Journal of
CHEMICAL INFORMATION
AND MODELING

Article

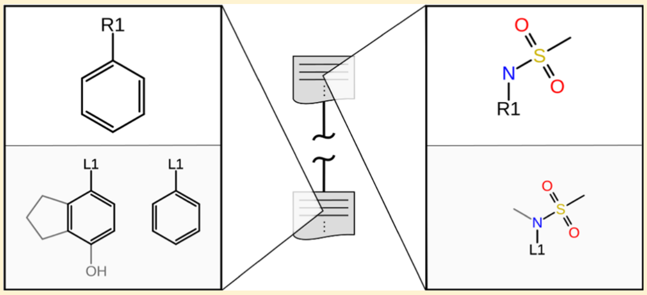
pubs.acs.org/jcim

Searching for Substructures in Fragment Spaces

Hans-Christian Ehrlich, Andrea Volkamer, and Matthias Rarey*

University of Hamburg, Bundestraße 43, 20146 Hamburg, Germany

ABSTRACT: A common task in drug development is the selection of compounds fulfilling specific structural features from a large data pool. While several methods that iteratively search through such data sets exist, their application is limited compared to the infinite character of molecular space. The introduction of the concept of fragment spaces (FSs), which are composed of molecular fragments and their connection rules, made the representation of large combinatorial data sets feasible. At the same time, search algorithms face the problem of structural features spanning over multiple fragments. Due to the combinatorial nature of FSs, an enumeration of all products is impossible. In order to overcome these time and storage issues, we present a method that is able to find substructures in FSs without explicit product enumeration. This is accomplished by splitting substructures into subsubstructures and mapping them onto fragments with respect to fragment connectivity rules. The method has been evaluated on three different drug discovery scenarios considering the exploration of a molecule class, the elaboration of decoration patterns for a molecular core, and the exhaustive query for peptides in FSs. FSs can be searched in seconds, and found products contain novel compounds not present in the PubChem database which may serve as hints for new lead structures.



INTRODUCTION

Finding molecules that fulfill specific structural or physicochemical features is of high practical interest in drug development. Due to the large and still growing number of commercially available and synthetically accessible molecule structures, efficient algorithms for searching large data sets are becoming more and more vital.[1,2] Traditionally, huge compound sets are maintained in large databases. Different computational methods have been developed to efficiently search through these data sets.[3−14] One major application is the retrieval of molecules that include a defined molecular substructure.

To efficiently process large databases, molecules are described as graphs, where nodes denote the atoms and edges the connecting bonds. With such a representation, search algorithms can take advantage of known graph theoretical concepts allowing for an efficient graph comparison. The applied methods range from matrix-based[7] and backtracking algorithms[11,12] for (sub)graph isomorphism, over branch-and-bound,[15,16] maximal clique,[17] and dynamic programming algorithms[18] for maximal common subgraph calculations, to path and radial fragment enumeration[19] for graph similarity search. Nevertheless, since the number of molecules in the chemical universe is almost infinite, databases can reach a critical size where iterative search strategies reach their limits.

Alternative storage principles have been introduced, e.g., Markush structures used in chemical patents. A Markush structure is usually given by a core fragment with open valences and a list of corresponding decoration fragments. A complete molecule is constructed by attaching these fragments to the core until all open attachment points are saturated. A more general concept of such a combinatorial space is a *fragment space* (FS). An FS follows the approach of the retrosynthetical combination analysis procedure (RECAP).[20] RECAP describes distinct rules that model chemical motifs which can easily be formed by combinatorial chemists. An FS is created by applying these rules to separate molecules into fragments. Therefore, an FS consist of molecular fragments with open valences and a set of rules defining their possible combinations to products. For example, the BRICS 4k[21] space comprises 4800 fragments and 64 connection rules. Alternatively, FS can be designed from combinatorial chemistry[22,23] describing reaction schemes in which building blocks are connected prohibiting the formation of cyclic products. Even though the number of fragments in an FS is small, their combinatorial properties allow for the construction of many different products, e.g, enumerating all possible products with up to five fragments in BRICS 4k yields 10^{16} molecules. That is a number which is difficult to handle with a conventional database.

A small number of algorithms to process FSs exist. They solve classical problems in cheminformatics such as the search for similar molecules,[24,25] the novel design of molecules,[26−29] and the creation of FSs focused around target molecules.[30−32]

While methods to search for substructures in molecules exist,[7,11,12] methods browsing through FSs under substructure constraints are rare. Three database systems have been published to search patents for query structures, GENSAL,[33−35] Markush Darc,[36] and MARPAT.[37,38] All systems hold Markush structures retrieved from patent information and store them as reduced graphs. Markush Darc restricts a query to an explicit substructure, whereas MARPAT allows for the occurrence of

Received: June 19, 2012

variable and generic groups. Both search methods employ a two step strategy, a screening phase based on limited-environment fragments which is followed by an iterative atom-by-atom search on the remaining structures. The major drawback of both search strategies is their limitation to only handle Markush structures. In more detail, both methods are designed to search the description of core fragments with varying decorations. Unfortunately, the concepts for storing patent information can not, at least without significant modifications, be applied to FSs. An FS consists of rules that may allow for the combination of all fragments and therefore the number of possible products is much higher than when only decorating a core fragment. The main challenge for a substructure search method that processes FSs arise from the possible combination of fragments. A query substructure might not be directly present in any of the fragments but can be constructed by joining two or more fragments into a product. Therefore, the exploration of possible fragment connections leads to a combinatorial large number of products that exceeds the scope of todays computational facilities. Even if a method is able to avoid product enumeration, the search over fragment borders while directly processing connection rules is still a complex task. The only method handling this task is described in a US patent application[39] from 2007. The algorithm uses a modified Ullmann subgraph isomorphism algorithm that assigns parts of the query substructure onto fragments and allows fragment linkers to be assigned to multiple substructure nodes. Multiple node assignment is resolved with respect to the FS connection rules to construct products that contain the full query substructure. Though, the overall algorithmic strategy is similar to our work, neither the modifications to the Ullmann algorithm nor the reconstruction of final products are described in detail.

