JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

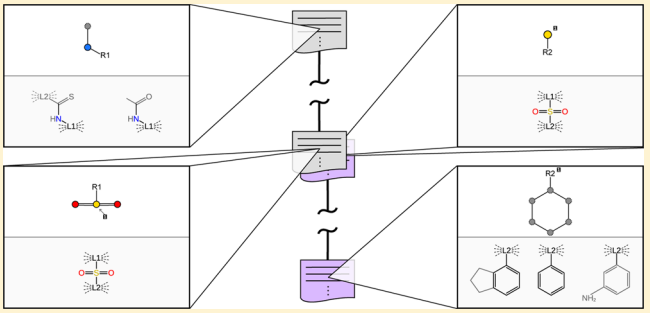
Searching for Recursively Defined Generic Chemical Patterns in Nonenumerated Fragment Spaces

Hans-Christian Ehrlich, Angela M. Henzler, and Matthias Rarey*

University of Hamburg, Bundestraße 43, 20146 Hamburg, Germany

Supporting Information

ABSTRACT: Retrieving molecules with specific structural features is a fundamental requirement of today's molecular database technologies. Estimates claim the chemical space relevant for drug discovery to be around 10^{60} molecules. This figure is many orders of magnitude larger than the amount of molecules conventional databases retain today and will store in the future. An elegant description of such a large chemical space is provided by the concept of fragment spaces. A fragment space comprises fragments that are molecules with open valences and describes rules how to connect these fragments to products. Due to the combinatorial nature of fragment spaces, a complete enumeration of its products is intractable. We present an algorithm to search fragment spaces for generic chemical patterns as present in the SMARTS chemical pattern language. Our method allows specification of the chemical surrounding of an atom in a query and, therefore, enables a chemically intuitive search. During the search, the costly enumeration of products is avoided. The result is a fragment space that exactly describes all possible molecules that contain the user-defined pattern. We evaluated the algorithm in three different drug development use-cases and performed a large scale statistical analysis with 738 SMARTS patterns on three public available fragment spaces. Our results show the ability of the algorithm to explore the chemical space around known active molecules, to analyze fragment spaces for the presence of likely toxic molecules, and to identify complex macromolecular structures under additional structural constraints. By searching the fragment space in its nonenumerated form, spaces covering up to 10^{19} molecules can be examined in times ranging between 47 s and 19 min depending on the complexity of the query pattern.



INTRODUCTION

Various *in silico* applications in drug discovery are confronted with the exploration of molecular databases. Often, the identification of specific query molecules or the filtering of databases according to predefined structural properties is of interest. Hence, most chemical software tools support database searches employing molecule, substructure, or chemical pattern matching algorithms. With increasing size of molecular databases, conventional molecule and especially chemical pattern search[1−12] becomes demanding concerning storage and search time requirements. With some exceptions, e.g., the Chemical Universe Database GDB-13, which comprises around 970 million entries,[13] molecular databases generally store only a few million compounds.[14,15] However, the size of the chemical space is estimated to be much larger. Reported numbers range between 10^{18} and 10^{200} with a consensus around 10^{60} possible structures,[16−18] amounts that by far exceed the critical limit of storage capacities provided by conventional databases. In order to allow a sampling of the chemical space, efficient storage and search strategies are required. Two concepts for this exist: fragment spaces[19−22] (FS's) and Markush structures.[23−26] Both allow a very compact description by an efficient graph-based representation of a large chemical space. Moreover, they enable the application of efficient graph-matching algorithms for molecular search.[27−31]

Fragment spaces are described by molecular fragments and connection rules. In this context, a molecular *fragment* is a graph composed of nodes and edges that represent atoms and bonds. Nodes are labeled with atomic properties, e.g., chemical symbol, charge, or open valence, whereas edges are labeled with bond orders. The number of edges attached to one node is linearly limited by the number of bonds an atom can form. Fragments possess *link atoms* that indicate open valences. Larger fragments, which we refer to as products, or molecules can be built by linking fragments according to their connection rules. Molecules are products without link atoms. The connection rules of an FS define how link atoms of different types can be combined and, thus, which products are encoded by the FS. Fragment spaces are obtained by retrosynthetic cleavage of complete molecules and forming a consensus on the resulting fragments. They intend to reflect the knowledge about reaction schemes of combinatorial chemistry. The combinatorial nature of FS's allows description of a large number of molecules with only a few fragments and rules; for example, the

Received: February 14, 2013
Published: June 11, 2013

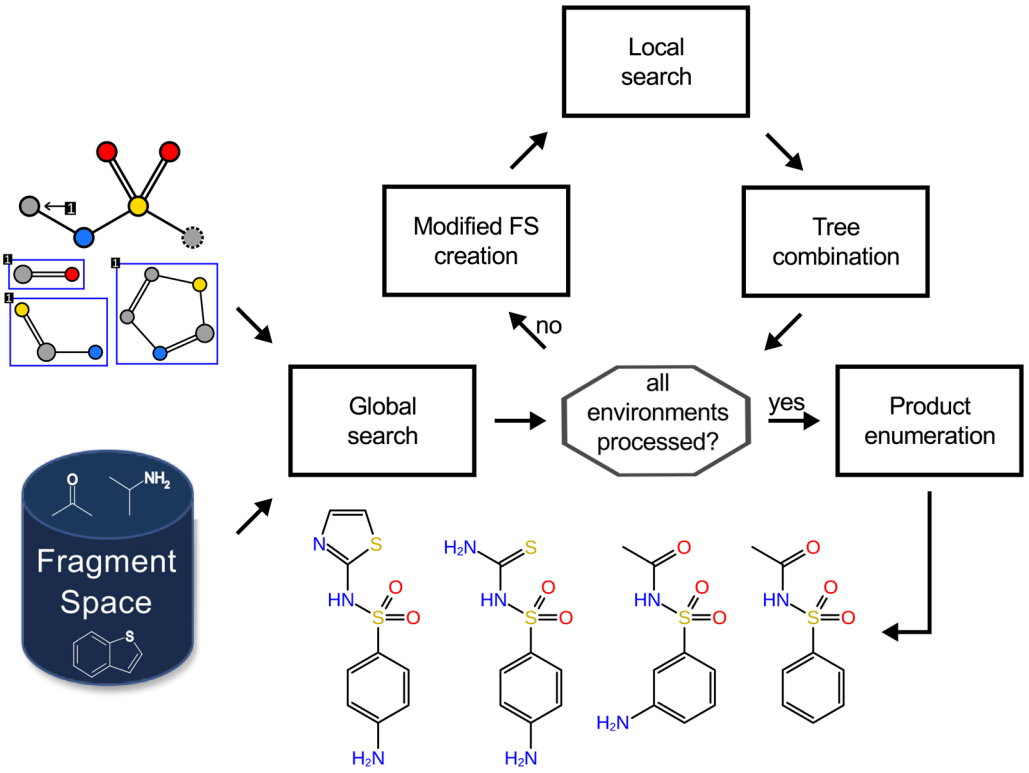


Figure 1. Pattern search for sulfonamides under additional structural constraints. The method utilizes a global search that scans the FS for the pattern neglecting all atomic environments. The algorithm subsequently adds all environments. In an iterative procedure, a modified FS is created for each recombination tree obtained from the global search. These spaces are searched for an environment pattern in a local search. The global tree is modified using the local search results, which generates multiple trees. The procedure is repeated for all trees until all recursive environments are fulfilled. From the final result, molecular products are enumerated to obtain complete molecules. (Substructure depiction created with SmartsViewer.[47])

BRICS 4k space[21] containing 4800 fragments and 64 connection rules encodes over 10^{16} molecules.

