetic pathway was originally calculated. Singletons are not displayed. Above is the structure of the approved drug molecule. The number of synthetic steps of each route is displayed in the upper right corner of their main structure.

Constrained optimization inherently has to address the trade-off between constraints too restrictive ending in no results or too loose ending in too many. Due to the sheer size of chemical space, however, the general trend in all presented search query applications is that even with the most restricting examples, substitutes can be found and products which fulfill the given constraints can be generated. Furthermore, there are several trends recognizable in the resulting numbers. The application of the filters to both the reactant and the product structures is the most restrictive. Whether constraints are more restrictive on the reactant or product level obviously depends on the constraint itself. Overall, as expected, the number of results highly depends on the given constraint type. Looking at the results of the SMARTS Filter constraint using SMARTS representations of the Pan-Assay Interference Compounds (PAINS)[35] one can see that all resulting numbers are almost as large as the baseline numbers. This is not surprising since the substitute data set used, the Enamine-Full-EU data set, itself rarely contains a structure that matches a PAINS pattern. In contrast to these results are the numbers of the ECFP_4 constraint, where only a few substitutes could be identified and transformed into products which respectively fulfill the similarity requirement. This is again an expected trend: the ECFP_4 descriptor together with a threshold of 0.6 for the Tanimoto similarity measure comparison is more restrictive and therefore selective. For the given data set it is apparently easier to find a substitute compatible with the tree and results in a product similar to the given target structure than a substitute which is similar to the reactant structures while still compatible with the retrosynthetic tree.

When the results of the application of the Rule-of-Five and the Rule-of-Three are compared, the expected lower restrictiveness of the Rule-of-Five is confirmed. This was expected since the Rule-of-Five is less restrictive than the Rule-of-Three in three

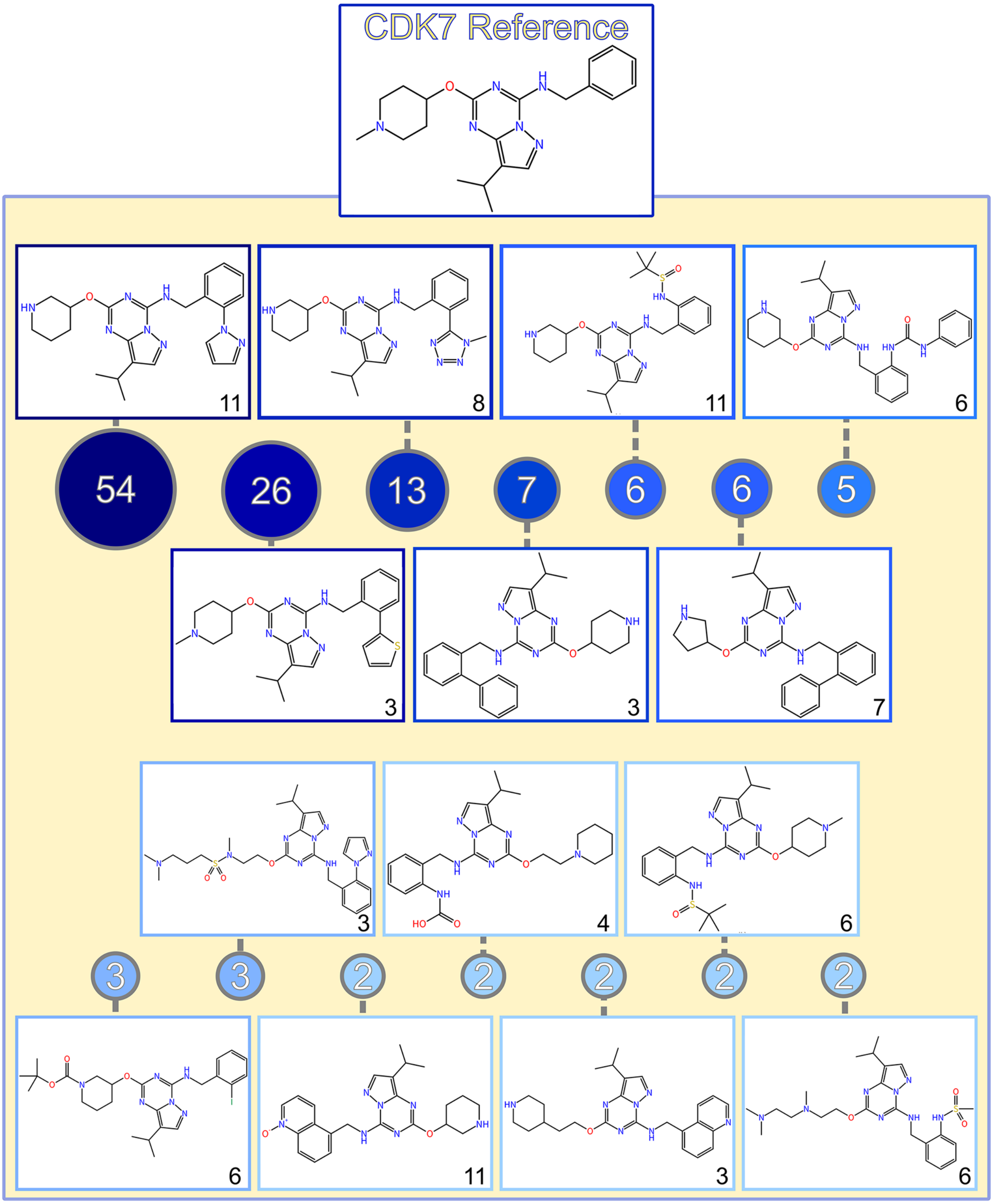


Figure 7. Minimum clustering of the retrosynthetic routes of the patent structures of CDK7. Visualized are the size of the clusters together with the structure for which the common retrosynthetic pathway was originally calculated. Singletons are not displayed. Above is the structure of the approved drug molecule. The number of synthetic steps of each route is displayed in the lower right corner of their main structure.

points: it consists of two fewer search constraints, only three of the specified four constraints must be met, and the thresholds are higher and therefore easier to fit than the ones of the Rule-of-Three. With the successful application of the Rule-of-Five and Rule-of-Three as search query constraints, it is shown that even

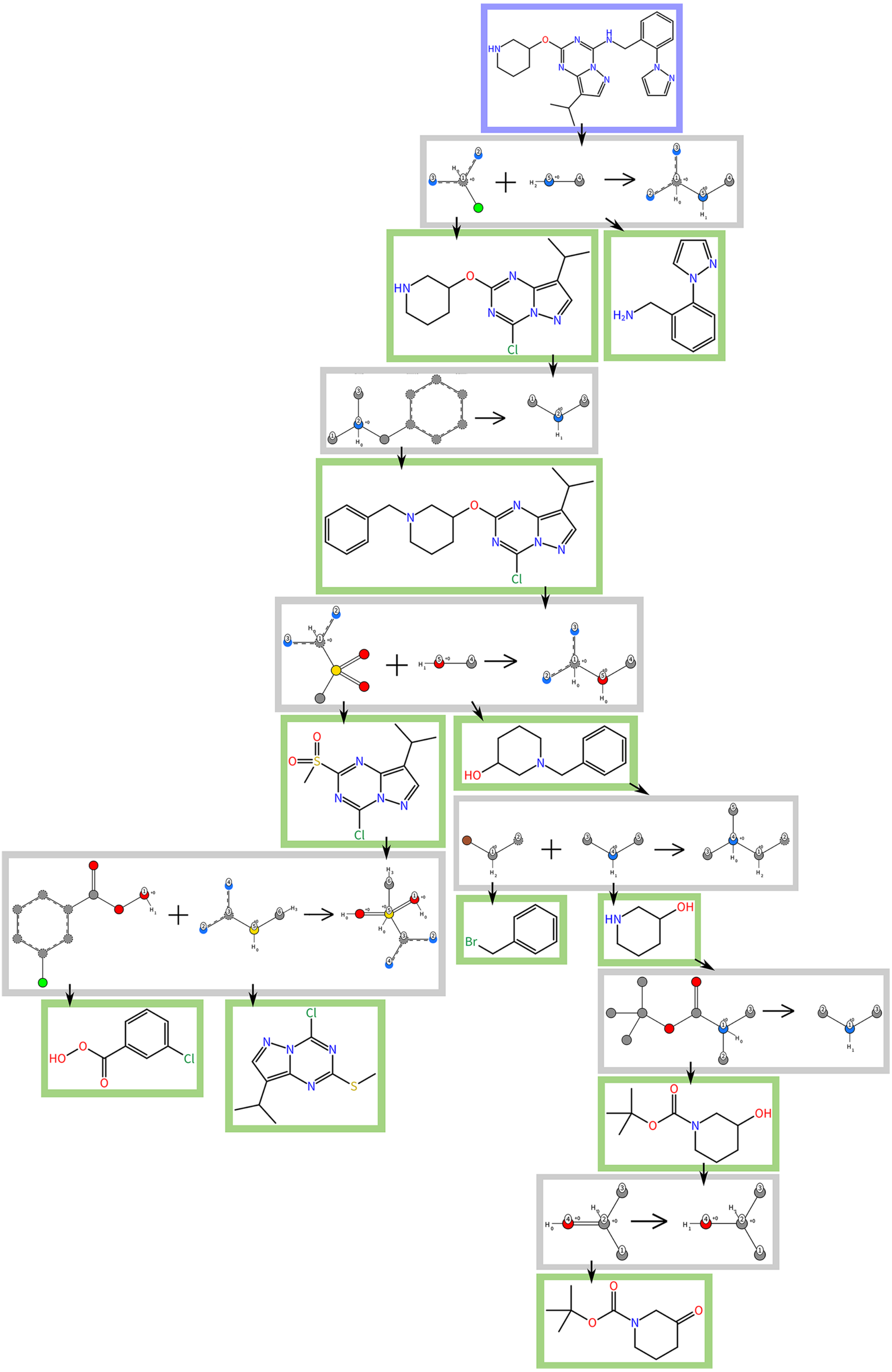


Figure 8. Retrosynthetic route of the compound Cc1ccc(NC(=O)NC2CCN(C)C(c3nc4ccccc4[nH]3)C2)cn1 generated by AiZynthFinder.[15] This route was used for the largest cluster of the CDK7 patent structures and can be adapted to generate 54 of the 155 structures. Green rectangles contain compounds, gray rectangles SMIRKS pattern. The target compound is in the blue rectangle in the root of the tree.

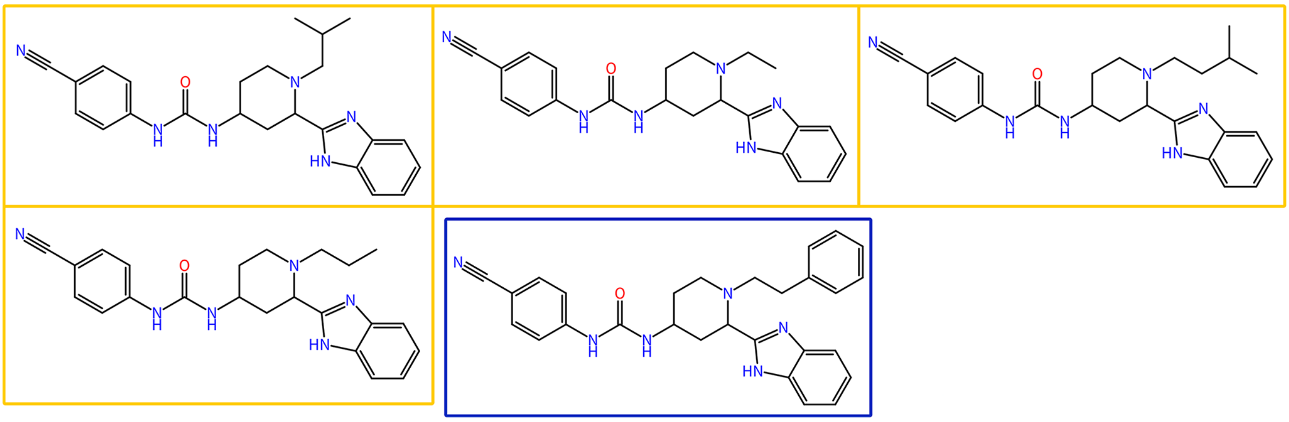


Figure 9. Example of all structures (yellow rectangles) of one cluster of the Daurismo cluster analysis with Synthesia. The corresponding retrosynthetic route was generated for the compound O=C(Nc1ccc(C#N)cc1)NC2CC(N(CC2)CCC)C3=Nc4c(N3)cccc4 (blue rectangle).

more specialized optimization goals can be realized and used to find appropriate analogues.