Here, we present a method for searching substructures in fragment spaces that avoids product enumeration by directly processing fragments and their connection rules. The method finds all products that include a given substructure even if the substructure spans over multiple fragments. The algorithm is designed to minimize the number of explored fragment connections to accomplish reasonable search times. The presented method is evaluated in three tests that mimic different drug development scenarios: the recovery of sulfonamides, a search for new substituents of a kinase inhibitor core, and the retrieval of peptide structures. All three tests are conducted in different FSs, BRICS 4k, BRICS 20k, and the KnowledgeSpace.[40]

■ PRELIMINARIES

The structural formula is closely related to the mathematical concept of graphs which allows for the direct application of graph theory and algorithms. Therefore, some graph theoretical concepts are introduced in the following.

Graph Theoretical Background. An undirected *graph* $G = (V, E)$ is a set of nodes V and edges E. Each edge $e \in E$ connects two nodes $v_1, v_2 \in V$. Two graph G_1 and G_2 are *isomorphic* if and only if a one-to-one mapping between their nodes V_1 and V_2 exists such that a pair of nodes $v_1, v_2 \in V_1$ is only connected if their images $w_1, w_2 \in V_2$ are connected. An *induced subgraph* of graph $G = (V, E)$ is a graph $G' = (V', E')$ composed of a subset of nodes $V' \subset V$ and edges $E' \subset E$ such that every edge $e = (v_1, v_2) \in E$ connecting two nodes $v_1, v_2 \in V$ is in E' if and only if $v_1, v_2 \in V'$. An *induced subgraph isomorphism* between a query graph G_1 and a target graph G_2 exists if G_1 is isomorphic to an induced subgraph of G_2, i.e., G_2 contains G_1.

A *molecular fragment graph* consist of nodes and edges representing atoms and bonds, respectively. Each edge connects two nodes if a bond connects the corresponding atoms. Nodes are labeled with atomic properties, e.g., atomic symbols, charge, aromaticity, or by an open valence. Two fragments can be combined at open valences to form a larger fragment or a molecule that is a fragment with no open valences. Edges are labeled with bond orders. The number of bonds an atom can form is bound by the atom's valence. Therefore, the node degree of a fragment graph is linearly bound. Note that we will refer to molecular fragment nodes as atoms and to edges as bonds.

A *substructure graph* describes a molecular substructure like a functional group or a molecular core. The graph describes atoms and their connecting bonds by labeled nodes and edges. Again, the node degree is linearly bound by the number of bonds an atom can form. Note that a substructure graph does not allow for the definition of stereochemical centers or alternative mesomeric or tautomeric forms.

■ METHODS

A method searching for substructures in FSs has to find all combinations of fragments that include the substructure of interest. The presented algorithm divides a query substructure in all possible substructure parts. These *subsubstructures* (SSSs) are searched inside fragments avoiding the combination of fragments to products. From all matches a solution is constructed that describes possible fragment combinations that lead to products incorporating the query substructure. Finally, the algorithm enumerates these products. In the following, each step of the algorithm is explained in detail, as illustrated in Figure 1.

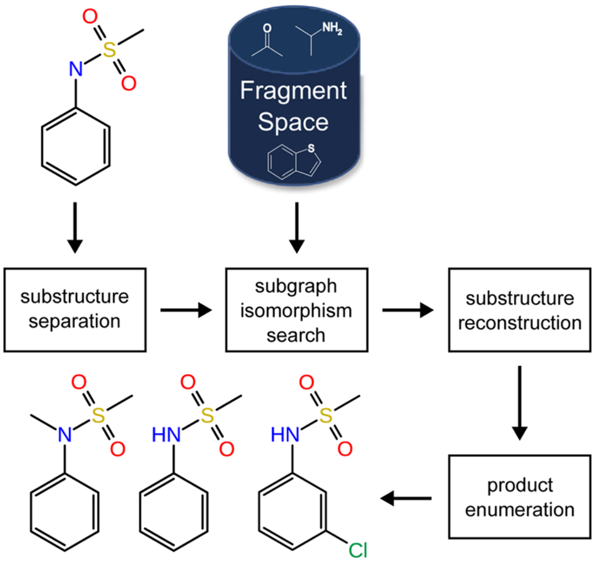


Figure 1. Workflow of matching substructures in fragment spaces (FS). A substructure is separated into subsubstructures. These are matched onto fragments of the FS. On the basis of these matches, recombination trees are constructed which form the basis for product enumeration.

Substructure Separation. A procedure that searches for substructures in FSs faces the problem that substructures might span over multiple fragments. Due to the large number of possible products, the direct examination of fragment connections is undesired. The presented algorithm avoids the combination of fragments during the search phase. It divides the query

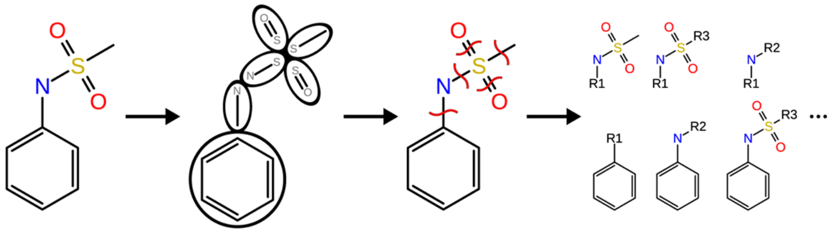


Figure 2. Separation of a query substructure into subsubstructures (SSSs). From the query substructure, cyclic and acyclic components are identified via a biconnected component algorithm. The algorithm assigns cut positions (red) and retrieves SSSs by enumerating all possible cut combinations. The figure only depicts a subset of all possible SSSs.