The concept of *Markush structures* is quite similar to that of FS's. Markush structures are designed to represent a series of homologous molecules. They usually consist of a single core fragment containing R-groups that are open attachment points. For each group, a list specifies alternative fragments or generic structures, e.g., any alkene. In general, it is undesired to explicitly enumerate the space encoded by Markush structures. Thus, database systems storing Markush structures, e.g. GENSAL,[29] Markush Darc,[30] and MAPRAT,[28] process and search entries avoiding an explicit enumeration.

Even though both concepts are quite similar, there is still a need for exact molecule, substructure, and chemical pattern search methods on FS's. Methods that process and search databases of Markush structures are not transferable to FS's. The main reason is that algorithms on FS's require completely resolved atoms of the fragments. This is contradictory to the concept of generic substituents in Markush structures, as they do not always have an explicit structural counterpart in the graph description. Current algorithms operating on FS's support mainly similarity searching,[31,32] the design of novel molecules,[33−36] and the creation of focused compound libraries[37−39] for virtual screening campaigns. Procedures for substructure search in FS's are rare. Domine and Cedric applied for a patent for substructure searches in nonenumerated chemical spaces that use a modified Ullmann algorithm to assign parts of the substructure to fragments which are subsequently reassembled into products.[40] However, the patent description misses details of the algorithmic concepts. We recently introduced a similar method for substructure search in FS's.[41] It recognizes the fact that a substructure can span over multiple fragments and avoids an explicit enumeration of products during the search.

The SMiles ARbitrary Target Specification (SMARTS),[42] as well as other *chemical pattern languages* such as the Sybyl Line Notation[43] or the Molecular Query Language (MQL),[44] defines properties of atoms and bonds and the topology of chemical patterns. Provided that the query in the form of a textual line notation is transferred to a graph, chemical pattern searches can benefit from graph matching algorithms.[5,9,10,45] Graphs describing such chemical patterns are further referred to as *pattern graphs*. As opposed to graphs representing molecules or substructures, the nodes and edges can be labeled with a generic description of atoms and bonds. The generic description accounts for a broader spectrum of atom and bond properties which can be combined in logical expressions. A useful and practical feature of the SMARTS language is its ability to specify recursive *atomic environments*. They describe the molecular surrounding of an atom, e.g., a nitrogen that is part of a sulfonamide group, and they are well suited to describe local alternatives such as mesomeric and tautomeric forms. Chemical atomic environments can be modeled as pattern graphs assigned to nodes of a higher level pattern graph. Due to the recursive nature of atomic environments, chemical patterns can be nested in patterns. As a result, an exact pattern search has to recursively traverse nested patterns in order to identify matching molecules.

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

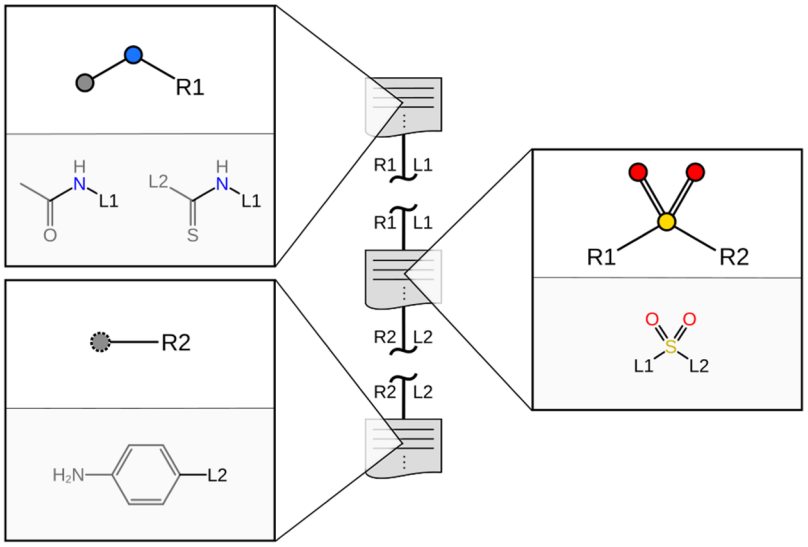


Figure 2. Recombination tree gained from a sulfonamide query search. The tree contains three nodes, each holding an SP, corresponding fragments, and the assignment of the SP to the fragments. The SP's are compatible at dummy link node *R*1, respectively *R*2, to form the sulfonamide query pattern. Fragments can be connected via linker *L*1, respectively *L*2, to form a sulfonamide product.

In this paper, we introduce the extension of our substructure search algorithm for nonenumerated FS's[41] with regard to SMARTS queries with recursive atomic environments. In order to promote a profound understanding, we initially overview the workflow of our new method and briefly describe the terminology and the basic concept of our previously introduced search procedure. We continue with a step-by-step explanation of the adaptations and extensions necessary to account for atomic environments. The method is evaluated by three use-case studies that simulate different molecular modeling scenarios. Each of them is conducted on the BRICS 4k, the BRICS 20k,[21] and the KnowledgeSpace[46] FS's. The results demonstrate that the method creates subspaces in which all molecules contain a specific chemical pattern. Concluding studies on two publicly available data sets finally reflect the run time behavior of our new method. At the current stage, a cyclic connection of fragments and stereoisomeric chemical patterns is not addressed by the algorithm.