Retrosynthetic Route Cluster Analysis. As a final application scenario, we performed an analysis in which we investigated the minimum number and distribution of retrosynthetic pathways required to create all active structures of a patent series. Therefore, we calculated all possible substitutes for all retrosynthetic pathways for all structures of the patents and created all possible target structures compatible with the retrosynthetic pathways. The Enamine-Full-EU data set was used as potential substitute list. By this, we created a new set of "family" target structures compatible with their retrosynthetic pathway for each structure. These family-structure-sets were then searched for the original target structures from the patent, resulting in a clustering of the original target structures according to their retrosynthetic pathways. It should be noted that a structure can belong to more than one cluster if it can be synthesized via more than one retrosynthetic pathway. We extracted the minimum number of clusters, with the constraint that each original structure must be contained in at least one cluster. This resulted in 16 clusters for the structures of the Daurismo patent and 36 for the CDK7 patent. The largest cluster of the Daurismo patent structures contains 26 structures; the largest cluster of the CDK7 patent structures contains 54 structures. Each cluster represents the largest number of structures which could theoretically be synthesized with the same retrosynthetic route, based on the available set of retrosynthetic routes given by the AiZynthFinder tool. Note that the results may differ significantly if a different set of retrosynthetic routes is given. The Daurismo clustering includes seven singletons and the CDK7 clustering 21. All structures placed in a singleton cannot be synthesized by the given remaining routes of the other structures of their patent. Figure 6 visualizes the clustering for the Daurismo patent; Figure 7 does the same for the CDK7 patent. The resulting cluster sizes along with the structure for which the common retrosynthetic pathway was originally calculated are shown. Singletons are not visualized. Tables S111 and S130 store all resulting clusters with their sizes along with the SMILES of the structure providing the retrosynthetic route.

Both cluster analyses revealed that for each patent structure series more than one-third of the structures can be synthesized by the same retrosynthetic pathway. In addition, only few structures are so unique that they require a specific retrosynthetic pathway. At this point, we would like to remind the reader that we used retrosynthetic routes generated by the AiZynthFinder method. We did not extract the synthetic routes from the original literature, which would likely have resulted in even larger clusters with fewer singletons. We intentionally performed the analysis with more arbitrary routes to model a more realistic scenario where the synthetic chemist may wish to test their simple or field-tested protocols for compatibility with a range of target compounds and where a fully formulated synthesis plan is only available for a few compounds, if any. The strategy of exchanging leaf structures of the retrosynthetic tree leads to a minimization of synthetic steps by maximizing the number of common steps and adding the main modifications at the lowest levels of the tree. Considering the largest cluster of the CDK7 structures, which contains 54 compounds and a retrosynthetic route with 11 synthetic steps, one can estimate that ten steps can be classified as common, with only the last retrosynthetic step showing the largest differences. This results in 10 + 54 different synthesis steps that must be performed to generate all 54 structures. This is significantly more efficient than the approximate 11·54 synthesis steps that would be required if a separate route were required for each structure. The corresponding route is visualized in Figure 8. Figure 9 shows one cluster of the Daurismo cluster analysis. All matching structures as well as the original structure for which the corresponding retrosynthetic route was generated are visualized.

Note that the retrosynthetic route is input for Synthesia and not altered except the selection of the final reactants. Whether the reaction is possible with the newly selected reactants can only be checked up to the level of the corresponding SMIRKS. Therefore, the theoretical analysis shown here needs to be proven and controlled by experiments, as is usual for in silico analyses. Retrosynthetic route clustering in such a way could benefit the medical chemist in two ways. First, it can help estimate the likely effort and expenses to synthesize a series of structures. Second, the synthetic chemist can choose simple or field-tested retrosynthetic routes and use Synthesia to determine how many of the target structures can be synthesized with the chosen ones.

■ CONCLUSION

Lead structure modification toward a drug candidate is a complex task in the drug development process. One crucial part for the successful design is the maintenance of synthetic accessibility of the suggested compounds. However, this property is often decoupled from the actual modification process and only considered in the late phase. Therefore, a lot of virtually designed structures are hard to synthesize or not at all synthetically accessible and have to be rejected.

In this work we presented Synthesia, a method to realize and support synthesis-aware lead structure modification. Synthesia is able to create structural analogues for a lead structure which meet user-defined molecular properties while ensuring their synthesizability by utilizing the synthetic routes as pathway to guide the process of structural modification. With the results of the proof-of-concept and the generalized search query constraint application experiment we can state that Synthesia fulfills the desired design goal. The method has proven to be successful in the modification and optimization of lead structures in a specified direction without compromising the applicability of their retrosynthetic pathway. Even for advanced modification goals, including the adaption of multiple molecular properties, Synthesia was able to identify suitable substitutes and build the corresponding target structures.

In addition, Synthesia has been successfully used in another interesting application scenario: cluster analysis of retrosynthetic routes for structure series, where the minimum number of retrosynthetic routes required to generate all structures in the series is determined. The analysis was performed for two compound series from patents, one containing structure variants of Daurismo, the second containing structurally related CDK7 inhibitors. The analysis revealed that for the Daurismo series a minimum number of 16 retrosynthetic routes is required to synthesize all 71 structures. For the CDK7 series, the minimum number of routes found were 36 for 156 structures. For both series, the largest cluster contained more than one-third of all structures. All resulting numbers refer to the available retrosynthetic routes generated by the AiZynthFinder tool. The results demonstrate the ability of our tool to identify the most useful retrosynthetic routes for the theoretical synthesis of a given set of compounds. Analysis of this type could be highly relevant to quickly estimate the approximate effort and costs that would be involved in fully synthesizing all structures in a series of newly suggested compounds, or to test the compatibility of a set of designed target compounds with field-proven or proprietary synthesis protocols.

In future work, we will extend the functionality of Synthesia by the possibility to not only exchange reactant structures, but also to provide alternative reaction schemes. A substitute reaction scheme will be selected from a variety of reaction schemes of the same class, using reaction classifiers such as NameRxn.[36] The substitution will be presented to the user if it leads to the modification of the target compound in the specified direction or if the exchanges increase the synthetic efficiency of the overall route without obstructing the modification goal. Thereby, we hope to allow medical chemist to explore a larger structural analogue space without compromising the modification goal of their lead structure.

With the results of this work, we have outlined a computational method for goal-oriented lead structure modification under direct consideration of the synthetic route. The method can directly support the medicinal chemists during lead structure modification, but might also be of interest in automated computational workflows. The resulting tool Synthesia can be seen as a technical key element toward further automation of the drug design process.

■ SOFTWARE AND DATA AVAILABILITY

Synthesia is available for Linux and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and is free for academic use and evaluation purposes.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c00246.

Overview and detailed information on all available search query constraints; a template for the expected file format of the retrosynthetic tree(s) and additional information and results of the experiments (PDF)

List of compounds used in the proof of concept experiment and the retrosynthetic routes of the generalized filter and cluster experiments (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *Universität Hamburg, ZBH - Center for Bioinformatics, Hamburg 20146, Germany;* orcid.org/0000-0002-9553-6531; Phone: +49 (40) 428387351; Email: matthias.rarey@uni-hamburg.de

Authors

Uschi Dolfus − *Universität Hamburg, ZBH - Center for Bioinformatics, Hamburg 20146, Germany;* orcid.org/0000-0002-2720-1086

Hans Briem − *Bayer AG, Research & Development, Pharmaceuticals, Computational Molecular Design Berlin, Berlin 13342, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c00246

Author Contributions

U.D., H.B., and M.R. developed the Synthesia concept. U.D. implemented the Synthesia approach. M.R. supervised the project, all authors participated in manuscript writing.

Notes

The authors declare the following competing financial interest(s): M.R., as a shareholder of BioSolveIT GmbH, declares a potential financial interest in the event that the Synthesia software is licensed for a fee to non-academic institutions in the future.

■ REFERENCES

(1) Fischer, J.; Ganellin, C. R. Analogue-based drug discovery. *Chem. Int.—Newsmag. IUPAC* **2010**, *32*, 12−15.

(2) Vleduts, G. Concerning one system of classification and codification of organic reactions. *Information Storage and Retrieval* **1963**, *1*, 117−146.

(3) Corey, E. J.; Wipke, W. T. Computer-Assisted Design of Complex Organic Syntheses: Pathways for molecular synthesis can be devised with a computer and equipment for graphical communication. *Science* **1969**, *166*, 178−192.

(4) Hoffmann, R. *Elements of Synthesis Planning*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2009; pp 145−148.

(5) Corey, E. J.; Long, A. K.; Rubenstein, S. D. Computer-assisted analysis in organic synthesis. *Science* **1985**, *228*, 408−418.

(6) Pensak, D. A.; Corey, E. J. *LHASA-logic and heuristics applied to synthetic analysis*; ACS Publications, 1977.

(7) Thakkar, A.; Johansson, S.; Jorner, K.; Buttar, D.; Reymond, J.-L.; Engkvist, O. Artificial intelligence and automation in computer aided synthesis planning. *Reaction chemistry & engineering* **2021**, *6*, 27−51.

(8) Segler, M. H.; Preuss, M.; Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604−610.

(9) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS central science* **2017**, *3*, 434−443.

(10) Gao, H.; Struble, T. J.; Coley, C. W.; Wang, Y.; Green, W. H.; Jensen, K. F. Using machine learning to predict suitable conditions for organic reactions. *ACS central science* **2018**, *4*, 1465−1476.

(11) Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehiers, P. P.; Byington, J.; Piotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, *365*, eaax1566 DOI: 10.1126/science.aax1566.

(12) *Askcos: software tools for organic synthesis*. https://askcos.mit.edu/ (last accessed on 04/05/2021).

(13) *Chemical.AI*. https://chemical.ai/ (last accessed on 04/05/2021).

(14) *Molecule.One*. https://molecule.one/ (last accessed on 04/05/2021).

(15) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J. Cheminform.* **2020**, *12*, 1−9. last accessed on 04/06/2021.

(16) Ghiandoni, G. M.; Bodkin, M. J.; Chen, B.; Hristozov, D.; Wallace, J. E.; Webster, J.; Gillet, V. J. RENATE: A Pseudo-retrosynthetic Tool for Synthetically Accessible de Novo Design. *Molecular Informatics* **2021**, 2100207.

(17) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. Recap retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of chemical information and computer sciences* **1998**, *38*, 511−522.

(18) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem.* **2008**, *3*, 1503.

(19) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: reaction-driven de novo design of bioactive compounds. *PLoS computational biology* **2012**, *8*, No. e1002380.

(20) Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F.; Heeres, J.; Koymans, L. M.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. Synopsis: synthesize and optimize system in silico. *Journal of medicinal chemistry* **2003**, *46*, 2765−2773.

(21) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M.; Hernández-Lobato, J. M. Barking up the right tree: an approach to search over molecule synthesis dags. *Adv. Neural Informa. Process. Syst.* **2020**, *33*, 6852−6866.

(22) Gao, W.; Mercado, R.; Coley, C. W. Amortized Tree Generation for Bottom-up Synthesis Planning and Synthesizable Molecular Design. *arXiv* **2021**, No. 2110.06389.

(23) *Daylight, SMIRKS - A Reaction Transform Language*. https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html (last accessed on 25/05/2021).

(24) *Daylight, SMILES - A Simplified Chemical Language*. https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html (last accessed on 25/05/2021).

(25) *Daylight, SMARTS - A Language for Describing Molecular Patterns*. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (last accessed on 25/05/2021).

(26) Urbaczek, S.; Kolodzik, A.; Fischer, J. R.; Lippert, T.; Heuser, S.; Groth, I.; Schulz-Gasch, T.; Rarey, M. NAOMI: on the almost trivial task of reading molecules from different file formats. *J. Chem. Inf. Model.* **2011**, *51*, 3199−3207.