substructure into subsubstructures. Subsequently, these SSSs are directly mapped onto fragments such that substructure separation points are assigned to fragment linkers. In a first step, the algorithm identifies cut positions that split the query substructure in cyclic and acyclic parts. The following search procedure does not connect fragments into cycles and therefore cyclic substructure parts are not separated. The preservation of cycles is encouraged, since ring formation reactions from combinatorial chemistry usually form the same ring which can be modeled as an independent fragment. The separation algorithm detects cyclic and noncyclic substructure parts using a modified biconnected component (BCC) algorithm.[41] A BCC is either the collection of edges in a cycle or a single acyclic edge. Cut positions are assigned to all acyclic edges except edges to terminal hydrogen nodes. The resulting BCC tree is ordered by a breadth-first-search (BFS) traversal starting from an arbitrary BCC node. On the basis of the BFS order, the method enumerates all possible BCC subtrees using a subtree enumeration algorithm.[42] Since BCC subtree nodes contain substructure edges, an SSS is constructed from the substructure nodes adjacent to edges present in the BCC nodes. Such an SSS represents a part of the original query substructure. Removed substructure parts are indicated by dummy link nodes. Link nodes are labeled such that SSSs can be recombined to the original substructure. Thereby, the method separates the substructure similar to the generation of fragments from molecules. Figure 2 shows a fragmentation example.

Subgraph Isomorphism Search. Since the substructure separation step divides the query substructure into SSSs, the subgraph isomorphism search must be transferred onto the subsubstructure level as well. An FS only allows for a noncyclic connection of fragments. Therefore, a substructure edge that spans over such a connection must also be noncyclic. Since the substructure separation step guarantees that all possible SSSs are generated by splitting the substructure at noncyclic edges, a matching procedure must only find SSSs occurring inside fragments. Later on, these matches are connected to products including the complete query substructure. The modified subgraph isomorphism algorithm[12] matches each SSS against all fragments of the FS (see also Figure 3). According to the described substructure separation procedure, each cut position of an SSSs is marked with a dummy link node. During this search step, SSS nodes are subsequently assigned to fragment atoms until all nodes have a corresponding atom. Dummy nodes are only mapped onto fragment link atoms. Assuming compatible links, matched fragments can be connected at link atoms to form a product in the same way that SSSs can be connected to form the query substructure. The result of the matching phase is a list of matching fragments for each SSS referred to as the *SSS match list.*

Substructure Reconstruction. For the reconstruction of the substructure, the algorithm examines the connectivity of SSSs and fragments to ensure that the matched fragments can be combined to a product containing the query substructure. A valid combination is described by a *recombination tree.* In such a tree, nodes are represented by SSS match lists. Two lists are connected by an edge if and only if the corresponding SSSs can be connected at dummy nodes and, at the same time, the link atoms matched to the dummy nodes are compatible. In order to achieve a low number of link examinations, the algorithm splits lists that hold fragments with different matched link atoms so that each resulting list holds only fragments with the same link atom types matched to the same dummy nodes. This procedure has the advantage that the link compatibility has to be examined only once, no matter how many fragments are contained in each list. Figure 4 shows an example of a recombination tree which consists of two lists.

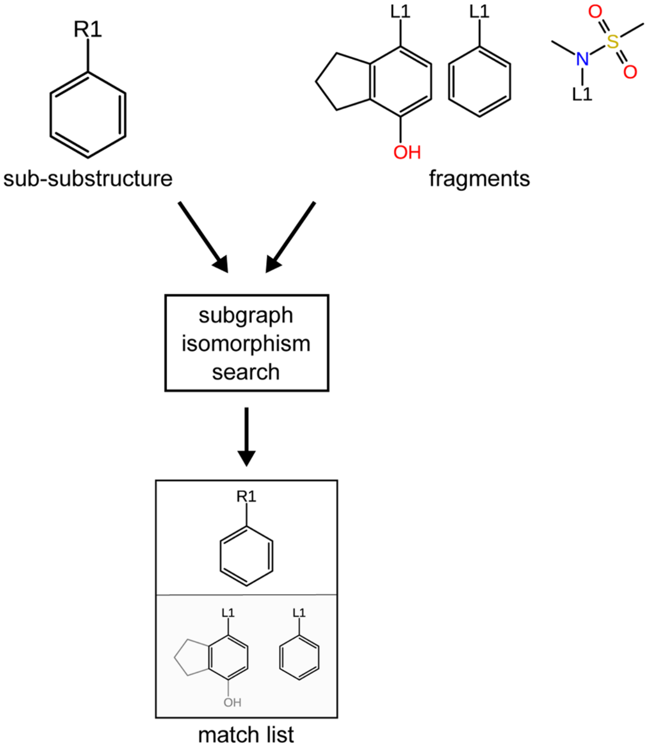


Figure 3. Subgraph isomorphism search example of an SSS that is matched against fragments of the FS. SSS dummy link nodes, labeled with R_1, are matched on fragment link atoms labeled L_1. In this example, an SSS is matched against three fragments. The result is a match list holding the SSS (top) and two matched fragments (bottom).

Product Enumeration. Each recombination tree represents a substructure separation pattern and holds the corresponding SSSs and fragments in its nodes. The tree's topology describes how SSSs can be connected to form the original query substructure.

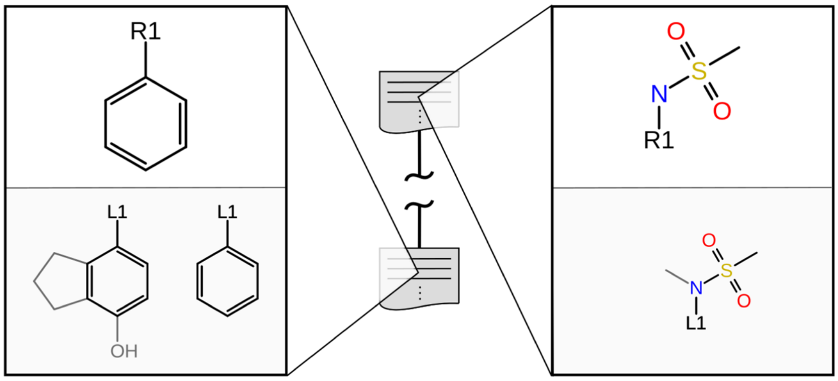


Figure 4. Recombination tree with two SSS match lists. Each list holds an SSS and fragments including that subsubstructure. The original query substructure is constructed by connecting the SSSs at dummy link nodes labeled R_1. A product can be formed by connecting one fragment from each list at the respective link atoms labeled L_1.