METHOD

The search strategy for patterns with recursive atomic environments follows the workflow depicted in Figure 1. In a first step, named *global search*, the algorithm processes the *global pattern* which is the query pattern neglecting the atomic environment information. The obtained global result is subsequently modified by *atomic environment searches*. An atomic environment is a pattern that defines the chemical surrounding of an atom expressed as recursive SMARTS. In the following, we will refer to such an atom as the *reference atom* of the environment. When searching in FS's, such an atomic environment either can be found in fragments itself or might be present in a combination of fragments. In the latter case, the atomic environment can only be indirectly identified by attaching additional fragments. In order to determine whether an atomic environment exists and which case occurs without an enumeration of products, the atomic environment search follows a recurring three step procedure: a modified FS is created from the global result, the new space is scanned for the atomic environment pattern, and the obtained results are combined. The search is repeated until all atomic environments are fulfilled or no acceptable combination of fragments can be found. Both searches utilize the *generic search strategy* to scan FS's for patterns without atomic environment information. At the end, the method enumerates the actual products.

Generic Search Strategy. In FS's a pattern can span over multiple fragments. To avoid a time-consuming exploration of fragment connections during the pattern search, the generic search follows the previously described strategy[41] by separating the query pattern into subpatterns (SP's), similarly to the separation of molecules into fragments. An SP is a connected part of the original pattern in which cyclic parts are fully contained and missing pieces of the pattern are indicated by dummy link nodes. The actual subgraph isomorphism step is a search that matches SP's against fragments and assigns dummy link nodes to fragment link atoms, avoiding the costly exploration of fragment link connections. The method records a list of matching fragments for each SP. Since the fragments of a list might have different link types assigned to the same dummy node, the lists are split to obtain smaller lists with unique underlying fragment linkers. This procedure has the advantage that the link compatibility has only to be checked once, no matter how many fragments are stored in each list. In preparation for the reconstruction process of molecules containing the complete query pattern, these lists are used as nodes to build a *recombination tree*. Each node stores fragments, an SP, and its possible assignments to fragments. In a recombination tree, two nodes are connected if and only if their SP's can be connected according to the connectivity of the query pattern and if the fragment link atoms are compatible according to the connection rules of the FS. The nodes are connected by two half edges. Each half is labeled with the dummy node of the SP and the corresponding link atoms. Therefore, a recombination tree describes a separation of the pattern and the associated combination of fragments. Since a pattern can be separated in various ways which lead to different sets of SP's, the result of the search is a collection of recombination trees describing a subspace of the searched FS. Figure 2 shows an example of a recombination tree.

Global Search. The global search scans an FS for the presence of the global pattern. It utilizes the generic search

strategy to gain a set of global recombination trees that describe combinations of fragments that include the global pattern but omits the atomic environment information. The environments of reference atoms are resolved by atomic environment searches.

Atomic Environment Search. Atomic environments of reference atoms occurring in the resulting trees are resolved by a three-step iterative procedure: for each combination of an atomic environment and a global recombination tree, a modified FS is constructed. The new FS enforces the combinations of fragments described by the tree. After the creation, the space is scanned with the local search strategy for the respective environment pattern. The result is a set of local recombination trees. As described below, each local tree is combined with the global tree to describe matches that include the global pattern and obey the environment information. The method repeats this process of FS creation, search, and result combination until all atomic environments are processed. The final result is a collection of recombination trees describing a portion of the FS that includes products incorporating the query pattern with regard to the environment information.

The SMARTS pattern language allows the use of atomic environments in combination with logical expressions such as logical NOT, AND, OR, and WEAK-AND. A logical WEAK-AND is a SMARTS specific term that serves as a replacement for a logical grouping of AND and OR terms; for example, the term "*A* WEAK-AND *B* OR *C*" represents "*A* AND (*B* OR *C*)". Such a definition is internally resolved into a disjunctive normal form (DNF) that only contains logical NOT, AND, and OR, e.g., "*A* AND *B* OR *A* AND *C*". In addition, each ANDed term is sorted such that non-negated environments precede negated environments. In the following, we describe the iteratively performed modification of a single global recombination tree depending on the logical markup given by the DNF. The other trees of the global result are accordingly modified. Figure 4 gives a schematic example.

The highest logical level of a DNF is a logical OR that combines logical terms in which negated and non-negated atomic environments are exclusively ANDed. If logical terms are ORed, they enforce the presence of an environment described by at least one of the terms. For each ORed term, the global recombination tree is duplicated and modified according to the logical AND description of the term. Therefore, the duplicated tree is updated to obey the atomic environments of the term. The final result is a collection of recombination trees describing the smallest combinations of fragments that include the query pattern and fulfill the logical specification of all atomic environments.

One level down in a DNF is a logical AND connecting, potentially with logical NOT negated, atomic environments. If environments are logically ANDed, the global recombination tree must be modified to simultaneously obey multiple atomic environments. The procedure processes environments in the essential order of non-negated followed by negated environments. For every non-negated environment, the inclusion of a single environment results in a modified global recombination tree in which fragments that contradict the AND description are removed from the nodes or additional fragments are attached to ensure that the environment is present. This inclusion modification is repeated on the resulting tree for each non-negated environment. If environments are negated, they have to be explicitly excluded from the recombination trees. The exclusion of a single environment from a tree results in multiple recombination trees. For each tree resulting from an exclusion, the next environment is excluded until all negated environments are processed. The result is a set of recombination trees that describe products including the global pattern and all logically ANDed environments.

In the following, we explain one iteration of the atomic environment search for the modification of a recombination tree to obey a single negated or non-negated environment specification. Recursively defined environments that define the surrounding of a reference atom with another atomic environment enforce an additional search. In such a case, the current local pattern is treated as the global pattern and the additional environment pattern is subjected to the atomic environment search.

Modifying the Fragment Space. A pattern that spans over multiple linkers might define an atomic environment that also spans over a number of fragments. A search method must therefore detect the common set of fragments and link atoms between trees obtained from the global and local search to decide whether the atomic environment exists or if compatible fragments need to be added to satisfy the atomic environment specification. This problem is addressed by creating a modified FS from a recombination tree. Such a space includes all fragments and link rules from the original space. In addition, the space contains all fragments of the tree in a modified fashion: the matched link atoms of its matching fragments are renamed, and connection rules are added that only allow the combination of these fragments according to the recombination tree. Unmatched link atoms are also renamed, and rules that prohibit a linkage to other matched fragments are formulated. A connection to unmatched fragments according to the original link connections is still possible to allow the further attachment of fragments. The renaming scheme allows a clear differentiation between matched and unmatched linkers. Figure 3 illustrates the process of constructing a modified FS.

Local Search. After constructing the modified FS, the method searches for the associated atomic environment pattern in the modified space. For the atomic environment under consideration, first all reference atoms have to be identified using the recombination tree. The search starts at such an atom in the corresponding fragment and proceeds using the generic search to detect the atomic environment pattern in the modified FS. The result is a set of local recombination trees which can be empty, indicating that the environment was not found. Otherwise, each local tree of the set contains modified fragments from the global recombination tree or unmodified fragments. If fragments from the global tree are identified, a connection to other fragments is only possible via renamed linkers. Since the renaming allows a clear differentiation between matched and unmatched linkers, the common set of fragments and linkers in the global and local recombination trees can be detected, which allows the following combination of results.