(27) *CoLibri: chemistry spaces*; BioSolveIT GmbH: Sankt Augustin, Germany, 2018; https://www.biosolveit.de/CoLibri/spaces.html (last accessed on Feb 2021).

(28) Bellmann, L.; Penner, P.; Rarey, M. Topological Similarity Search in Large Combinatorial Fragment Spaces. *J. Chem. Inf. Model.* **2021**, *61*, 238.

(29) Munchhof, M. J.; Reiter, L. A.; La Greca, S. D.; Jones, C. S.; Li, Q. Benzimidazole derivatives. US Patent US 20090005416 A1, 2009.

(30) Eickhoff, J.; Zischinsky, G.; Koch, U. Pyrazolo - triazine derivatives as selective cyclin-dependent kinase inhinitors. WO Patent WO 2013128028 A1, 2012.

(31) Lowe, D. *Chemical reactions from US patents (1976-Sep2016)*, 2017; https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873 (last accessed on 27/05/2022).

(32) Sterling, T.; Irwin, J. J. ZINC 15−ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.

(33) 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. *Advanced drug delivery reviews* **1997**, *23*, 3−25.

(34) Congreve, M.; Carr, R.; Murray, C.; Jhoti, H. A'rule of three'for fragment-based lead discovery? *Drug discovery today* **2003**, *8*, 876−877.

(35) Baell, J. B.; Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *Journal of medicinal chemistry* **2010**, *53*, 2719−2740.

(36) *NameRXN (Nextmove Software)*. http://www.nextmovesoftware.com/namerxn.html (last accessed on 06/12/2021).

D.3 Full Modification Control Over Retrosynthetic Routes for Guided Optimization of Lead Structures

[D3] **U. Dolfus**, H. Briem, T. Gutermuth, and M. Rarey. “Full modification control over retrosynthetic routes for guided optimization of lead structures”. In: *Journal of Chemical Information and Modeling* 63.21 (2023), pp. 6587–6597.

The following pages include the published manuscript together with the Supporting Information. The Supporting Information includes the initial retrosynthetic routes for the target structures and contains additional results of the structural space coverage between different exchange modes, and the bioisosteric linker replacement based on route compatibility. JSON files of the initial retrosynthetic routes and the a list of the generated abrocitinib analogue routes can be found here `https://doi.org/10.1021/acs.jcim.3c01155`

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Full Modification Control over Retrosynthetic Routes for Guided Optimization of Lead Structures

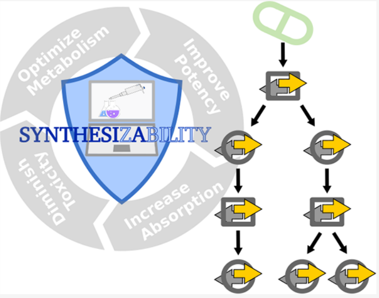
Uschi Dolfus, Hans Briem, Torben Gutermuth, and Matthias Rarey*

Cite This: J. Chem. Inf. Model. 2023, 63, 6587−6597

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Synthesizability is essential for compounds designed *in silico*. Regardless, synthetic accessibility is often considered only as an afterthought in the design and optimization process. In addition, the trend with modern computer-aided drug design methods is going toward full automation and away from the possibility of incorporating user knowledge. With this work, we present the second major release of our software tool, Synthesia, for synthesis-aware lead structure modification, where the user's expertise is now fully utilized. A provided retrosynthetic route is used as a pathway to guide structural modifications that introduce desired structural changes in the target compound. Moreover, the approach allows the user to define the exact position or component in the retrosynthetic route, which should be modified, further integrating the user's expert knowledge. This paper describes the functionality of Synthesia, its basic concepts, and several application scenarios ranging from simple examples to a comparison of the effects of the different exchange functions to an analysis of a set of bioisosteric linker structures, highlighting potential synthetically feasible replacements.



■ INTRODUCTION

Lead structure modification is crucial in modern drug development, where chemists modify the initial lead compound to improve its potency, selectivity, pharmacokinetics, and safety. It involves the rational design, optimization, and synthesis of chemical compounds to enhance their biological activity and therapeutic potential. The design and optimization process is often more in focus, whereas the synthesis is only regarded as an afterthought. Computer-aided synthesis planning tools (CASP) try to cover the complex matter of molecule synthesis. One fundamental technique is retrosynthetic analysis,[1] but other usages of modern CASP tools can include predicting reaction conditions or searching for novel reactions.[2−6] Focusing on the synthetic accessibility of target compounds, various methods already provide propositions of synthetic pathways for novel molecules. Early approaches apply heuristic reaction rules to generate retrosynthetic routes.[7,8] Modern methods incorporate machine learning to tackle retrosynthetic route prediction.[9−11] There are already complete commercial and open-source software suites that provide various methods for synthesis planning. Examples are ASKCOS,[12] Molecule.One,[13] Chemical.AI,[14] and AiZynthFinder.[15] However, these tools for retrosynthetic route prediction can be applied only after finishing the design and modification process. Subsequent changes in the structure cannot be taken into account, i.e., they require a full redesign of the synthesis route.

Synthetic feasibility calculations are to some extent already combined with the classic *de novo* design approach.[16−21] The general procedure of these tools includes the construction of new molecules through retrosynthetic reactions to fragment query structures and then the assembly of new molecules based on the fragments and the reactions. Oftentimes, the methods are based on a limited set of retrosynthetic rules, which restrict the chemical space for generated target structures.

Initial attempts with generative models are further examples of how the synthesis of generated compounds has been decoupled from the design and optimization process. Models that use SMILES LSTM, SMILES GA, and Graph GA[22−25] can lead to molecules that have been ideally adapted to the desired properties, but which might show limited synthetic feasibility.[26] There are recent advances where generative models include synthesizability as an additional or secondary objective, utilizing respective scoring functions or including CASP tools directly into the generation process.[27] Synthesizability scores[28−31] are used to quickly access the synthetic accessibility to guide molecular discovery in generative models. Combining more complex CASP tools with generative models leads to optimized molecules generated with a synthetic pathway.[32,33] For example, Bradshaw et al.[34] use generative models to create synthetic pathways as directed acyclic graphs iteratively and includedFor example, Bradshaw et al.[34] use generative models to create synthetic pathways as directed acyclic graphs iteratively and include their architecture in either latent generative model or reinforcement learning (RL) procedures to sample and optimize

Received: July 28, 2023
Revised: October 9, 2023
Accepted: October 11, 2023
Published: November 1, 2023

https://doi.org/10.1021/acs.jcim.3c01155

novel molecules. Gottipati et al.[35] proposes a related approach using Policy Gradient for Forward Synthesis, where initial commercially available molecules are subjected to valid chemical reactions during the optimization process. The resulting environment is again explored by using RL algorithms. Both types of models, based on synthetic scores or complete CASP tools, provide novel molecules that are more likely synthetically feasible. However, both are dependent on their underlying method and therefore inherent limitations. Synthetic scores are designed to estimate synthetic accessibility quickly but can cover only a limited set of aspects. CASP tools depend on the availability of training data, which can be particularly challenging for reaction data.[27]

So, on one hand, we can try to calculate synthesis routes for compounds that are already fully optimized, hoping to find ones that fit our requirements, or we can try to generate routes directly during the optimization of the molecules, hoping that the resulting molecules and the route fit. Both approaches result in ready-made solutions, leaving little or no room for further adjustments. In addition, the richest source of information, the user's expertise, is mostly neglected in this process. Here, we decided to shift the focus and present a method to modify a lead structure without losing the applicability of a predefined suitable retrosynthetic route. Instead, we present a method utilizing the retrosynthetic route as a pathway to guide the modification process. The initial algorithm is already published,[36] realizing the following approach: to create a structural analogue of the lead structure, precursor compounds in the retrosynthetic pathway are exchanged, followed by reconstruction in a forward-synthesis manner. In this process, potential substitutes are selected as being compatible with the retrosynthetic route and as being able to modify specific molecular properties in the desired direction. By doing so, the synthetic route remains intact, while the compound properties related to bioactivity or ADMET profiles get adjusted. While in this first approach modification was limited to the exchange of reactants, the method presented here enables modifications to all components of the retrosynthetic route.

The algorithm behind Synthesia presented here allows us to either specify the precise location for modifications in the route and receive suitable alternatives or specify a substructure of the target molecule to be modified, allowing the method to identify the corresponding subtree and propose modification options automatically. In addition, Synthesia allows to exchange or skip reactions, modify multiple reactant structures at once, and define a target function that allows the identification of desired or undesired substructures within the target molecule. We present examples for each type of modification and a comparison of the effects and results of the modifications between them. In addition, we use Synthesia to screen a set of bioisosteric linker structures to identify those that can be exchanged in a target structure without losing the applicability of a given retrosynthetic route. Overall, we show that the medicinal or synthetic chemist's expertise can be fully exploited, giving full control over the modifications made to the route and the target.

■ METHODS

In the following, we describe the underlying algorithms that modify and transform given retrosynthetic routes in the user-defined direction. The necessary input and parameters are specified, and the current limitations are summarized. All functions expect at least one fully formulated retrosynthetic route and a set of building blocks to be used as possible substitute candidates. When choosing building blocks, opting for commercially or in-house-available ones or those that are simple to synthesize is recommended. Regarding retrosynthetic pathways, the ones that have already been tested in practice and are feasible to replicate in one's laboratory or routes with a high success rate would be the obvious choice.

All new functions are based on the data structures described in our previous publication.[36] The retrosynthetic route is represented internally as a tree structure consisting of chemical structure nodes and generic reaction nodes in hierarchical order. The root of the tree contains the target structure. Generic reactions must be given as Reaction SMARTS[37] or SMIRKS,[38] and chemical structures must be given as SMILES.[39] The SMILES and Reaction SMARTS/SMIRKS languages have been widely used and established in *in silico* drug development. They have minimal space requirements but still contain essential information about their underlying base data. The Reaction SMARTS and SMIRKS languages especially are frequently applied in computer-aided reaction contexts and are beneficial due to their flexibility and chemical precision as generic representations of synthetic reaction rules.[21,40,41] For details, we refer to our first Synthesia publication.[36]

Modes Overview. As visualized in Figure 1, five modes are available to give the user complete modification control over a retrosynthetic route. From left to right in Figure 1, the available functions are called single exchange (SE), simultaneous multiple exchange (SME), reaction exchange (RE), reaction skipping (RS), and product exchange (PE). The modes can be divided into three categories: they allow the user to focus on the reactant structures (SE and SME), on the reactions (RE and RS), or on the target structure alone (PE). Of course, a combination of the categories is possible, too. In the following, we explain the different modes and their algorithmic realizations.

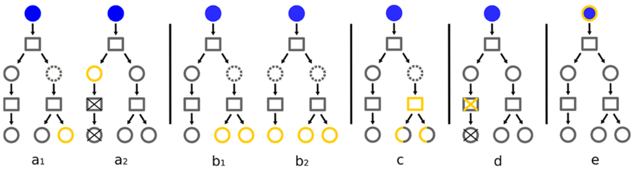


Figure 1. Generic representation of the different modes included in Synthesia. Displayed are abstract retrosynthetic routes. Circles represent chemical structures, and rectangles represent generic reactions. The target structure forming the root of the tree is marked in blue; the component of the tree open for exchange is marked in yellow. Components that are indirectly changed by an exchange further down the tree are dotted. From left to right, the following modes are visualized: (a) single exchange, (b) simultaneous multiple exchange, (c) reaction exchange, (d) reaction skipping, and (e) product exchange.

Single Exchange. The single exchange (compare with Figure 1a) is the basic function of our method. A detailed description can be found in our previous publication.[36] The algorithm consists of three steps: exchange a single reactant structure in the tree, verify the integrity of the route, and generate the modified target structure with the structural changes introduced by the new reactant. The verification is based on the given Reaction SMARTS or SMIRKS pattern associated with the respective reaction nodes. The modifications are traversed up toward the root until the target structure is adapted or a reaction is incompatible with the desired changes.