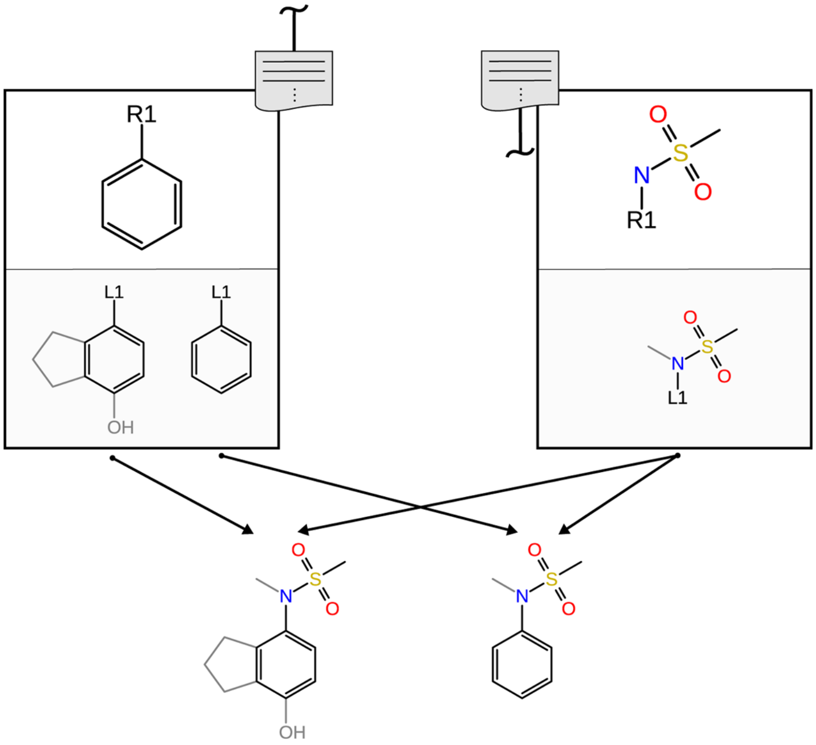


Figure 5. Enumeration of a product from a solution tree. One fragment from each match list of the solution is chosen, and fragments are connected according to their matched link atoms. The result is a molecule or a larger fragment (not shown) that includes the query substructure.

Therefore, fragments from these nodes can be connected accordingly, and the resulting product is guaranteed to include the full query substructure. For enumeration purposes, the algorithm picks one fragment per node and concatenates the fragments according to the connectivity of tree nodes. To avoid multiple enumeration of equal products, the method reduces the fragment lists stored in each node of the recombination tree such that each fragment is contained only once. Fragments are compared with regard to their orientation using a unique number assigned during the FS construction. Additionally, for each generated product the enumeration procedure compares unique SMILES strings.[43] Thereby, equal products arising from different recombination trees are identified and deleted. Repeating this procedure for each recombination tree, the method enumerates the smallest combination of fragments resulting in products including the query substructure. Figure 5 shows a simple enumeration example.

■ DATA SETS

The efficiency and usefulness of the presented algorithm to search for substructures in FSs is demonstrated in three test scenarios using three publicly available FSs. BRICS 4k and BRICS 20k[21] are generic FSs retrieved from retrosynthetical decomposition of molecules, and KnowledgeSpace[40] is compiled from various synthesis protocols.

The breaking of retrosynthetically interesting chemical substructures (BRICS) approach follows the RECAP concept by describing 16 chemical environments containing different link atoms and 64 rules for connecting them. Figure 6 depicts an example of fragment prototypes and their possible connections. BRICS 4k contains 4800 fragments, and BRICS 20k represents an enrichment of BRICS 4k with an additional 17 200 fragments to include a total of 22 000 building blocks. The BRICS spaces allow the construction of an arbitrary amount of products. A general

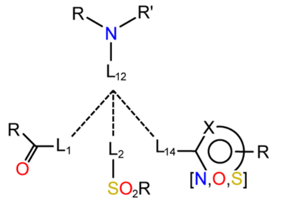


Figure 6. Subset of the BRICS fragment space connection rules. Chemical environments and the corresponding linkers (*L*) are shown. Omitted parts of environments are indicated with *R*, and *X* marks generic atoms. A dotted line between two linkers indicates their compatibility.

measure to describe the size of an FS is the number of possible products that include up to five fragments, which is about 10^{16} for the BRICS 4k space and even more for BRICS 20k.

KnowledgeSpace is based on 82 synthesis protocols obtained from the literature. The protocols cover compounds of specific targets, e.g., GPRCs, proteases, and kinases, as well as purely chemistry-driven substances. KnowledgeSpace comprises 10876 fragments with 488 distinct chemical environments and 7130 connection rules. The chemical space covered reaches about 12×10^9 possible products.

■ RESULTS

The presented algorithm is tested on three different FSs for its ability to supply alternating molecules of a defined chemical class, different decorations of an arbitrary core, and the extraction of large macromolecules. The measurements include the number of products present in each FS and the search and enumeration times needed on a single Intel(R) Xeon(R) CPU E5630 2.53 GHz core with 64 GB RAM.