Combining Recombination Trees. From the global and the local search the obtained matches are represented by a global recombination tree and a set of local recombination trees. The aim of the next step is to combine these recombination trees such that they describe the matching of the global pattern including the atomic environment represented by the local trees. If the set of local recombination trees is empty and the corresponding atomic environment is not negated, the global tree is discarded because the atomic environment was not found. If the environment is negated, the global tree is not

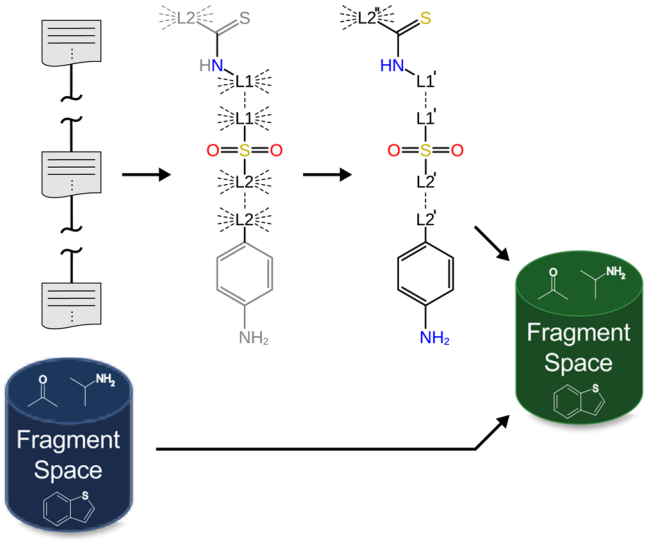


Figure 3. Modification of an FS after a sulfonamide search. The original FS is enriched with three fragments obtained from the recombination tree. These fragments are modified such that the matched link atoms L_1 and L_2 are renamed to L_1' and L_2' and a connection rule is added that only allows a connection of the two shown fragments to form a sulfonamide. The unmatched link atom L_2 is renamed to L_2'' such that a linkage to matched fragments is prohibited and connections to other fragments are still allowed; for example, if $L2$ is compatible to $L1$, the new FS allows a connection of $L2''$ to $L1$ but not to $L1'$. Note that all products previously included in the FS are still contained in the modified FS, since no fragments or rules are removed.

modified. Otherwise, each local recombination tree is subsequently superimposed with the global recombination tree as follows: both trees share at least one common fragment, the fragment containing the reference atom. Initially, the associated pair of tree nodes are superimposed, and subsequently, their incident edges are searched for equally renamed linkers. This procedure ensures that both the global and the local match use identical linkers to attach the same fragments. The superimposition proceeds for adjacent nodes containing fragments that can be attached by equally renamed linkers. If linkers have no appropriate counterpart, the node is not superimposed. Figure 4 shows an example for the superimpositions of recombination trees.

For each superimposition of the global and a single local recombination tree, the procedure modifies the global tree depending on the logical markup of the atomic environment pattern. If an environment is non-negated, the molecular surrounding of the reference atom must be included in the global recombination tree. An *inclusion* demands that the global recombination tree is modified to describe combinations of fragments that include the global pattern and the atomic environment pattern at the same time. To generate such a tree, the procedure intersects the set of matched fragments associated with superimposed nodes and stores the resulting set in the corresponding global tree node. If the resulting set is empty, the global tree is discarded. Local tree nodes not superimposed are attached to the global recombination tree in order to add fragments that fulfill the atomic environment specification.

If an atomic environment is negated, it must not exist around the reference atom. In order to *exclude* it, multiple recombination trees are generated. For each local tree node, the global tree is duplicated. The procedure modifies this tree copy according to the currently chosen local tree node: if it was previously superpositioned, the set exclusion of the local fragments from the global fragments is stored in the corresponding node of the tree copy. If the exclusion generates an empty set, the tree copy is discarded. Otherwise, the node stores only fragments that include the global SP but do not contain the atomic environment SP. Consequently, the full environment is not formed when the fragments of the tree copy are connected. To exclude the environment of a node that is not superimposed, a new node is attached to the tree copy that stores an inversion of the matched fragments present in the current local tree node. A set of fragments is inverted by excluding the set from all fragments of the FS that are compatible with the current linker. Again, if the inversion results in an empty set, the tree copy is not further considered. If such a node is not directly attached to a superimposed node, the path to this node is attached to the copy of the global tree and the set of fragments of the node is inverted.

Creating a new tree for every node related to the excluded atomic environment generates all trees that describe matches that do not include the negated pattern. Even if additional fragments are attached to these products, the environment is guaranteed to not emerge around the reference atom.

Example: Atomic Environment Search. Figure 4 shows four iterations of the atomic environment search. In this example, the global search resulted in a global recombination tree (white). This tree is extended to follow the logical expression "*X* AND *Y* OR *Z* AND NOT *W*" that specifies the environment of an atom. We assume this reference atom is contained in fragments localized in the set *A* of the global tree. In the first iteration, the atomic environment specification *X* is detected utilizing the local search on a modified fragment space and the search results in the blue tree. The white and the blue trees are superimposed. The inclusion of the first environment *X* (blue) in the global tree is realized by intersecting the fragment sets *A* and *D*. The resulting set stores fragments that simultaneously contain the global SP and the atomic environment SP of *X*. The fragments stored in *E* are attached to the global tree, to complete the atomic environment *X*.

In the following two iterations, the resulting intermediate tree (gray) is modified to include the atomic environment *Y* (turquoise and yellow). Since the local search for the environment *Y* gains two recombination trees, the intermediate tree is duplicated and the copies are extended to include the associated atomic environment in the second iteration: the first turquoise environment is present in a subset of fragments contained in the intermediate tree; therefore, the corresponding sets of fragments are intersected. The second turquoise environment is partially contained in the set $A \cap D$ and *E*. These sets are intersected, and the fragments in set *K* are attached. The two resulting green trees describe products that obey the complete logical specification, since the second part of the logical environment specification is ORed.