Since our last publication, we have changed and added some steps to the algorithm to improve efficiency. The algorithm performs fast filtering of the presented substitute candidates for all modes that include the exchange of reactant structures. Only structures that match the original reactant structure's corresponding SMARTS expression are considered to be substitute candidates. All other presented structures are incompatible and therefore disregarded early. To steer the structural properties of the target structure, the substitute reactants, or both in a desired direction, the algorithm provides a set of 29 physicochemical property constraints. A detailed description can be found in the Supporting Information of our previous publication.[36]

The score calculation is extended by a calculation of the percentage deviation for all constraints for which the desired property is defined not as an exact numeric value but as a range or bound. The percent deviation is calculated between the property value of the original structure and the property value of the substitute structure. If more than one structural constraint is defined, then the mean deviation is calculated. The deviation value is used to sort the resulting target structures in ascending or descending order.

Simultaneous Multiple Exchange. After realizing single reactant structure exchange in a retrosynthetic route, the logical next step is to allow the simultaneous exchange of multiple reactant structures (see Figure 1b). The corresponding algorithm, named simultaneous multiple exchange, follows the same principles as single exchange. Multiple substitutes and their corresponding structural changes are introduced to the tree with multiple reactant structures open for exchange. Therefore, the algorithm starts with all open reactant nodes and simultaneously calculates the changes traversing the tree upward instead of only regarding one subtree originating in one reactant structure. The nodes are sorted and processed in reverse topological order to ensure a correct and efficient calculation. All chemical nodes can be exchanged. If an internal node is open for exchange, the corresponding subtree originating from this node to the next leaf is disregarded.

Even though the algorithm utilized multithreading, the simultaneous multiple exchange mode is computationally expensive due to the combinatorial possibilities that need testing to guarantee the complete exploration of the possible structural space. We recommend defining structural constraints to limit the number of suitable substitute candidates. The user is informed about the number of possible combinations at the beginning of the calculations. While processing, the already checked number of combinations is displayed and updated so that the user can decide whether the current run time is acceptable or if parameters have to be adjusted. Note that for all of Synthesia's functions, the run time can vary distinctly between different runs because the calculations are highly dependent on the given input and the parameters set, resulting in a large variety of required compatibility checks and how many compatibility checks must be made.

Reaction Exchange. Exchanging reactions enables the exploration of a wider range of analogues such that molecules with different properties and characteristics can be created (see Figure 1c). In addition to the set of building blocks and a retrosynthetic route, the user must provide a list of possible reaction substitutes. Often, these can be derived from in-house laboratory journals or from public resources.[15] The algorithm executes the following four steps to exchange specified reactions and propose suitable substitutes:

1. Prefilter the set of given generic reactions and identify possible suitable substitutes.
2. Check the compatibility of the proposed substitutes with the retrosynthetic tree.
3. Create a new tree for each proposed substitute.
4. For each tree, start a reactant exchange if specified.

The prefiltering of the generic reactions depends on the availability of differentiation among the given reactions. The algorithm supports filtering based on the reaction's name or the numerical classification scheme provided by NameRxn.[42] The three-level NameRxn code ("super"-, "common"-, or "specific"-class) is based on the hierarchy proposed by Carey et al.[43] The other filter criteria selectable are "name" and "none".

There are three possible scenarios, and all are visualized in Figure 2. The first option (Figure 2a) is that only the reaction is open for exchange, e.g., no structure node of the original tree can be exchanged. Therefore, only substitute reactions with the same number of reactants as the initial reaction are suitable. All substitute reactions that can be applied to the original reactant and product structures are kept. This means that all SMARTS patterns of the reaction describing reactants (in the following called ReactantSMARTS) have to match one original reactant structure and that the specified atom and bond changes must result in the original product structure. In the second scenario (Figure 2b), at least one but not all original reactant structures are open for exchange. This allows three options for the substitute reactions: First, reactions that have fewer ReactantSMARTS than the initial reaction are allowed, which results in the deletion of the exchangeable reactant structure nodes in the tree (and the whole subtree under this node, if the specified node is an internal node and not a leaf). Second, reactions with the same number of ReactantSMARTS as the initial reaction are allowed. Third, substitute reactions with more ReactantSMARTS than the number of original reactant structures are allowed. In this case, a new child node for the reaction node is created. In all three cases, only for the remaining nonexchangeable reactant structures, the substitute reaction has to have matching ReactantSMARTS patterns. In addition, the SMARTS patterns of the reaction describing products (in the following called ProductSMARTS) of the substitute reaction must match the initial reaction's product structure. The complete application of the substitute reaction cannot be checked until the exchange routine for the open reactant structures is started. The last option (Figure 2c) is that all reactant structures of the initial reaction are open for exchange. This allows substitute reactions with an arbitrary number of ReactantSMARTS. In this case, only

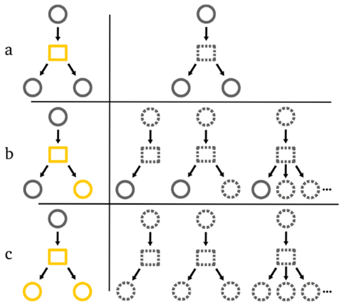


Figure 2. Abstract visualization of a reaction open for exchange. On the left is the initial reaction, where the parts open for exchange are marked in yellow. On the right are the options for substitute reactions. Exchanged parts are dotted. From top to bottom, the following cases are visualized: (a) no reactant is open for exchange, (b) some reactants are open for exchange, and (c) all reactants are open for exchange.

the compatibility of the ProductSMARTS of the substitute reaction is checked at this algorithm stage.

For all reactions passing this second filtering, a new tree is created. In each new tree, the initial reaction is replaced with the substitute reaction, and all additional node modifications (deletions or insertions) specified in the last step are included. If no other structure node in the tree is open for exchange, then the created tree is returned without further modification. Otherwise, the simultaneous multiple or single exchange function for the specified structure nodes is started as the last step. If substitute reactions with additional ReactantSMARTS are allowed, simultaneous multiple exchange needs to be selected.

Reaction Skipping. After taking reactions more into focus, we discovered that many otherwise suitable substitutes in our exchange functions fail because of de-/protection reactions in combination with the absence or existence of protection groups in the presented substitute. To tackle this problem, we included a reaction skipping mode (see Figure 1d). The algorithm automatically detects and skips reaction nodes if they hinder otherwise valid tree traversal. More specifically, the new function starts during verification of the tree validity if one reaction cannot be applied and no modified structures can be generated. In this case, the algorithm checks if the currently used reactant structures of the reaction can be used for the subsequent reaction traversing the tree upward. If possible, the reaction is skipped, and the verification process continues (see Figure 3).

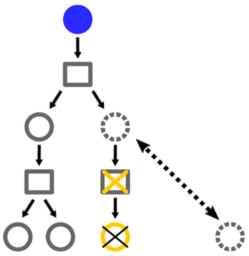


Figure 3. Abstract visualization of the reaction skipping function in a retrosynthetic tree. The skipped reaction and the exchange not made are marked with a cross. The new reactant exchange is symbolized by the arrow.

Currently, skipping reactions is allowed only once during the tree traversing and only for transformations, i.e., reactions with exactly one reactant and one product structure. Both limitations are installed to maintain the integrity and composition of the route.

Product Exchange. The modes described above, in total, enable the modification of all parts of a retrosynthetic route. For any modification the user can explicitly define the location and the wanted change in the computed physicochemical properties. However, this requires an already existing idea or knowledge about the route and the site of the modification. This will not be the case in all application scenarios; sometimes, only the location of the modification within the target structure is known. For this scenario, we designed the product exchange mode (see Figure 1e). This option simplifies the request for the user and enables the exchange process without prior knowledge of the route.

The algorithm starts with a specification of the (un)desired substructures of the target structure by the user. This is done with a target function for the target structure, which defines the substructures of the target molecule that should be kept (desired) or modified (undesired). The target function must be written as a SMARTS expression, which uniquely matches one substructure of the target molecule. If the user requests to ban one functional group occurring multiple times in the target molecule, then this can be done by defining an exclusion SMARTS pattern as a structural constraint. To identify the responsible nodes, each atom of the target structure is retraced to its originating node in the tree during initial parsing. The node can be not only a reactant but also a reaction. With the target function and the target structure's atom mapping, the algorithm automatically determines the responsible subtree. Finally, the suitable exchange routine starts with the identified nodes, and structural analogues are generated to fulfill the target function. Figure 4 shows an abstract visualization of the target structure's atom mapping and identification of the responsible nodes.

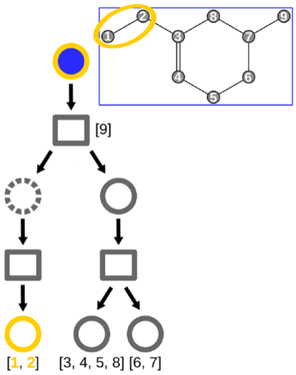


Figure 4. Abstract visualization of the product exchange mode in a retrosynthetic tree. The unwanted substructure is marked in yellow in the target structure in the upper right corner. All atom mappings are visualized under the responsible node by using the corresponding atom labels. Indirectly modified components due to exchange are dotted.

RESULTS AND DISCUSSION

In the following sections, examples to demonstrate the utility of Synthesia's exchange routines are shown. In addition, we provide a brief analysis of the different structural analogue sets generated by exchange algorithms for one target. Some general behavioral tendencies are discussed. As a final step, we demonstrate the method's applicability by combining it with a well-established drug design strategy: bioisosteric linker exchange. We analyze which bioisosteric linkers can be exchanged in a target structure without violating the integrity of a given route.

We created an initial retrosynthetic route for the targets in the following experiments with AiZynthFinder.[15] AiZynthFinder utilizes a Monte−Carlo tree search guided by a Keras neural network model as the rollout policy. Unique template codes were extracted from the United States Patent and Trademark Office (USPTO) and are used together with the policy to generate new precursors in the tree search. A list generated from the ZINC database in April 2020 was used for the available stock compounds.[15] In a real-life scenario, the user would provide synthetic pathways tested in practice that are feasible in one's own laboratories. For all of the following experiments, we employed the Enamine Building Blocks Global Stock collection[44] as potential substitute reactant structures. As of the access date, this collection encompassed 1,189,873 compounds. The required computing times vary significantly

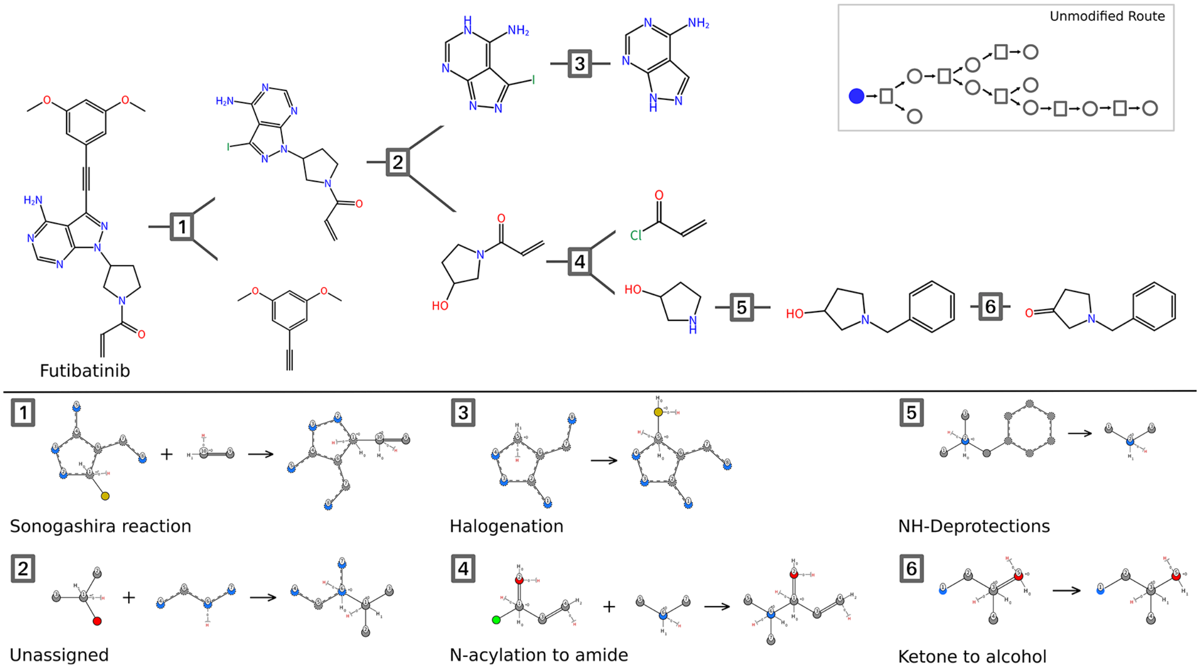


Figure 5. Unmodified retrosynthetic route of futibatinib. Each reaction is represented with a gray rectangle, the generic Reaction SMARTS pattern, and a classification provided by ref 15. The general composition of the route is visualized in the upper right corner.