■ EXPLORATION OF A CHEMICAL CLASS

Sulfonamides are the basis for several groups of drugs such as antibacterials, anticonvulsant, and diuretics. Typical sulfonamides are sulfamethoxazole, sulfadoxine, and sulfasalazine. Even though many sulfonamides are known, the search in FSs may reveal novel members with diverse physicochemical properties. Therefore, a substructure search of a sulfonamide defining pattern (Figure 7) against BRICS 4k, BRICS 20k, and KnowledgeSpace, is performed. Table 1 gives an overview of the results. The number of single fragments containing the sulfonamide substructure is rather small, e.g., 280 in BRICS 20k. Nevertheless, the combinatorial properties of an FS allow for the construction of almost 10^8 sulfonamides by combining up to three fragments from BRICS 20k.

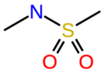


Figure 7. Query substructure defining the class of sulfonamides.

Due to the chemical environment definition of the respective FSs the maximal product size is limited to three fragments. All three FSs only contain fragments with complete sulfone groups and linking rules that allow the connection of a nitrogen to a sulfone. Furthermore, BRICS contains sulfones with two attached linkers but no linking rule allows the formation of a methyl-sulfone. Therefore, the sulfonamide substructure only spans over three fragments, one of them containing a methyl-sulfone group. KnowledgeSpace allows for the connection of carbon and sulfone but does not contain a sulfone with two attached linkers. Therefore, either the methyl-sulfone or the sulfonamide part must be inside a single fragment. Since all three FSs contain a large number of sulfonamides (Table 1), the potential to find an interesting compound is large. Figure 8 depicts examples from each FS, including commercially available molecules as well as sulfonamides not present in the PubChem database. The second group represents the majority of the retrieved products, showing the potential of the algorithm to find new lead compounds.

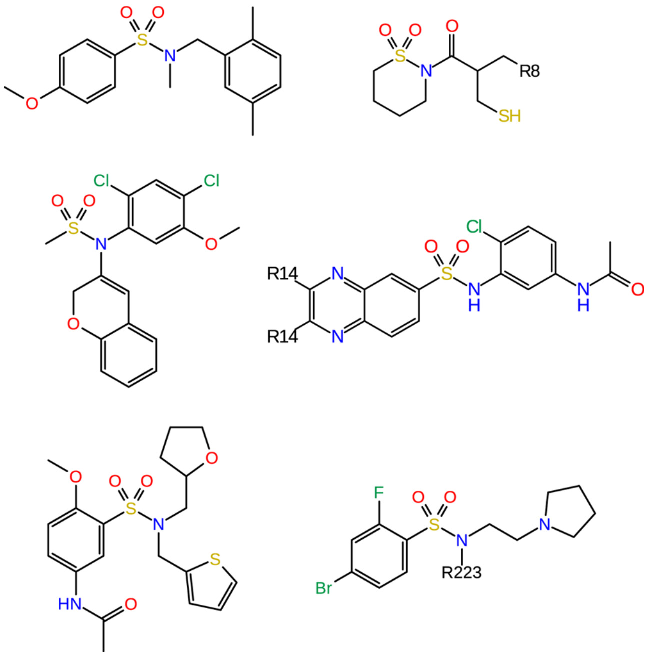


Figure 8. Sulfonamide product examples in BRICS 4k (top), BRICS 20k (middle), and KnowledgeSpace (bottom). Molecules on the left side are present in the PubChem database. Products on the right side are shown with open valences. Open valences allow the attachment of further fragments.

Table 1. Search Times, Enumeration Times, and the Number of Products with One, Two, and Three Fragments for a Search of Sulfonamides in BRICS 4k, BRICS 20k, and KnowledgeSpace[a]

			products		
	search time [s]	enum time [m]	1 fragment	2 fragments	3 fragments
BRICS 4k	5.82	2.84	26	3.2×10^4	6.3×10^5
BRICS 20k	31.60	120.08[a]	280[a]	1.3×10^6 [a]	9.3×10^7 [a]
KnowledgeSpace	18.31	49.60	33	3.3×10^4	8.6×10^6

[a] Enumerations stopped at 20 million products due to memory limitations and the number of products is calculated from recombination trees with unique fragments per node (no deduplication by enumeration).

The substructure search takes between 5 and 32 s to find all fragments that can be combined to a sulfonamide in the respective FS. Enumeration times last from 2.8 min for 63 000 to 2 h for 20 million products. These numbers demonstrate the usability of the search method, especially when considering the large number of products. Enumeration times are 2 orders of magnitude higher in comparison to the search times. An enumeration procedure must account for the possibility that the same product might be generated out of different fragment combinations. Therefore, each enumerated product is checked for uniqueness which is computationally expensive. Nevertheless, the enumeration of 20 million products takes about two hours, which we consider to be acceptable.

The options on how to further process the found products are manifold. They might be subject to further steps in a drug development process, such as similarity queries or molecular property or fingerprint filters. Another option is the construction of a focused FS which itself provides valuable opportunities for lead generation, e.g., allows the use of FS algorithms. In the presented example, a focused FS contains the fragments used in products found during the substructure search and represents a space focused on sulfonamides. For example, a constructed sulfone FS from products found in BRICS 4k contains 2986 fragments, which is 1.6-fold smaller than the original FS and allows an analysis that is more focused on sulfonamides. In general, the large number of products an FS can incorporate makes such an analysis impossible on the set of enumerated products.