The third iteration processes the second part of the logical OR term. The global recombination tree is duplicated and the atomic environment *Z* represented by the purple tree is included. The atomic environment *W* is negated; accordingly the fourth iteration detects the combinations of fragments that contain the atomic environment (red tree) and generates the trees that exclude the respective environment. The result of the exclusion is a set of four trees. In the first of these trees, the atomic environment is excluded by removing the fragments in

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

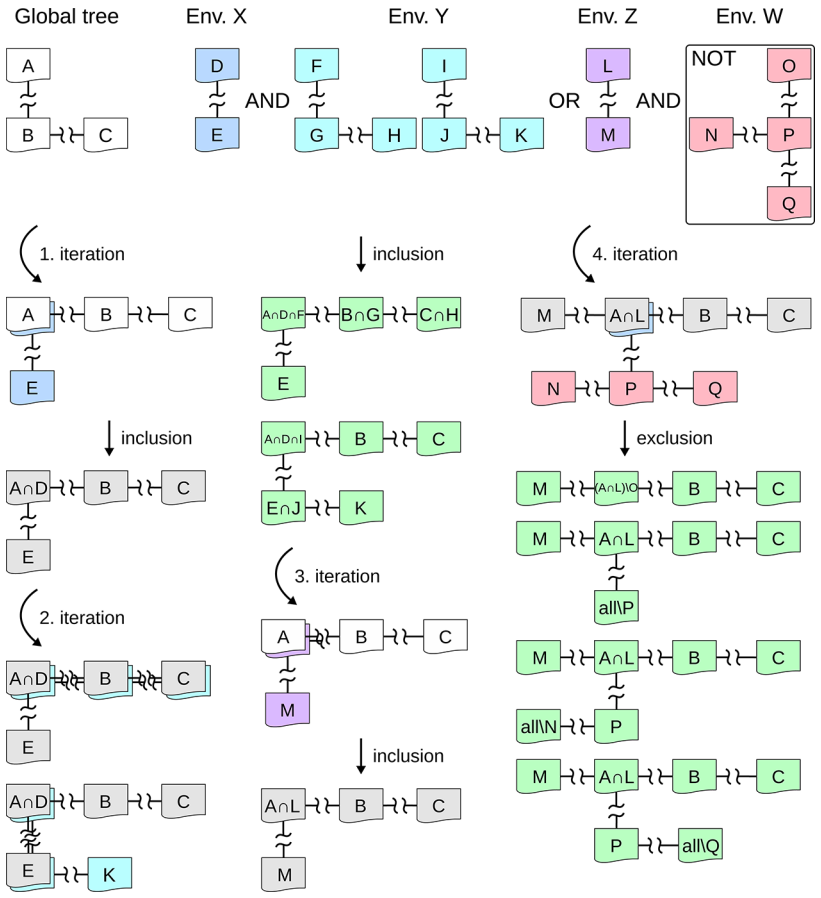


Figure 4. Example showing four iterations of the atomic environment search to include an atomic environment description of the form "*X* AND *Y* OR *Z* AND NOT *W*". The white tree shows the global recombination tree. The logical parts of the atomic environment descriptions are color-coded in blue, turquoise, purple, and red (top). Intermediate trees are gray. Combined trees that represent the result of the four iterations are shown in green. The letters *A* to *Q* represent the sets of fragments contained in nodes of recombination trees. Expressions such as $A \cap D$ indicate the set resulting from intersecting the fragment sets *A* and *D*. Fragment sets such as *all**P* hold all fragments of the fragment space compatible at the current linker without fragments of corresponding set *P*.

set *O* from the fragments in set $A \cap L$. Therefore, any fragments can be attached at fragments of set $(A \cap L)\backslash O$ and the atomic environment *W* will not emerge. In the second exclusion tree, the inversion of set *P* is attached, which prohibits the formation of atomic environment *W*. In the last two exclusion trees, the modification attaches the path to tree nodes and stores the inversion of sets *N* and *Q*. All green trees in Figure 4 describe combinations of fragments that follow the complete logical environment specification and depict the result of processing the atomic environment for a reference atom. If the overall query pattern includes additional atomic environments, these are processed in subsequent iterations similar to the described example.

Enumeration of Products. In the final step of a pattern search, the resulting recombination trees are converted into the set of products containing the search pattern. A product is a fragment or a molecule constructed by connecting fragments. A recombination tree describes such a combination of fragments and holds a list of fragments in each node. Moreover, each node stores possible assignments of the SP to fragments. Since an SP may be assigned to the same fragment in different ways which would result in the enumeration of the same product, the enumeration procedure discards such assignments and removes duplicate fragments with regard to their orientation. Afterward, nodes store only unique sets of fragments. For the enumeration of products from each recombination tree, the algorithm subsequently selects one fragment from every node and connects these fragments according to the recombination tree topology. Even though the fragment sets are unique during that selection step, the combination of different fragments might still lead to the same product, for example, fragments selected from different recombination trees. In order to detect equal products and to circumvent memory limitations, the products are stored in a persistent database using unique SMILES as database keys. A single product represents the smallest fragment composition that contains the pattern of interest. Such a product may still contain open valences that allow a further attachment of fragments or that can be saturated with hydrogen atoms to form a complete molecule.

■ DATA SETS

We evaluated our search method in three use-case examples and additionally in a large scale experiment containing 738 SMARTS patterns.[45] All four experiments were conducted on three different FS's. The breaking of retrosynthetically interesting chemical substructures (BRICS)[21] follows the RECAP[19] approach and comprises 16 types of link atoms and 64 connection rules. The number of fragments varies from 4800 for BRICS 4k to 22000 for BRICS 20k. Even though both FS's contain a relatively small number of fragments, link types,

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

and connection rules, the number of products that can be created from theses spaces is arbitrarily large. A general measure for an FS is the number of products that can be created using up to five fragments. BRICS 4k and BRICS 20k contain 10^{16} and 10^{19} of such products, respectively. The KnowledgeSpace is a combination of 82 chemical synthesis protocols and contains 10876 fragments, 488 link types, and 7130 connection rules. Due to its chemical source, the KnowledgeSpace covers about 1.2×10^{10} products that are presumably synthetically accessible.

RESULTS

The following experiments reflect drug development scenarios in which a chemical pattern search is of central interest. In each experiment, FS's are searched for molecules that include a user-defined query pattern. We automatically verified that only molecules including the query were retrieved. The tests reveal insides on the algorithmic run time of our search method and show its general applicability in drug development.

The measurements are intended to reveal dependencies on the number of products contained in an FS and evaluate the search and enumeration times on a single Intel(R) Xeon(R) CPU E5630 2.53 GHz core. The recorded times only comprise the search and the enumeration but exclude preprocessing, e.g., molecule and pattern initialization, and I/O times.

Exploration of Chemical Homologues. The search for molecules that follow specific structural constraints is often applied after a molecular core or chemical class of interest has been experimentally identified. Since FS's contain many molecules that are present neither in vendor catalogs nor in in-house databases, a subsequent search can reveal novel molecules.