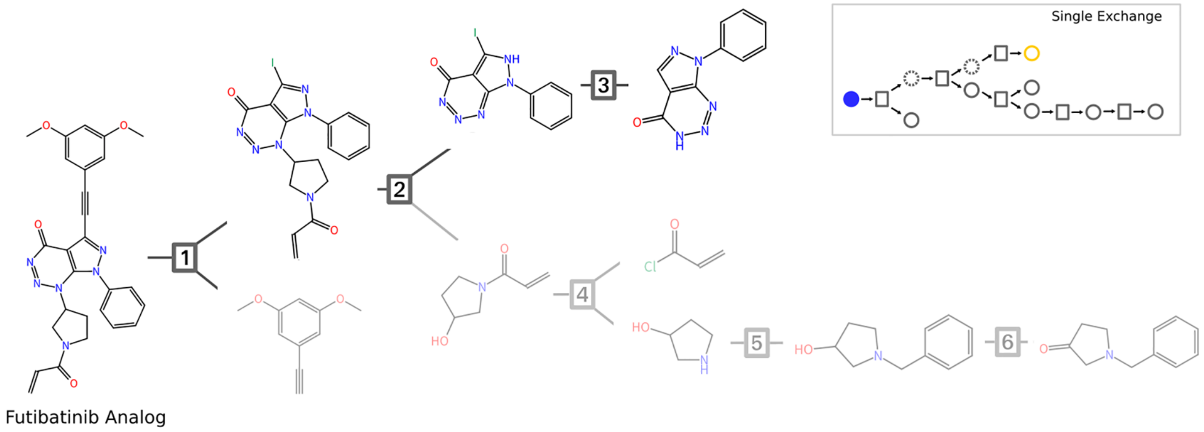


Figure 6. Modified retrosynthetic route of a futibatinib analogue. A single reactant structure is exchanged. The original node open for exchange is circled in yellow. All resulting structural changes can be viewed by traversing the route upward to the root. The general composition of the route and an abstract representation of the results of the selected mode are visualized in the top right corner.

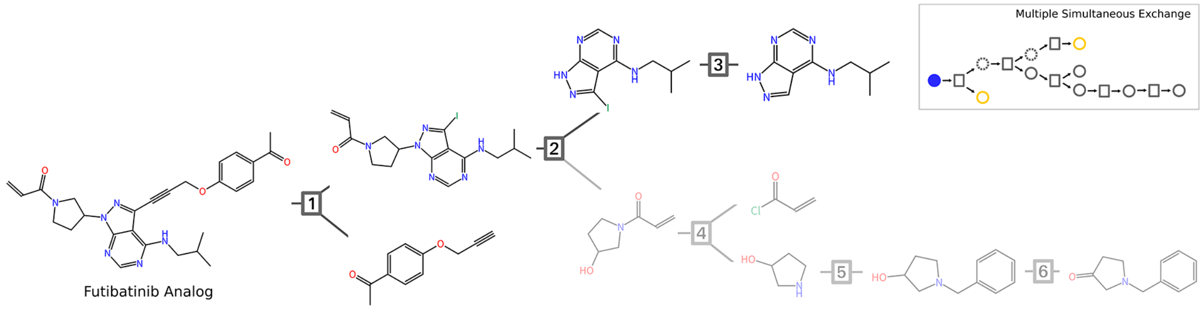


Figure 7. Modified retrosynthetic route of a futibatinib analogue. Two reactant structures are exchanged simultaneously. The original nodes open for exchange are circled in yellow. All resulting structural changes can be viewed by traversing the route upward to the root. The general composition of the route and an abstract representation of the results of the selected mode are visualized in the upper right corner.

based on the selected exchange mode, the frequency of pattern match, the size of the synthetic route, and other parameters. In

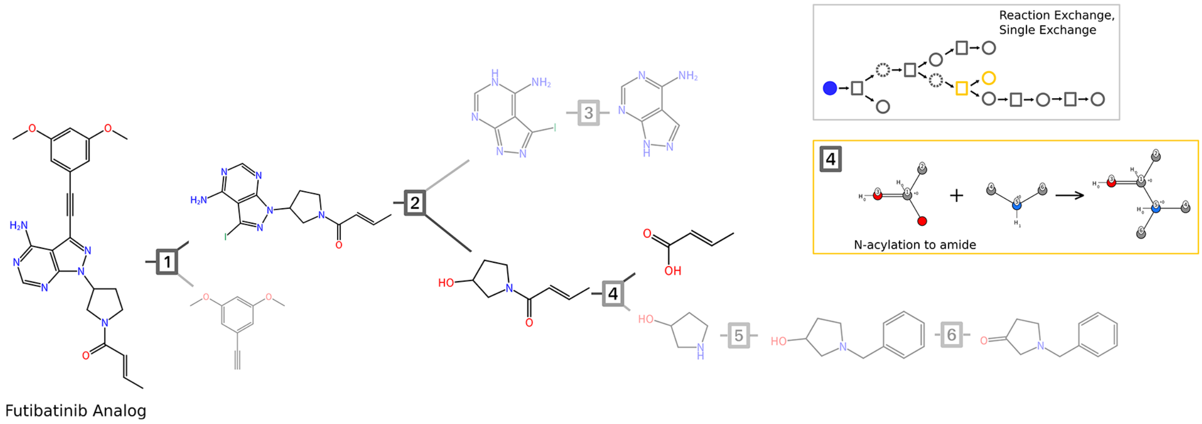


Figure 8. Modified retrosynthetic route of a futibatinib analogue. A reaction node is exchanged. The original node open for exchange is circled in yellow. All resulting structural changes can be viewed by traversing the route upward to the root. The general composition of the route and an abstract representation of the results of the selected mode are visualized in the upper right corner.

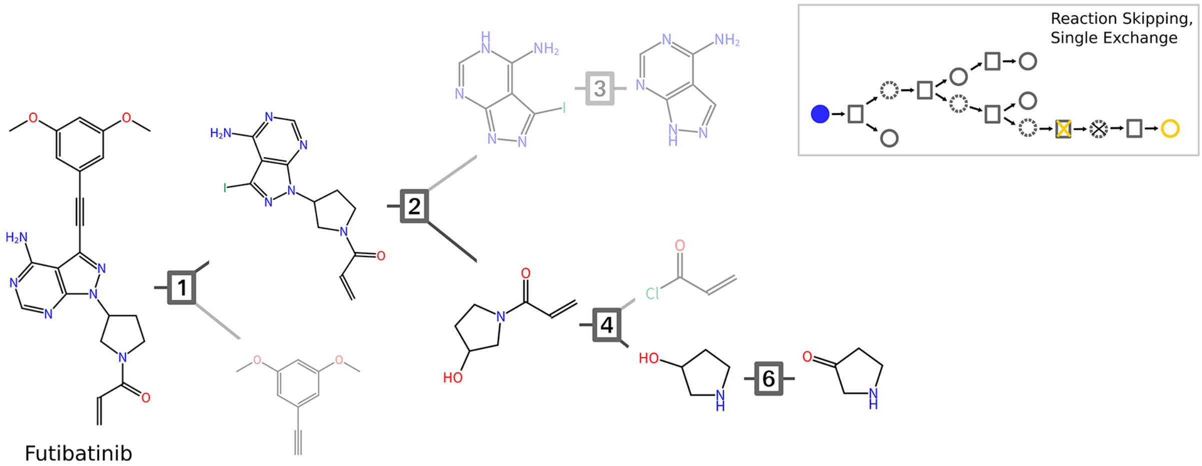


Figure 9. Modified retrosynthetic route of a futibatinib analogue. A reaction node is skipped. The original node open for exchange is circled in yellow. All resulting structural changes can be viewed by traversing the route upward to the root. The general composition of the route and an abstract representation of the results of the selected mode are visualized in the upper right corner.

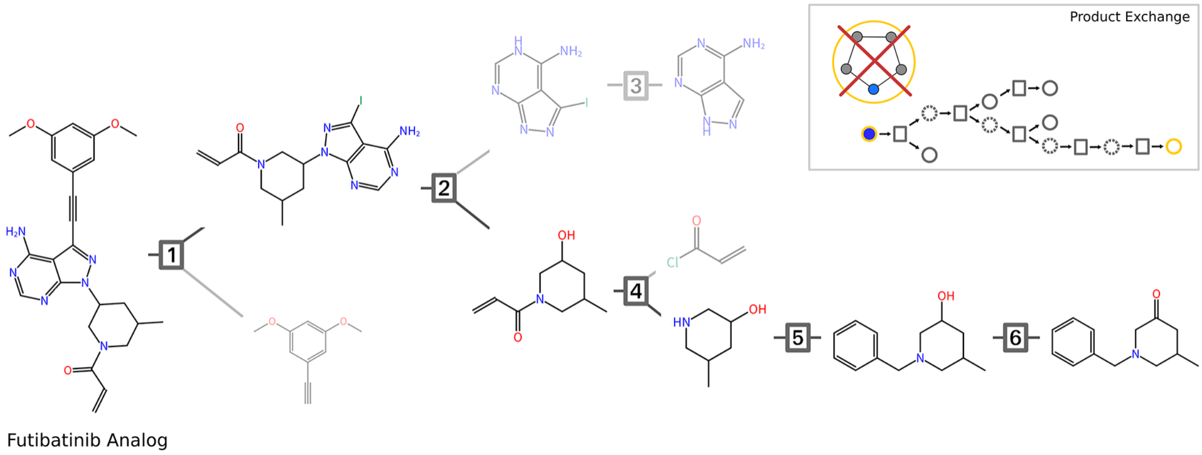


Figure 10. Modified retrosynthetic route of a futibatinib analogue. The product exchange function is used. A single reactant structure is exchanged. All resulting structural changes can be viewed by traversing the route upward to the root. The general composition of the route and an abstract representation of the results of the selected mode are visualized in the upper right corner.

our experiments, we attained a throughput of approximately 2000 building blocks per second and thread on a standard desktop machine (i5−8500 CPU, 16 GB RAM).

Mode Use Cases with the Example of Futibatinib. In the following, futibatinib,[45] a kinase inhibitor approved by the FDA in September 2022, will be used as our target structure.

Futibatinib irreversibly binds to the ATP-binding pocket of FGFR1−4, inhibiting the FGFR-mediated signal transduction pathway. It is used for the treatment of cholangiocarcinoma. We used AiZynthFinder[15] with the default settings and the provided trained model to create a retrosynthetic route for futibatinib. The unmodified route is shown in Figure 5. The route is comprised of six reactions and nine precursor structures containing four reactant structures included in the ZINC[46] database. AiZynthFinder scored the route as 0.96. Note that the route is used as an example to demonstrate our method and is not extracted from the original patent. The initial route and all modified routes are shown in Figures 6, 7, 8, 9, and 10.

We performed a single exchange of a reactant structure with three structural constraints in the presented route as a starting point. The structural constraints formulated are "logP less than 5", "molecular weight less than 500", and "number of aromatic rings greater than the original target structure". A structural analogue that satisfies these conditions is shown along with the modified route in Figure 6. All resulting structural changes are automatically calculated and can be viewed by traversing the tree upward to the root, starting with the modified node. No additional structural constraints were set to further guide the modification. The resulting modified target structure contains a new scaffold between the acetylene and pyrrolidine groups originating from the exchanged reactant structure in the middle of the molecule.

In the next step, we simultaneously exchanged two reactant structures using the simultaneous multiple exchange mode. The resulting route is visualized in Figure 7. Both exchanged reactant structures are responsible for large substructures in the target. Therefore, the exchange leads to a more significant structural difference in the target structure.