■ MOLECULAR CORE DECORATION

The presented algorithm is well suited for the structure−activity relation exploration of molecular cores. Given the core as substructure, the search procedure generates products containing the core with different decorations. In this experiment, we search for alternatives of Afatinib, shown in Figure 9, developed by Boehringer Ingelheim for the treatment of solid tumors. Afatinib is a tyrosine kinase inhibitor. More precisely, it interacts with the epidermal growth factor receptor (EGFR) and the human epidermal growth factor receptor-2 kinases.[44] Figure 10 shows our definition of the basic core of Afatinib. Table 2 shows that the search retrieves 3900 and 38 967 different core decorations from BRICS 4k and BRICS 20k, respectively. A closer examination of the results shows that the 38 967 products retrieved from BRICS 20k include all 3900 products from BRICS 4k. This is an expected result, since BRICS 4k resembles a subset of BRICS 20k. KnowledgeSpace does not contain any product with the desired core structure. Figure 11 shows randomly selected examples from the BRICS fragment spaces. Molecules on the left side are examples documented in PubChem as active against EGFR kinases. A detailed visual inspection of EGFR kinase inhibitors obtained from the PubChem database shows a protonated nitrogen present at R3 of the core definition. A search for a redefined query reveals that 418 out of the 3900 products in BRICS 4k and 3703 from the original 38 967 in BRICS 20k follow this substitution pattern. A query explicitly missing such a protonated hydrogen retrieves the other 3482 and 35 264 products. Therefore, the second query confirms the ability of the search algorithm to retrieve an exact set of products. Search times are in a similar range to the sulfonamide query with 4−40 s and enumeration times of 0.7−8 s are much lower due to the lower number of retrieved products. Again, the found products can be further processed as described in the sulfonamide experiment. For example, a focused FS from BRICS 4k and BRICS 20k would contain 3388 and 16 461 fragments, respectively.

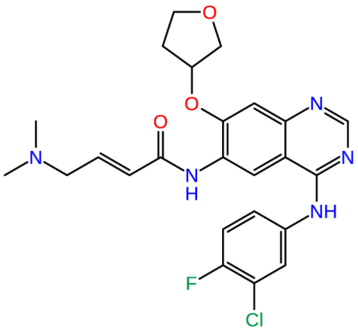


Figure 9. Afatinib structure.

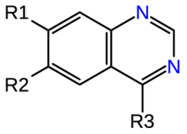


Figure 10. Basic core of Afatinib. The decoration pattern is indicated by the R-groups.

Table 2. Search and Enumeration Times for Unique Decoration of the Afatinib Core in BRICS 4k, BRICS 20k, and KnowledgeSpace

			products	
	search time [s]	enum time [s]	1 fragment	2 fragments
BRICS 4k	4.86	0.72	7	3893
BRICS 20k	39.27	7.93	7	38960
KnowledgeSpace	19.57	0.00	0	0

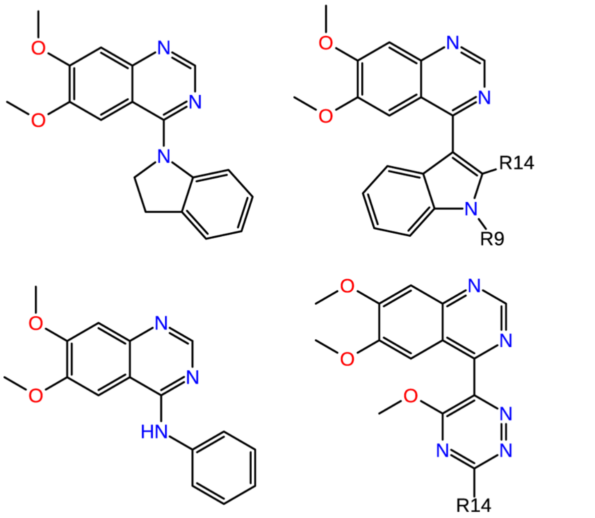


Figure 11. Examples of Afatinib core decorations in BRICS 4k (top) and BRICS 20k (bottom). Molecules on the left side are marked as active against EGFR (top) and EGFR kinases (bottom) in the PubChem database. Products on the right side are shown with open valences. Open valences allow the attachment of further fragments.

EXTRACTION OF MACROMOLECULAR STRUCTURES

Oligopeptides are short polymers of amino acids connected by peptide bonds. They are used as inhibitors for kinases, proteases, and HIV-1 assembly.[45] In order to demonstrate the abilities of our method, we search the three FSs for oligopetides with six peptide bonds, shown in Figure 12, requiring an N-terminus and a C-terminus for the oligopeptides. 13.3 million and 1.9 billion peptides are found in BRICS 4k and BRICS 20k as shown in Table 3. Figure 13 shows randomly selected examples. Search times of 1.35 min and 9.04 min are achieved on the respective FSs. KnowledgeSpace does not contain any peptide-like products. Since the run time is below 10 min for a search in BRICS 20k, we conclude that even large FSs can be searched with large query substructures in a reasonable time. However, a fingerprint or fragment-based screening step prior to the actual search removing fragments that cannot be part of the final solution would be beneficial for such complex queries. Enumeration times are 3 h for 13 million products and 4.7 h for 20 million products. Nevertheless, run times in the range of hours from a query substructure and a FS to a list of enumerated products render this method very useful. At this point, it should be noted that queries can span over multiple fragments and are not restrained by size limitations.

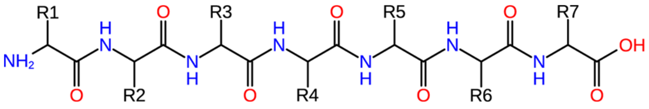


Figure 12. Peptide query substructure.

Interestingly, 45 and 102 fragments form the focused FSs from the search in BRICS 4k and BRICS 20k (see examples in Figure 13), respectively. The low number of fragments show that peptides are formed from few building blocks into billions of products. Therefore, we conclude that both FSs allow the extraction of peptides with arbitrary size by further extending the peptides with more fragments.