We designed our first use-case example to reflect such a screening and queried three different FS's for the presence of two common sulfonamides, sulfachlorpyridazine and sulfachlorpyrazine, as depicted in Figure 5. An initial exact search

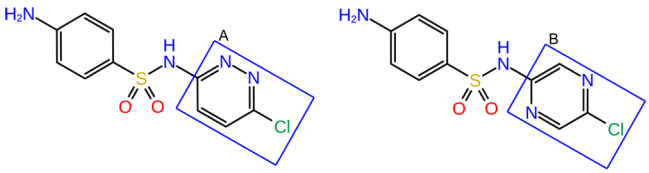


Figure 5. Structure of sulfachlorpyridazine and sulfachlorpyrazine. The ring systems used in the SMARTS query are indicated with A and B.

revealed that both molecules are not contained in either of the three spaces. Nevertheless, the FS's might encode homologue compounds of interest. Therefore, we refined the query as shown in Figure 6 to describe sulfonamides with specific ring systems as present in sulfachlorpyridazine and sulfachlorpyrazine and queried the FS's again. Table 1 summarizes the number of products obtained from the three searches. The two BRICS spaces contain a small number of such sulfonamides, all composed of two fragments joined at the bond between sulfur and nitrogen of the sulfonamide group. This connection is directly defined in the BRICS connection rules. Many of these molecules are similar to sulfachlorpyridazine and sulfachlorpyrazine. Figure 7 shows examples of the extracted molecules and products. The products still include open valences which allow for the further attachment of fragments to obtain even larger molecules and to forward them to lead optimization.

Search times ranged from 3.8 to 47.9 s and enumeration times from 0.08 to 0.7 s. With regard to the large number of possible products contained in each FS, 2.6×10^6 and 1.7×10^8 products with up to two fragments in BRICS 4k and BRICS 20K, respectively, the search times are below 1.5 ms for each product. (The number of possible products with n fragments was numerically calculated with respect to the link types and connection rules. It describes all theoretically possible combinations of up to n fragments, taking into account the connection topology when more than three fragments are connected.) Conventional database searches need around 0.04 ms per molecule,[45] which is in comparison to our search times about 27 times slower. However, a search only supplies a combination of fragments and the corresponding connection of link atoms. If explicit products are required, the relative search times per product moreover include the enumeration and then range around 1.6 ms per product. Nevertheless, an FS in combination with the described search procedure enables a fast pattern search of an arbitrarily large number of molecules in a reasonable time.

Detection of Undesired Products. Since FS's cover a large number of molecules, they might also include compounds that are inappropriate for drug development. They might have undesired physicochemical or structural properties, for example, contain reactive functional groups. Figure 8 shows a pattern that describes skin toxic molecules.[48] A pattern search revealed that both BRICS spaces in fact contain a large number of skin toxic products while the KnowledgeSpace only contains 29 such molecules (Figure 9). Table 2 shows an overview of the search results. The composition of the products with up to five fragments retrieved from the BRICS spaces is due to the generic pattern definition that constructs products involving multiple connection rules. The KnowledgeSpace is much more restricted on how fragments can be connected, and thus, the query leads to a significantly smaller number of skin toxic products. In addition, the KnowledgeSpace originates from chemical synthesis protocols in which toxic products are rather unlikely.

The search times are in accordance with the first experiment and range from 2.0 s to 1.3 min. At the first sight, the enumeration times, especially in BRICS 20k, are surprisingly high with about 6.3 days for 2.3×10^8 products. However, considering that the method enumerated over 200 million products, the average enumeration time per product is 2.3 ms. Therefore, the enumeration time is 1.4 times higher compared to the first experiment, which is a result of the additional demands on the persistent product database when millions of products need to be compared.

Fragment spaces are often used to supply new directions in a drug discovery process. Obviously, only molecules free of reactive groups are suited for lead optimization, and therefore, the number of reactive molecules should be as small as possible in a utilized FS. Our method allows a quantification of products contained in an FS's that include user-defined reactive groups. Moreover, the results in the form of recombination trees obtained from such a search reveal how such reactive products are composed. This information can be used to optimize the FS by modifying the connection rules to reduce the formation of toxic products.

Extraction of Macromolecules under Structural Constraints. Besides small molecules, FS's contain a number of large macromolecules such as oligopeptides. Oligopeptides are used as inhibitors for protein targets such as kinases,

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

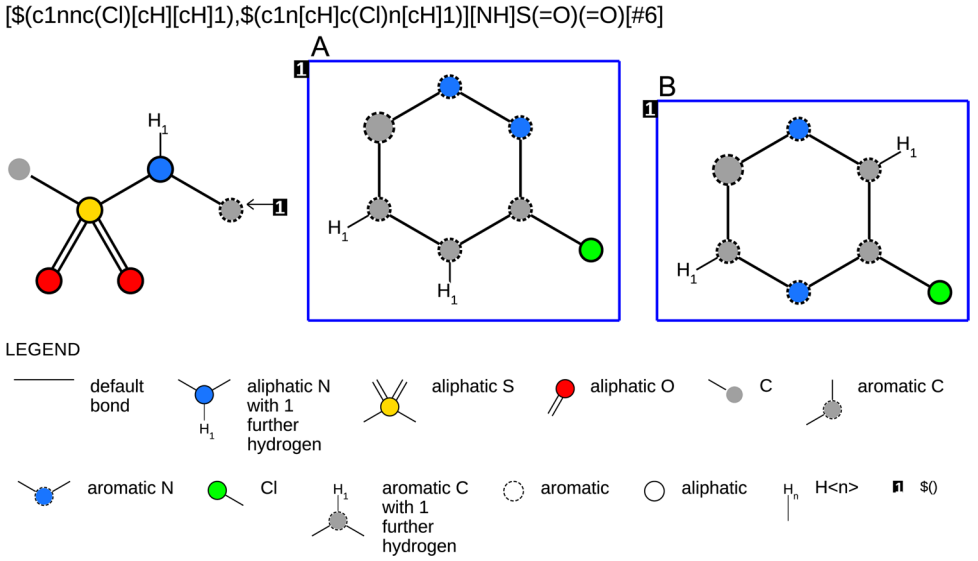


Figure 6. Sulfonamide pattern with two different ring systems defined as recursive atomic environment. The sulfachlorpyridazine ring system is marked with A and the sulfachlorpyrazine with B. The SMARTS string is given in the Supporting Information as "sulfonamides". Figure created with SmartsViewer.[47]

Table 1. Results Querying for Sulfonamides with Restricted Ring Systems

	search time (s)	enum. time (s)	products	
			1 fragment	2 fragments
BRICS 4k	3.84	0.08	0	49
BRICS 20k	47.90	0.70	0	446
KnowledgeSpace	17.81	0.00	0	0