For demonstrating the reaction exchange algorithm, we picked the N-acylation-to-amide reaction. The selected reaction is quite specific and only allows structures containing an acyl chloride and a vinyl substructure as the first reactant. The reaction exchange function was applied with the name exchange filter setting. All generic reactions labeled as "N-acylation-to-amide" were checked for compatibility to find a suitable substitute. With these settings, the algorithm was able to identify a more generic reaction of the same reaction class, which allows a carboxylic acid as the first reactant. This allows for the use of a larger number of substitutes without straying far from the original retrosynthetic path. Figure 8 displays one resulting new route. In this case, the modifications lead to a futibatinib analogue registered in the patent. With the single or simultaneous multiple exchange mode and the unmodified route, this structural analogue would not have been accessible.

Figure 9 shows an example of the reaction skipping function during a single exchange of one reactant structure. The algorithm automatically detects that the selected substitute is incompatible with the deprotection reaction in the given retrosynthetic route. In other words, the algorithm detects a blocking reaction and tries to skip it by checking if the substitute is compatible with the subsequent reaction in the tree. In the case shown, selecting the appropriate substitute and skipping the deprotection reaction can shorten the retrosynthesis route, and the original target, futibatinib, can still be created. As a general application scenario, the reaction skipping can be used with in-house or in-stock compounds to analyze if a given route can be simplified. Of course, this function is also available when the target structure is actually modified.

We chose the pyrrolidine substructure open for exchange to demonstrate the effects of the product exchange algorithm. One resulting route is visualized in Figure 10. The algorithm automatically detects the responsible nodes and proposes suitable substitutes. The modified target structure lacks the specified substructure.

Structural Space Coverage between the Different Exchange Modes. The subsequent experiment aims at elucidating the space of structural analogues with respect to size and structural variation. For a target and a retrosynthetic route, all possible exchanges are performed. No additional structural constraints are set. The resulting structural analogue sets are analyzed by calculating the average similarity to the original structure and comparing in terms of overlap. We used the Extended Connectivity Fingerprint[47] (ECFP_4) and the Tanimoto coefficient for the similarity measurements. As target structure, we chose oteseconazole,[48] a cytochrome P450 (CYP) 51 inhibitor used to treat fungal infections. The AiZynthFinder[15] software provided a retrosynthetic route that scored 0.98. The route consists of 13 nodes with four reactions. The initial synthetic route is included in the Supporting Information as a JSON file. As input structures, we selected 35 compounds from the Enamine Building Blocks Global Stock[44] so each node had at least five compatible compounds. Node 12, a reactant structure and leaf node, is an exception because no compatible substitute could be found. Tables 1 and 2 show a selection of the results. We provide the full results and the input data in the Supporting Information.

Table 1. Results of All Single Exchange Options of Synthesia with the Target Structure Oteseconazole[a]

N-IDs	R-IDs	# analogues	averaged sim.	overlap
AllChemicals	-	31	0.521	31
AllLeaves	-	20	0.644	20
11 12	10	929	0.553	8
8 9	7	27	0.437	1
5 6	4	240	0.336	6
2 3	1	1923	0.327	13

[a] Starting with row three, the single exchange mode is combined with the reaction exchange mode. From left to right, the columns contain the exchanged node IDs of the chemical nodes (N-IDs), the exchanged node IDs of the reaction nodes (R-IDs), the number of generated structural analogues (# analogues), the resulting average similarity (averaged sim.), and the calculated overlap. The settings "AllChemicals" and "AllLeaves" automatically exchange all chemical components and all leave components in the tree, respectively.

While the presented experiment is a minimal example, some notable general trends can be realized. While Synthesia is highly dependent on the given input data, we hope to give the reader some insight and general ideas on how to use the algorithms.

The number of generated structural analogues varies highly between the exchange functions and the nodes open for exchange. As expected, in most cases, the more degrees of freedom, i.e., exchange possibilities, the more structural analogues are generated. However, some nodes result in significantly more structures than others after an exchange, although the same number of compatible substitutes are available. This may be due to the level at which the nodes are located in the tree or the restrictiveness of the generic reactions following the node in the synthesis route.

Table 2. Results of All Simultaneous Multiple Exchange Options of Synthesia with the Target Structure Oteseconazole[a]

N-IDs	R-IDs	# analogues	averaged sim.	overlap
AllLeaves	-	100	0.139	0
11 12	-	4	0.403	0
8 2	-	5	0.725	5
11 8 2	-	30	0.450	0
11 2 5	-	150	0.285	0
11 12 2 5	-	100	0.139	0
11 8 12 5	-	20	0.258	0
11 12	10	27	0.388	0
8 9	7	6	0.452	0
5 6	4	154	0.28	0
2 3	1	10087	0.082	0

[a] Starting with row eight, the simultaneous multiple exchange mode is combined with the reaction exchange mode. From left to right, the columns contain the exchanged node IDs of the chemical nodes (N-IDs), the exchanged node IDs of the reaction nodes (R-IDs), the number of generated structural analogues (# analogues), the resulting average similarity (averaged sim.), and the calculated overlap. The settings "AllChemicals" and "AllLeaves" automatically exchange all chemical components and all leave components in the tree, respectively.

The average similarity and number of structures generated vary significantly between the options presented. Logically, the less exchanges occur, the more similar the structures remain on average. Consequently, we get the highest similarities when only one or two leaf structures are exchanged simultaneously (compare Table 1 rows one and two and Table 2 rows two and three). The most structural modification measured in similarity occurs if multiple nodes are exchanged simultaneously (compare Table 2 rows six and seven). Note that all subtrees beneath the nodes are disregarded if internal structure nodes are exchanged.

The closer the reaction chosen for the exchange is to the root, i.e., the target compound, the greater the dissimilarity of the generated structural analogues and the further one deviates from the original route. This trend can be seen explicitly in the reaction exchange of node 10, with an overall similarity of 0.553, and the reaction exchange of node 1 (first after the target structure in the root), with a similarity of 0.28. But again, even with the reaction exchange, one can see that the structural analogues produced become more dissimilar when the reactants are exchanged simultaneously.

Looking at the number of common structures between the sets, which are almost none, it is clear that each exchange method reaches a different part of the possible structural analogue space. Each method introduces structural modification in a different way, which results in different modified target structures. It can be seen that all exchange routines have their raison d'être and meet the needs of different use cases.

Bioisosteric Linker Replacement Based on Route Compatibility. One valid question that can be leveled against the prior validation is how different the molecules really are. While the usage of fingerprints partially answers this question due to its one-dimensional nature, it does not properly highlight if these changes are solely minor changes to terminal groups of the resulting molecules or modifications of the molecular scaffold. During synthesis, changes to the scaffold of the molecule are regarded as more challenging than those to terminal groups, especially if these changes have to be made under conservation of the biological activity. To showcase Synthesia's ability to change the scaffold of the molecule while retaining its activity by using only bioisosteric replacements,[49] we utilized the recently published list of the most common linkers in bioactive molecules provided by Ertl et al.[50] We searched for potential linker substitutes that are synthetically accessible using a given retrosynthetic route. As the target structure, we use abrocitinib.[51] Abrocitinib was first approved in September 2021 in the UK for the treatment of moderate-to-severe atopic dermatitis in adults and adolescents. It is a small-molecule inhibitor of Janus kinase 1 (JAK1), a tyrosine kinase protein essential for signaling certain type I and II cytokines. We used AiZynthFinder[15] with default settings and the provided trained model to generate a retrosynthetic route for abrocitinib. The route scored 0.785 and can be found in the Supporting Information.

In the first step, we identified all linker substructures present in our target structure, abrocitinib. We selected only substructures that divide the target structure into precisely two parts, where each remaining part consists of at least three heavy atoms. With this, we found four linker substructures in abrocitinib. They are visualized in Figures 11 and 12 on the left half of the images.

We generated all possible structural analogues using Synthesia. We extracted those where one of the initially

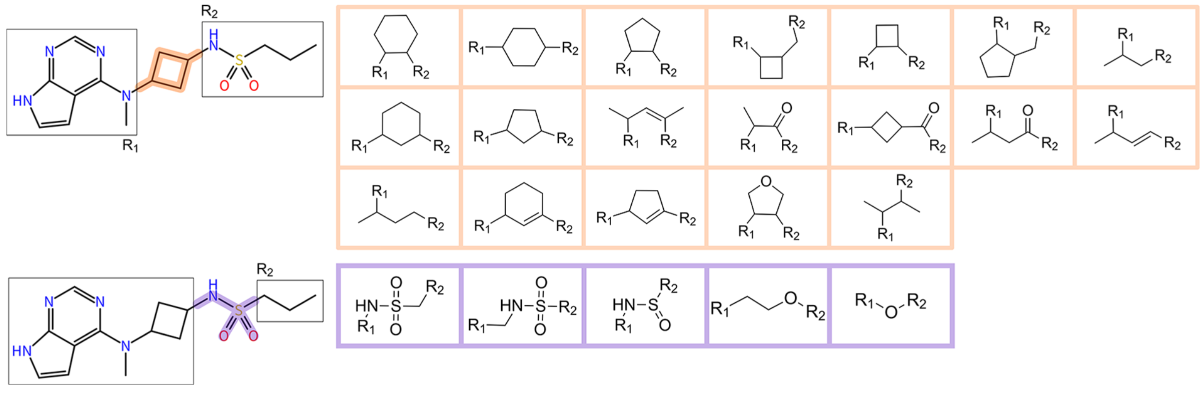


Figure 11. Visualization of the target structure abrocitinib containing two linker substructures (left top, cyclobutane; left bottom, sulfonamide) and the identified linker substitutes (right), which are synthetically accessible with the given retrosynthetic route. Each linker is marked in color, and the remaining molecule parts are surrounded by a rectangle labeled R_1 and R_2, corresponding to the attachment points of the linker substitutes.

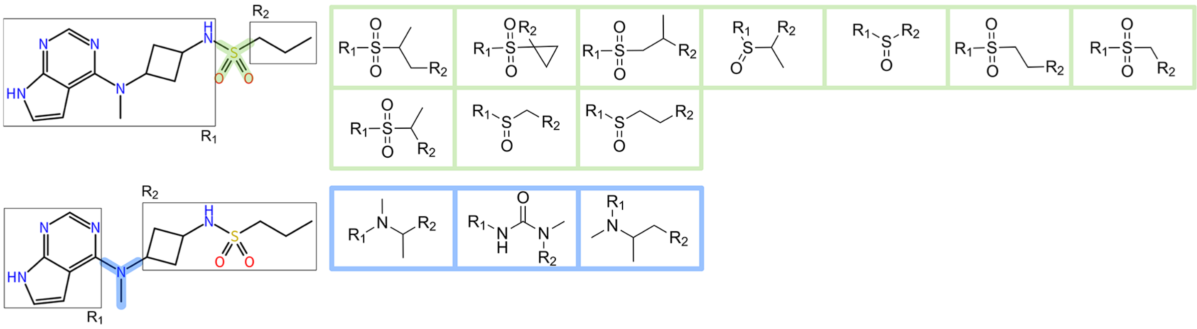


Figure 12. Visualization of the target structure abrocitinib containing two linker substructures (left top, sulfone; left bottom, methylamine) and the identified linker substitutes (right), synthetically accessible with the given retrosynthetic route. Each linker is marked in color, and the remaining molecule parts are surrounded by a rectangle labeled R_1 and R_2, corresponding to the attachment points of the linker substitutes.

identified linker substructures was replaced with one other linker from the set provided by Ertl et al.[50] The identified synthetically accessible linker substitutes are displayed on the right side in Figures 11 and 12. Each linker set marked with one color can be exchanged with the linker marked with the same color on the left. For the central linker, cyclobutane, we identified 19 synthetically compatible linker substructures. We found ten compatible linker structures for the sulfone linker and five for the sulfonamide linker. We could only find three possible substitutes in exchange for the methylamine linker. Each identified linker can be exchanged in the original target structure while theoretically still using the original retrosynthetic route for synthesis, with minor modifications. The corresponding target structures and their retrosynthetic routes can be found in the Supporting Information. Analyses such as these are instrumental because the exchange of linker substructures, as opposed to terminal groups, typically presents more severe challenges for synthesis. Identifying synthetically accessible linker structures from a predefined set based on a given retrosynthetic route for a target structure can significantly improve and simplify the process of bioisosteric replacements.