CONCLUSIONS

We have presented a novel method that is able to search for substructures in FSs. The method finds all products containing a desired substructure even if the query substructure spans over multiple fragments. The search is not limited in the substructure size or the number of fragments forming a product. The conducted experiments show that the search procedure is fast, below 10 min for a peptide query, especially with respect to the number of matches found and the number of possible products contained in an FS. The computationally most expensive step is the enumeration of products in order to generate a unique set. Regarding the fact that billions of products needed to be compared, we consider a run time of a few hours on a single core acceptable. Since our test products needed to be kept in memory for comparison, memory limitations where encountered at 20 million products. This limitation can be solved by using appropriate database technologies to store and compare enumerated products based on their unique SMILES identifier. Nevertheless, the general applicability and usefulness of the method in a drug development scenario has been demonstrated. The possible

Table 3. Search Times, Enumeration Times, and the Number of Peptides with Six Amide Bonds in BRICS 4k, BRICS 20k, and KnowledgeSpace[a]

			products		
	search time [m]	enum. time [h]	1−5 fragment(s)	6 fragments	7 fragments
BRICS 4k	1.35	3.03	0	5.9×10^5	1.3×10^7
BRICS 20k	9.04	4.72[a]	0	6.4×10^7[a]	1.9×10^9[a]
KnowledgeSpace	5.90	0.00	0	0	0

[a] Enumerations stopped at 20 million products due to memory limitations and the number of products calculated from recombination trees with unique fragments per node (no deduplication by enumeration).

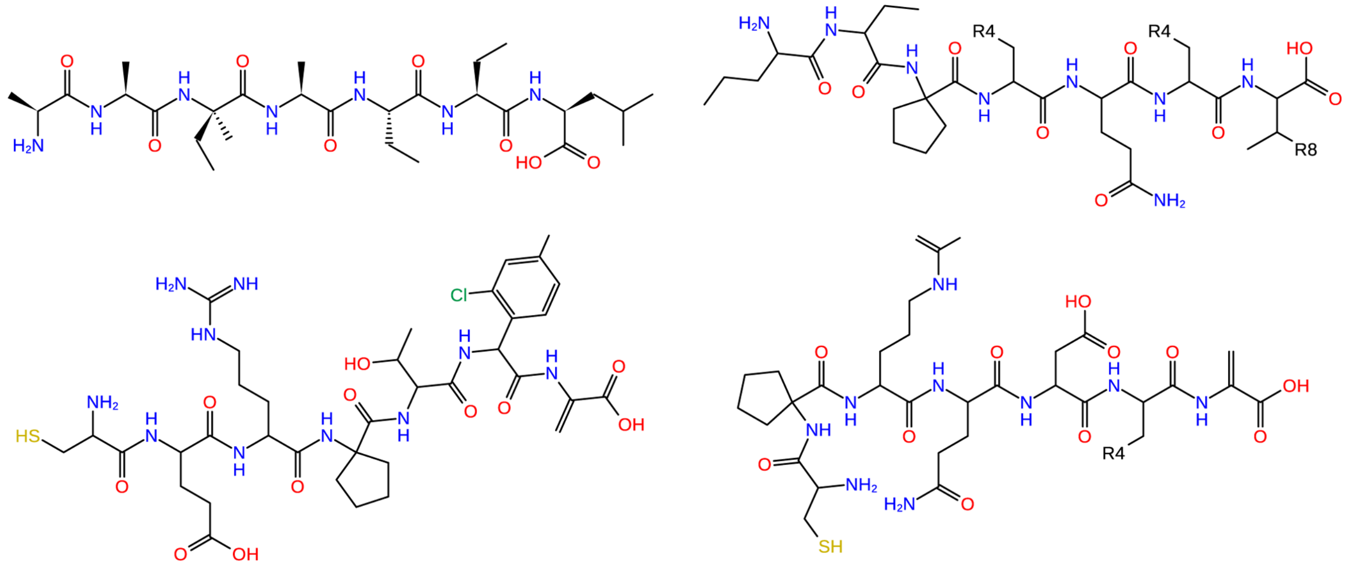


Figure 13. Examples of peptides extracted from BRICS 4k (top) and BRICS 20k (bottom).

products of an FS might be a valuable source of interesting and novel molecules not contained in publicly available databases. With respect to the number of retrieved matches, products might be visually inspected or subject to further steps in drug development such as analog searches or molecular property filters. A valuable option is the creation of a focused FS from the search results which reduces the number of fragments and focuses the FS for further investigation.

The substructure search method handles explicit substructures quite well. Unfortunately, chiral and generic expressions, such as substructure nodes that match a set of different molecule atoms, alternative tautomeric forms, or atomic properties, e.g., atoms with a defined number of neighbors, are currently not supported. Another limitation of the method is the restriction of cyclic structures occurring only inside fragments. The algorithm will therefore not find structures describing large macromolecular cycles. In most cases, however, the formation of such cyclic structures can be circumvented by a careful FS design. Our future work will extend the search procedure to handle substructure queries with variable atom and bond type definitions as well as logical alternatives, e.g., alternative tautomeric forms, such as present in the Smiles arbitrary target specification (SMARTS).[46]

AUTHOR INFORMATION

Corresponding Author

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

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

Thanks to Florian Lauck and others for revising the manuscript.

REFERENCES

(1) Irwin, J.; Shoichet, B. ZINC—a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.

(2) Bolton, E. E.; Wang, Y.; Thiessen, P. A.; Bryant, S. H. PubChem: Integrated Platform of Small Molecules and Biological Activities. In *Annual Reports in Computational Chemistry*; Wheeler, R. A., Spellmeyer, D. C., Eds.; Elsevier: New York, 2008; Vol. *4*, Chapter 12, pp 217−241.

(3) Sussenguth, E. H. A Graph-Theoretic Algorithm for Matching Chemical Structures. *J. Graph. Theor.* **1965**, *5*, 36−43.

(4) Figueras, J. Substructure Search by Set Reduction. *J. Graph Theory* **1972**, *12*, 237−244.

(5) Read, R. C.; Corneil, D. G. The graph isomorphism disease. *J. Graph Theory* **1977**, *1*, 339−363.

(6) Gati, G. Further annotated bibliography on the isomorphism disease. *J. Graph Theory* **1979**, *3*, 95−109.

(7) Ullmann, J. R. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31−42.

(8) Attias, R. DARC substructure search system: a new approach to chemical information. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 102−108.

(9) Heyman, J.; Karasinskia, E.; Giles, P. CAS information services for medicinal chemists. *Drug Inf. J.* **1982**, *16*, 185−190.