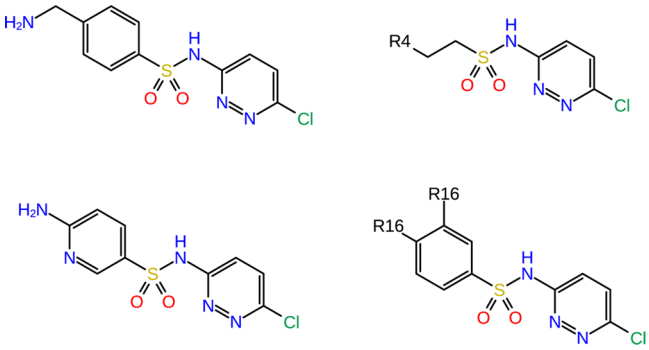


Figure 7. Examples of sulfonamide molecules (left) and products (right) retrieved from BRICS 4k (top) and BRICS 20k (bottom).

proteases, and HIV-1 assembly.[49] In order to show the ability of our method to handle large query patterns including a number of atomic environments, we searched each FS for the occurrence of tripeptides. The search was restricted to identify only tripeptides with specific hydrophobic side chains. Figure 10 depicts the associated query pattern. Table 3 and Figure 11 show that all three FS's contain such tripeptides: BRICS 4k allows the construction of 6 unique peptides. As expected from the higher number of fragments contained in BRICS 20k, the number of found tripeptides rose to 88 products. The largest number of tripeptides was identified in the KnowledgeSpace with 256 molecules. The search times ranged between 45.58 s and 19.03 min. Even though the number of retrieved products was relatively small, the method indirectly considers all possible

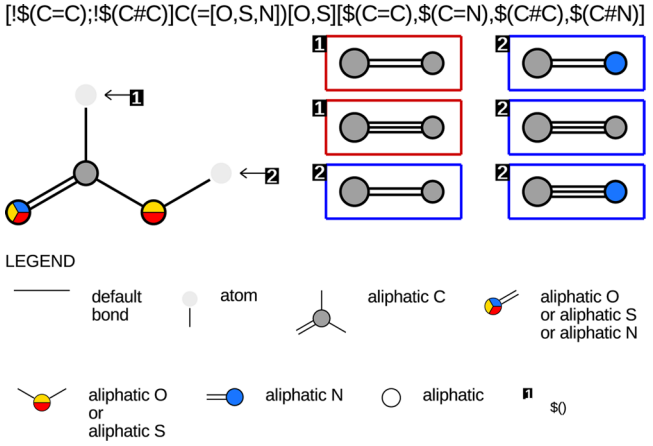


Figure 8. A SMARTS pattern describing skin toxic compounds.[48] All red atomic environments are forbidden at position 1. At least one of the blue environments must be present at position 2 in the retrieved products. The SMARTS string is given in the Supporting Information as "skin toxic". Figure created with SmartsViewer.[47]

products with up to five fragments, i.e., 2.3×10^{16} in BRICS 4k, 1.3×10^{19} in BRICS 20k, and 1.2×10^{10} products in the KnowledgeSpace. Assuming that the search would be performed in a fully enumerated space, a single product of the FS was scanned in 1.98 fs in BRICS 4k, 0.09 fs in BRICS 20k, and 82.6 ns in the KnowledgeSpace.

Run Time Statistics. Besides the selected use-case examples, we extensively evaluated the run time of our algorithm with 738 additional searches on all three FS's. The employed SMARTS patterns comprise 504 patterns without and 234 patterns with atomic environment definitions.[45] An overview of the search times in BRICS 4k, BRICS 20k, and KnowledgeSpace is presented in Figure 12. Hydrogens are implicitly contained in the patterns. The histograms show that the algorithm finished 95% of the queries without atomic environments in below 7 s in BRICS 4k, below 300 s in BRICS 20k, and below 200 s in KnowledgeSpace. The median search times for a single query ranged between 0.06 and 3.2 s and the

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

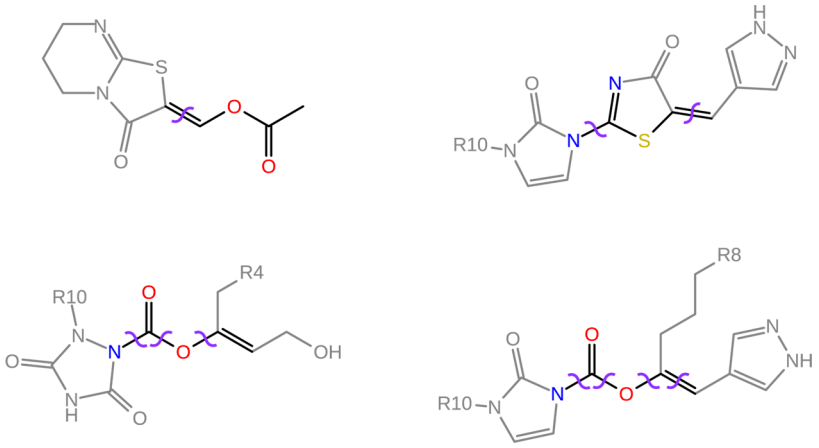


Figure 9. Examples of skin toxic molecules and products retrieved from BRICS 20k. Parts matching the pattern are in color. Former fragment boarders are indicated by purple lines.

Table 2. Results of a Query Describing Skin Toxic Compounds

			products		
	search time (s)	enum. time (h)	1 fragment	2−3 fragments	4−5 fragments
BRICS 4k	2.07	0.09	1	9.7×10^4	1.2×10^5
BRICS 20k	77.61	150.40	21	5.0×10^7	1.8×10^8
KnowledgeSpace	4.03	0.05	5	24	0

maximum between 8.7 and 27.9 min. The time needed to retrieve the desired products drastically rose when atomic environments were defined in the patterns. In these cases, 95% of the queries were finished in 10 min in BRICS 4k and in 2.8 h in BRICS 20k and 8.3 h in KnowledgeSpace. Four instances were aborted after the time limit of 48 h of computational time was reached in BRICS 20k and KnowledgeSpace. These instances were excluded from the median and maximum search time calculations. Median search times for a single query ranged between 8.3 s and 3.6 min, and the maximum ranged between 12.4 and 21.2 h. The SMARTS expressions of the patterns that exceeded the time limit are given in the Supporting Information as Smarts1, Smarts2, Smarts3, and Smarts4. A detailed inspection of Smarts1 and Smarts2 showed that these patterns are of exceptional size and include an outstanding number of atomic environments. In general, both features lead to a dramatic increase in search time. The recursive definitions in Smarts3 of the nitrogen nodes are unfortunate for the algorithm, since they result in a lot of matches that all had to be inspected. We rewrote the pattern (Smarts3′) with an explicit specification of nitrogen nodes which drastically reduced the search time. The pattern was then found in 21.5 min and in 1.2 h in BRICS 20k and KnowledgeSpace, respectively. Smarts4 defines a small symmetric pattern with equal atomic environment definitions in both nodes. The atomic environments are quite generic and lead to over a billion matches that have to be combined into a global match. As a result, the run time exceeds the limit of 2 days.