■ CONCLUSION

The development of lead compounds is dominated by the design−make−test−analyze cycle. In this process, synthetic accessibility is a bottleneck. A common reason for this is that synthesizability is mostly considered separately from the design process. This can result in compounds perfectly suited for their designed task but requiring significant effort in synthesis, especially if the already established routes cannot be kept. Therefore, many structures designed *in silico* must be discarded because of their synthetic accessibility. With the progress in machine learning, modern computer-aided synthesis planning methods are shifting more and more into focus. The current generation of these methods provides impressive results in searching for synthesis routes for novel compounds. However, the possibilities for medicinal or synthetic chemists to contribute expertise are usually limited to the final selection of the routes. In particular, the consideration of the SAR landscape under investigation is usually not supported.

Here, we introduced Synthesia, enabling a change in perspective and giving medicinal chemists access to all modification possibilities in a retrosynthetic route to continue the design process on the basis of a synthetically accessible target structure. Individually defined physicochemical constraints steer the structural properties in the desired direction. Regions of modification can be specified, ensuring that the identified key interactions remain untouched.

Using futibatinib as an exemplary target structure, we have presented and discussed the effects of the various modification options in a retrosynthetic route. For each exchange routine, we present the modified retrosynthetic route and the resulting structural variants. We demonstrate the exchange of a single reactant with structural constraints, the simultaneous exchange of multiple reactants, the exchange and skipping of reactions, and the definition of a target function that defines undesirable substructures in the target compound. All exchange modes can be combined with structural constraints that define the physicochemical properties of the target structure. An example is given, together with the exchange of a single reactant. By a respective combinatorial search, Synthesia is able to determine the responsible subtree and propose modification options automatically. These examples demonstrate that the additional exchange functions meet the desired design goal. The method can incorporate modifications in the desired direction and location without compromising the applicability of the retrosynthetic route.

In addition, we conducted a simple experiment to discuss the difference among the various exchange features. Although we used only one target structure and a small set of possible substitute candidates, some trends can be identified that can be generalized. The number of generated structural analogues increases with the possible degrees of freedom, i.e., the components of the route that are modified. However, the structural similarity to the original target structure also decreases to the same extent. As for many computational approaches, the results of Synthesia are highly dependent on input data and parameters, and individual experiments may produce a variety of different results.

In a last step, we presented Synthesia in the frequent design scenario of bioisosteric linker replacement. Synthesia can be used to analyze which linker structures from a given set of linkers can be used for substitution in a target structure without compromising the applicability of a given retrosynthetic route. We use abrocitinib as an exemplary target structure and the list of most common linkers in bioactive molecules provided by Ertl et al.[50] Synthesia identified linker substitutes that could theoretically be synthesized using the same retrosynthetic pathway with slight modifications for all four linkers in the original target structure. Depending on the location of the original linker and the responsible reactant structures, different exchange features had to be used to access all of the substitute linkers presented. Analyses such as these can highly simplify the selection of linker structures for bioisosteric replacements, which are otherwise usually more complicated in their synthetic accessibility. If presented with field-proven or proprietary synthesis protocols, linker structures are identified for substitution that can theoretically be readily synthesized in the respective route.

With this work, we have introduced a computer-based method that places synthetic pathways at the center of attention. Focused on directly supporting synthetic and medicinal chemists in the targeted modification of lead structures, the method allows for a high level of integration of their expertise but could also be incorporated into automated computational workflows.

■ ASSOCIATED CONTENT

Data Availability Statement

Synthesia is available for Linux and Windows as part of the NAOMI ChemBio Suite at https://uhh.de/naomi and is free for academic use and evaluation purposes. The original and modified JSON files of the retrosynthetic routes as well as the used target structures are available in the Supporting Information material.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c01155.

Initial retrosynthetic routes of the target structures, structural space coverage between the different exchange modes, and bioisosteric linker replacement based on route compatibility (PDF)

JSON files for the initial retrosynthetic routes and abrocitinib analogue routes (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Matthias Rarey − *Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany;* orcid.org/0000-0002-9553-6531; Phone: +49 (40) 428387351; Email: matthias.rarey@uni-hamburg.de

Authors

Uschi Dolfus − *Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany;* orcid.org/0000-0002-2720-1086

Hans Briem − *Bayer AG, Research & Development, Pharmaceuticals, Computational Molecular Design Berlin, 13342 Berlin, Germany;* orcid.org/0000-0002-8498-2448

Torben Gutermuth − *Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany;* orcid.org/0000-0002-9304-8251

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.3c01155

Author Contributions

U.D., H.B., and M.R. developed the Synthesia concept. U.D. implemented the Synthesia approach. T.G. did the concept design for the experiments. M.R. supervised the project. All authors participated in manuscript writing.

Notes

The authors declare the following competing financial interest(s): M.R., as a shareholder of BioSolveIT GmbH, declares a potential financial interest in the event that the Synthesia software is licensed for a fee to nonacademic institutions in the future. This work is related to the PhD research topic of U.D., financially supported by Bayer.

■ REFERENCES

(1) Corey, E. J.; Wipke, W. T. Computer-Assisted Design of Complex Organic Syntheses: Pathways for Molecular Synthesis Can Be Devised with a Computer and Equipment for Graphical Communication. *Science* **1969**, *166*, 178−192.

(2) Thakkar, A.; Johansson, S.; Jorner, K.; Buttar, D.; Reymond, J.-L.; Engkvist, O. Artificial Intelligence and Automation in Computer Aided Synthesis Planning. *React. Chem. Eng.* **2021**, *6*, 27−51.

(3) Tu, Z.; Stuyver, T.; Coley, C. W. Predictive Chemistry: Machine Learning for Reaction Deployment, Reaction Development, and Reaction Discovery. *Chem. Sci.* **2023**, *14*, 226.

(4) Baskin, I. I.; Madzhidov, T. I.; Antipin, I. S.; Varnek, A. A. Artificial Intelligence in Synthetic Chemistry: Achievements and Prospects. *Russ. Chem. Rev.* **2017**, *86*, 1127.

(5) Park, S.; Han, H.; Kim, H.; Choi, S. Machine Learning Applications for Chemical Reactions. *Chem. - Asian J.* **2022**, *17*, No. e202200203.

(6) Szymkuć, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: the End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.

(7) Pensak, D.; Corey, E. LHASA−Logic and Heuristics Applied to Synthetic Analysis. *Computer Assisted Organic Synthesis* **1977**, *61*, 1.

(8) Corey, E. J.; Long, A. K.; Rubenstein, S. D. Computer-Assisted Analysis in Organic Synthesis. *Science* **1985**, *228*, 408−418.

(9) Schwaller, P.; Petraglia, R.; Zullo, V.; Nair, V. H.; Haeuselmann, R. A.; Pisoni, R.; Bekas, C.; Iuliano, A.; Laino, T. Predicting Retrosynthetic Pathways Using Transformer-Based Models and a Hyper-Graph Exploration Strategy. *Chem. Sci.* **2020**, *11*, 3316−3325.

(10) Tetko, I. V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-Art Augmented NLP Transformer Models for Direct and Single-Step Retrosynthesis. *Nat. Commun.* **2020**, *11*, 5575.

(11) Dong, J.; Zhao, M.; Liu, Y.; Su, Y.; Zeng, X. Deep Learning in Retrosynthesis Planning: Datasets, Models and Tools. *Briefings Bioinf* **2022**, *23*, bbab391.

(12) *Askcos: Software Tools for Organic Synthesis*. https://askcos.mit.edu/ (accessed 2023-09-05).

(13) *Molecule.One*. https://molecule.one/ (accessed 2023-09-05).

(14) *Chemical.AI*. https://chemical.ai/ (accessed 2023-09-05).

(15) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: A Fast, Robust, and Flexible Open-Source Software for Retrosynthetic Planning. *J. Cheminf.* **2020**, *12*, 70.

(16) Christ, C. D.; Zentgraf, M.; Kriegl, J. M. Mining Electronic Laboratory Notebooks: Analysis, Retrosynthesis, and Reaction Based Enumeration. *J. Chem. Inf. Model.* **2012**, *52*, 1745−1756.

(17) Ghiandoni, G. M.; Bodkin, M. J.; Chen, B.; Hristozov, D.; Wallace, J. E.; Webster, J.; Gillet, V. J. RENATE: A Pseudo-Retrosynthetic Tool for Synthetically Accessible De Novo Design. *Mol. Inf.* **2022**, *41*, No. 2100207.

(18) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. Recap Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Model.* **1998**, *38*, 511−522.

(19) Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F.; Heeres, J.; Koymans, L. M.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. Synopsis: Synthesize and Optimize System In Silico. *J. Med. Chem.* **2003**, *46*, 2765−2773.

(20) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem.* **2008**, *3*, 1503−1507.

(21) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. Dogs: Reaction-Driven De Novo Design of Bioactive Compounds. *PLoS Comput. Biol.* **2012**, *8*, No. e1002380.

(22) Bjerrum, E. J.; Threlfall, R. Molecular Generation with Recurrent Neural Networks (RNNs). *arXiv* **2017**, n/a.

(23) Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; et al. Deep Learning Enables Rapid Identification of Potent DDR1 Kinase Inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038−1040.

(24) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.* **2020**, *11*, No. 565644.

(25) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. GuacaMol: Benchmarking Models for De Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59*, 1096−1108.

(26) Gao, W.; Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.* **2020**, *60*, 5714−5723.

(27) Bilodeau, C.; Jin, W.; Jaakkola, T.; Barzilay, R.; Jensen, K. F. Generative Models for Molecular Discovery: Recent Advances and Challenges. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2022**, *12*, No. e1608.

(28) Ertl, P.; Schuffenhauer, A. Estimation of Synthetic Accessibility Score of Drug-Like Molecules Based on Molecular Complexity and Fragment Contributions. *J. Cheminf.* **2009**, *1*, 1−11.

(29) Fukunishi, Y.; Kurosawa, T.; Mikami, Y.; Nakamura, H. Prediction of Synthetic Accessibility Based on Commercially Available Compound Databases. *J. Chem. Inf. Model.* **2014**, *54*, 3259−3267.

(30) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. SCScore: Synthetic Complexity Learned from a Reaction Corpus. *J. Chem. Inf. Model.* **2018**, *58*, 252−261.

(31) Thakkar, A.; Chadimová, V.; Bjerrum, E. J.; Engkvist, O.; Reymond, J.-L. Retrosynthetic Accessibility Score (RAscore): Rapid Machine-Learned Synthesizability Classification from AI-Driven Retrosynthetic Planning. *Chem. Sci.* **2021**, *12*, 3339−3349.

(32) Zhang, Q.; Liu, C.; Wu, S.; Hayashi, Y.; Yoshida, R. A Bayesian Method for Concurrently Designing Molecules and Synthetic Reaction Networks. *Sci. Technol. Adv. Mater.: Methods* **2023**, *3*, No. 2204994.

(33) Horwood, J.; Noutahi, E. Molecular Design in Synthetically Accessible Chemical Space Via Deep Reinforcement Learning. *ACS omega* **2020**, *5*, 32984−32994.

(34) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M.; Hernández-Lobato, J. M. Barking Up the Right Tree: An Approach to Search Over Molecule Synthesis DAGs. *Adv. Neural Inf. Process.* **2020**, *33*, 6852−6866.

(35) Gottipati, S. K.; Sattarov, B.; Niu, S.; Pathak, Y.; Wei, H.; Liu, S.; Blackburn, S.; Thomas, K.; Coley, C.; Tang, J., et al. Learning to Navigate the Synthetically Accessible Chemical Space using Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, July 2020; pp 3668−3679.

(36) Dolfus, U.; Briem, H.; Rarey, M. Synthesis-Aware Generation of Structural Analogues. *J. Chem. Inf. Model.* **2022**, *62*, 3565−3576.

(37) *Daylight, SMARTS - A Language for Describing Molecular Patterns.* https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (accessed 2023-03-25).

(38) *Daylight, SMIRKS - A Reaction Transform Language.* https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html (accessed 2023-03-25).

(39) *Daylight, SMILES - A Simplified Chemical Language.* https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html (accessed 2023-03-25).