(10) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Model.* **1998**, *38*, 983−996.

(11) Cordella, L.; Foggia, P.; Sansone, C.; Vento, M. Performance evaluation of the VF graph matching algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, Venice, Italy, Sep 27−29; IEEE Computer Society, 1999, pp 1172−1177.

(12) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE J. Pattern. Anal.* **2004**, *26*, 1367−1372.

(13) Yan, X.; Yu, P. S.; Han, J. Substructure similarity search in graph databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, Baltimore, MD, June 13−17; ACM, New York, 2005; pp 766−777.

(14) Golovin, A.; Henrick, K. Chemical Substructure Search in SQL. *J. Chem. Inf. Model.* **2009**, *49*, 22−27.

(15) Raymond, J. W.; Gardiner, E. J.; Willett, P. RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. *Comput. J.* **2002**, *45*, 631−644.

(16) Raymond, J.; Gardiner, E.; Willett, P. Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 305−316.

(17) Chao, S.-Y. Maximum Common Substructure Extraction in RNA Secondary Structures Using Clique Detection Approach. *World Acad. Sci., Eng. Technol.* **2008**, *45*, 219−228.

(18) Schietgat, L.; Ramon, J.; Bruynooghe, M.; Blockeel, H. An Efficiently Computable Graph-Based Metric for the Classification of Small Molecules. In *Proceedings of the 11th International Conference on Discovery Science*, Budapest, Hungary, Oct 13−16; Springer-Verlag: Berlin, Heidelberg, Germany, 2008; pp 197−209.

(19) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(20) 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. Comput. Sci.* **1998**, *38*, 511−522.

(21) 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.

(22) Boehm, M.; Wu, T.-Y.; Claussen, H.; Lemmen, C. Similarity searching and scaffold hopping in synthetically accessible combinatorial chemistry spaces. *J. Med. Chem.* **2008**, *51*, 2468−2480.

(23) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with feature trees. *J. Chem. Inf. Model.* **2009**, *49*, 270−279.

(24) Rarey, M.; Stahl, M. Similarity searching in large combinatorial chemistry spaces. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 497−520.

(25) Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. "Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew. Chem., Int. Ed. Engl.* **1999**, *38*, 2894−2896.

(26) Schneider, G.; Clément-Chomienne, O.; Hilfiger, L.; Schneider, P.; Kirsch, S.; Böhm, H.-J.; Neidhart, W. Virtual Screening for Bioactive Molecules by Evolutionary De Novo Design. *Angew. Chem., Int. Ed.* **2000**, *39*, 4130−4133.

(27) Hartenfeller, M.; Proschak, E.; Schüller, A.; Schneider, G. Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. *Chem. Biol. Drug. Des.* **2008**, *72*, 16−26.

(28) Lippert, T.; Schulz-Gasch, T.; Roche, O.; Guba, W.; Rarey, M. De novo design by pharmacophore-based searches in fragment spaces. *J. Comput.-Aided Mol. Des.* **2011**, *25*, 931−945.

(29) 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*, e1002380.

(30) Good, A. C.; Lewis, R. A. New methodology for profiling combinatorial libraries and screening sets: cleaning up the design process with HARPick. *J. Med. Chem.* **1997**, *40*, 3926−3936.

(31) Gillet, V. J.; Willett, P.; Fleming, P. J.; Green, D. V. S. Designing focused libraries using MoSELECT. *J. Mol. Graphics Modell.* **2002**, *20*, 491−498.

(32) Fischer, J.; Lessel, U.; Rarey, M. LoFT: Similarity-Driven Multiobjective Focused Library Design. *J. Chem. Inf. Model.* **2010**, *50*, 1−21.

(33) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer storage and retrieval of generic chemical structures in patents. 2. GENSAL, a formal language for the description of generic chemical structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151−161.

(34) Lynch, M. F.; Holliday, J. D. The Sheffield Generic Structures Projecta Retrospective Review. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 930−936.

(35) Downs, G. M.; Barnard, J. M. Chemical patents and structural information - the Sheffield research in context. *J. Doc.* **1998**, *54*, 106−120.

(36) Benichou, P.; Klimczak, C.; Borne, P. Handling Genericity in Chemical Structures Using the Markush Darc Software. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 43−53.

(37) Fisanick, W. The Chemical Abstract's Service generic chemical (Markush) structure storage and retrieval capability. 1. Basic concepts. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 145−154.

(38) Ebe, T.; Sanderson, K. A.; Wilson, P. S. The Chemical Abstracts Service generic chemical (Markush) structure storage and retrieval capability. 2. The MARPAT file. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 31−36.

(39) Domine, D.; Cedric, M. *Method for fast substructure searching in non-enumerated chemical libraries*. US Patent Application US 2007/0260583 A1, 2007.

(40) Detering, C.; Claussen, H.; Gastreich, M.; Lemmen, C. KnowledgeSpace - a publicly available virtual chemistry space. *J. Cheminf.* **2010**, *2*, O9.

(41) Hopcroft, J.; Tarjan, R. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* **1973**, *16*, 372−378.

(42) Rarey, M.; Dixon, J. S. Feature trees: a new molecular similarity measure based on tree matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471−490.

(43) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.

(44) Minkovsky, N.; Berezov, A. BIBW-2992, a dual receptor tyrosine kinase inhibitor for the treatment of solid tumors. *Curr. Opin. Investig. Drugs* **2008**, *9*, 1336−1346.

(45) Owens, R. J.; Tanner, C. C.; Mulligan, M. J.; Srinivas, R. V.; Compans, R. W. Oligopeptide inhibitors of HIV-induced syncytium formation. *AIDS Res. Hum. Retroviruses* **1990**, *6*, 1289−1296.

(46) *Daylight Theory Manual*, version 4.9; Daylight Chemical Information Systems Inc.: Aliso Viejo, CA, 2008.