(40) Sommer, K.; Flachsenberg, F.; Rarey, M. NAOMInext−Synthetically Feasible Fragment Growing in a Structure-Based Design Context. *Eur. J. Med. Chem.* **2019**, *163*, 747−762.

(41) Hartenfeller, M.; Eberle, M.; Meier, P.; Nieto-Oberhuber, C.; Altmann, K.-H.; Schneider, G.; Jacoby, E.; Renner, S. A Collection of Robust Organic Synthesis Reactions for in Silico Molecule Design. *J. Chem. Inf. Model.* **2011**, *51*, 3093−3098.

(42) *NextMove, NameRXN - Expert System for Named Reaction Identification and Classification.* https://www.nextmovesoftware.com/namerxn.html, last accessed on 25/03/2023.

(43) Carey, J. S.; Laffan, D.; Thomson, C.; Williams, M. T. Analysis of the Reactions Used for the Preparation of Drug Candidate Molecules. *Org. Biomol. Chem.* **2006**, *4*, 2337−2347.

(44) *Enamine, Enamine Building Blocks Global Stock.* https://enamine.net/building-blocks/building-blocks-catalog (accessed 2023-02-07).

(45) Syed, Y. Y. Futibatinib: First Approval. *Drugs* **2022**, *82*, 1737−1743.

(46) Sterling, T.; Irwin, J. J. ZINC 15−Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.

(47) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(48) Hoy, S. M. Oteseconazole: First Approval. *Drugs* **2022**, *82*, 1017−1023.

(49) Lima, L. M.; Barreiro, E. J. Bioisosterism: A Useful Strategy for Molecular Modification and Drug Design. *Curr. Med. Chem.* **2005**, *12*, 23−49.

(50) Ertl, P.; Altmann, E.; Racine, S. The Most Common Linkers in Bioactive Molecules and Their Bioisosteric Replacement Network. *Bioorg. Med. Chem.* **2023**, *81*, No. 117194.

(51) Deeks, E. D.; Duggan, S. Abrocitinib: First Approval. *Drugs* **2021**, *81*, 2149−2157.

Supporting Information: Full Modification Control over Retrosynthetic Routes for Guided Optimization of Lead Structures

Uschi Dolfus,[†] Hans Briem,[‡] Torben Gutermuth,[†] and Matthias Rarey*,[†]

[†] *Universität Hamburg, ZBH - Center for Bioinformatics, Bundesstraße 43, 20146 Hamburg, Germany*

[‡] *Bayer AG, Research & Development, Pharmaceuticals, Computational Molecular Design Berlin, Building S110, 711, 13342 Berlin, Germany*

E-mail: matthias.rarey@uni-hamburg.de

Phone: +49 (40) 428387351

1 Initial Retrosynthetic Routes of Target Structures

Synthesia expects the retrosynthetic trees as JSON files. We provide the used initial JSON files of the target structures Futibatinib, Oteseconazole and Abrocitinib in the Supporting Information material. All initial routes were generated with AiZynthFinder[S1].

2 Structural Space Coverage between the Different Exchange Modes

In the following, we provide the list of compatible substitutes extracted from the Enamine Building Blocks Global Stock[S2] (compare table S3). Table S1 and S2 show the complete results of the space coverage experiment.

Table S1: Results of all single exchanging options of Synthesia with the target structure Oteseconazole. Starting with row three the single exchange mode is combined with the reaction exchange. From left to right, the columns contain the exchanged node ids of chemical nodes (N-Ids), the exchanged node ids of reaction nodes (R-Ids), the number of generated structural analogs (# Analogs), the resulting average similarity (Avg. Sim.), the calculated Overlap. The settings 'AllChemicals' and 'AllLeaves' exchanges automatically all chemical components respectively all leave components in the tree.

N-Ids	R-Id	# Analogs	Avg Sim.	Overlap
AllChemicals	-	31	0.521	-
AllLeaves	-	20	0.644	20
11 12	10	929	0.553	8
8 9	7	27	0.437	1
5 6	4	240	0.336	6
1	2 3	240	0.336	13

3 Bioisosteric Linker Replacement based on Route Compatibility

As already mentioned the initial retrosynthetic route of the target structure Abrocitinib[S3] can be found in the Supporting Information material. The list of most common linkers in bioactive molecules needs to be extracted from the corresponding publication by Ertl et al.[S4] Table S4 displays all structural analogs with a replaced linker structure. The corresponding modified retrosynthetic routes can be found in the Supporting Information material. Note that the first linker substitute of the original Sulfonamide linker and the seventh linker substitute of the original Sulfone-linker result in the same compound. Therefore, only 36

Table S2: Results of all simultaneous multiple exchanging options of Synthesia with the target structure Oteseconazole. Starting with row three the simultaneous multiple exchange mode is combined with the reaction exchange. From left to right, the columns contain the exchanged node ids of chemical nodes (N-Ids), the exchanged node ids of reaction nodes (R-Ids), the number of generated structural analogs (# Analogs), the resulting average similarity (Avg. Sim.), the calculated Overlap. The settings 'AllChemicals' and 'AllLeaves' exchanges automatically all chemical components respectively all leave components in the tree.

N-Ids	R-Id	# Analogs	Avg Sim.	Overlap
AllLeaves	-	100	0.139	0
AllChemicals	-	100	0.139	0
11 8	-	6	0.622	6
11 12	-	4	0.403	0
11 2	-	30	0.45	0
11 5	-	30	0.406	0
8 12	-	0	-	-
8 2	-	5	0.725	5
8 5	-	5	0.640	5
12 2	-	0	-	-
12 5	-	0	-	-
2 5	-	25	0.467	0
11 8 12	-	4	0.403	0
11 8 2	-	30	0.450	0
8 2 5	-	25	0.467	0
11 8 5	-	30	0.406	0
11 12 2	-	20	0.247	0
11 12 5	-	20	0.258	0
11 2 5	-	150	0.285	0
8 12 2	-	0	-	-
8 12 5	-	0	-	-
12 2 5	-	0	-	-
11 8 12 2	-	20	0.247	0
11 8 2 5	-	150	0.285	0
8 12 2 5	-	0	-	-
11 12 2 5	-	100	0.139	0
11 8 12 5	-	20	0.258	0
11 12	10	27	0.388	0
8 9	7	6	0.452	0
5 6	4	154	0.28	0
1	2 3	10087	0.082	0

compounds are listed.

Table S3: List of used compatible substitute compounds in the space coverage experiment. The compounds were extracted from the Enamine Building Blocks Global Stock[S2] set.

SMILES
O(c1cc(ccc1)B(O)O)C
Clc1cc(Cl)cc(c1)B(O)O
Brc1ccc(cc1)B(O)O
OB(O)c1cc(ccc1)C(C)C
Fc1c(F)cc(cc1F)B(O)O
Clc1cc(OC)c(cc1)C(=O)O
Clc1cc2NC(S)=Nc2cc1
Clc1ccc(S(=O)(=O)NC(C(=O)O)C(O)C)cc1
Clc1c(S(=O)(=O)N(c2ccc(F)cc2)Cc3ccccc3)cc(cc1)C(=O)O
Brc1cc(Cl)c(N)cc1
O=C1N(N=NN1)c2ccccc2
O(c1cc(ccc1)C2=NN=NN2)C
BrC=1OC(C2=NN=NN2)=CC1
O=C(O)CC1=NN=NN1
N1=NNC(=N1)C(N)c2ccccc2
Fc1ccc(cc1)C2OC2
Fc1cc(ccc1)C2OC2
Brc1cc(ccc1)C2OC2
O1C(c2ccc(C#N)cc2)C1
Clc1c(cccc1)C2OC2
ClCC(=O)C1=C(N(C(=C1)C)CCc2ccccc2)C
O=C(C1=C(OC(=C1)C)C)C
ClCC(=O)c1cc2c3N(C(=O)C2)CCCc3c1
ClCC(=O)C1=C(N(c2ccc(S(=O)(=O)N)cc2)C(=C1)C)C
Clc1cc(ccc1)C(=O)C=C2SCC(=O)N2
Brc1c2OCOc2cc(c1)C=O
Brc1cc(Cl)c(N)cc1
Brc1cc(c(N)cc1)C(=O)OC
BrC1=CSC=C1
Brc1ccc(cc1)C2=NNC(C(=O)O)=C2
O=C1N(C(=O)CN1)CC(=O)OCC
O=C1OC(=O)CC1(CC)CC
ClCC(=O)OCCCCCCCCCC
S1C(=NC(C(=O)C(=O)OCC)=C1)N
S(c1c(cccc1)C(=O)O)CC(=O)OC2C(C(C)C)CCC(C2)C

Table S4: List of structural analogs of the target structure Abrocitinib with exchanged linker structure generated by Synthesia.

SMILES
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)C(CCC)C
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)C(CCCC)C
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)C4(CC4)CCC
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CC(CCC)C
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCCC
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCCCC
C1(CC(CNS(=O)(=O)CCC)C1)N(C)c2ncnc3NC=Cc23
S(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)C(CCC)C
S(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCC
S(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCCC
S(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCCCC
C(C=C(NS(=O)(=O)CCC)C)(C)N(C)c1ncnc2NC=Cc12
O=C(NS(=O)(=O)CCC)C(C)N(C)c1ncnc2NC=Cc12
O=C(NS(=O)(=O)CCC)C1CC(C1)N(C)c2ncnc3NC=Cc23
O=C(NS(=O)(=O)CCC)CC(C)N(C)c1ncnc2NC=Cc12
C(C(NS(=O)(=O)CCC)C)(C)N(C)c1ncnc2NC=Cc12
C1(C=C(NS(=O)(=O)CCC)CC1)N(C)c2ncnc3NC=Cc23
C1(C=C(NS(=O)(=O)CCC)CCC1)N(C)c2ncnc3NC=Cc23
C1(C(NS(=O)(=O)CCC)CC1)N(C)c2ncnc3NC=Cc23
C1(C(NS(=O)(=O)CCC)CCC1)N(C)c2ncnc3NC=Cc23
C1(C(NS(=O)(=O)CCC)CCCC1)N(C)c2ncnc3NC=Cc23
C1(C(NS(=O)(=O)CCC)COC1)N(C)c2ncnc3NC=Cc23
S(=O)(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)CC1)CCC
C1(CC(NS(=O)(=O)CCC)CCC1)N(C)c2ncnc3NC=Cc23
C1(CCC(NS(=O)(=O)CCC)CC1)N(C)c2ncnc3NC=Cc23
C(C=CNS(=O)(=O)CCC)(C)N(C)c1ncnc2NC=Cc12
C(CNS(=O)(=O)CCC)(C)N(C)c1ncnc2NC=Cc12
C1(C(CNS(=O)(=O)CCC)CC1)N(C)c2ncnc3NC=Cc23
C1(C(CNS(=O)(=O)CCC)CCC1)N(C)c2ncnc3NC=Cc23
C(CCNS(=O)(=O)CCC)(C)N(C)c1ncnc2NC=Cc12
S(=O)(NC1CC(N(c2ncnc3NC=Cc23)C)C1)CCC
O(CCC1CC(N(c2ncnc3NC=Cc23)C)C1)CCC
O(C1CC(N(c2ncnc3NC=Cc23)C)C1)CCC
N(C(=O)N(C)c1ncnc2NC=Cc12)C3CC(NS(=O)(=O)CCC)C3
C(CC1CC(NS(=O)(=O)CCC)C1)(C)N(C)c2ncnc3NC=Cc23
C(C1CC(NS(=O)(=O)CCC)C1)(C)N(C)c2ncnc3NC=Cc23

References

(S1) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: A Fast, Robust, and Flexible Open-Source Software for Retrosynthetic Planning. *J. Cheminf.* **2020**, *12*, 70.

(S2) Enamine, Enamine Building Blocks Global Stock. `https://enamine.net/building-blocks/building-blocks-catalog`, last accessed on 07/02/2023.

(S3) Deeks, E. D.; Duggan, S. Abrocitinib: First Approval. *Drugs* **2021**, *81*, 2149–2157.

(S4) Ertl, P.; Altmann, E.; Racine, S. The Most Common Linkers in Bioactive Molecules and Their Bioisosteric Replacement Network. *Bioorg. Med. Chem.* **2023**, *81*, 117194.

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate

Hamburg, den August 12, 2024

Uschi Dolfus