Obviously, the search times strongly differ between BRICS 4k and the other two FS's. This result is not surprising since BRICS 20k includes about five times more fragments and allows the construction of 100 times more products. All of these products had to be indirectly considered during the search. KnowledgeSpace contains fewer products than BRICS 4k but uses about 30 times more link types and over 100 times more connection rules. The results of KnowledgeSpace led to the conclusion that not only the number of possible products influences the run time but also the complexity of the FS in terms of the number of link types and connection rules.

CONCLUSIONS

The presented algorithm allows an exact search of chemical patterns in nonenumerated chemical spaces. The method supports generic queries with a recursive definition of atomic environments. These atomic environment specifications are sometimes essential for a clear definition of desired chemical patterns. The algorithm identifies the smallest combinations of fragments in an FS for which the corresponding products include the query pattern and obey the atomic environment definitions. The conducted experiments show the applicability of our method to explore chemical homologous, to sample an FS for the occurrences of toxic molecules, and to search for macromolecular structures under structural constraints. In general, large chemical spaces can be searched to identify molecules with desired structural features in reasonable times. However, the search times depend on the complexity of the patterns in terms of size and the number of atomic environments. The complexity of the searched FS with regard to the covered chemical space, the number of link types and connection rules, and the number of products that follow the query pattern has an impact on the search times, as well. In rare cases, the search times exceed a critical limit. As shown in the Supporting Information, such queries are complex and generic and include a large number of atomic environments. However, they are quite rare in practice.

At the current stage, the algorithm enforces a noncyclic connection of fragments to products which is in agreement with most FS processing algorithms and currently available FS's. Nevertheless, in special cases the formation of macromolecular cycles might be wishful. Stereoisomers are currently not addressed by our algorithm. In general, the assignment of a stereocenter is not possible during the connection of fragments,

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676−1688

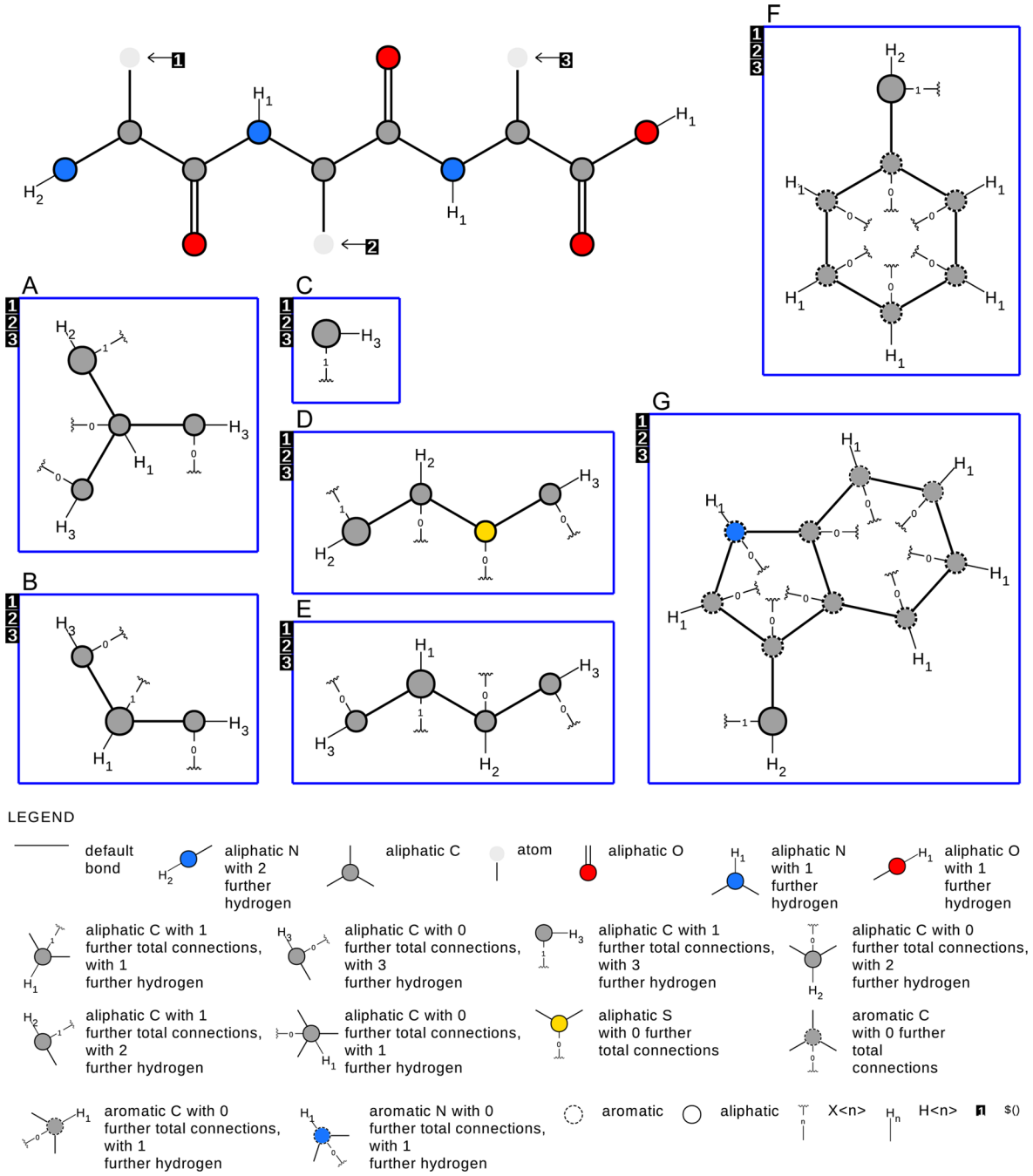


Figure 10. Tripeptide backbone pattern with the restriction to hydrophobic side chains. Atomic environment: (A) leucine; (B) valine; (C) alanine; (D) methionine; (E) isoleucine; (F) phenylalanine; (G) tryptophan. The SMARTS string is given in the Supporting Information as "oligopeptides". Figure created with SmartsViewer.[47]

Table 3. Results of Querying for Tripeptides with Hydrophobic Side Chains

	search time (m)	enum time (s)	products		
			1 fragment	2–3 fragments	4–5 fragments
BRICS 4k	0.76	0.08	0	6	0
BRICS 20k	19.03	0.16	0	50	38
KnowledgeSpace	16.51	0.34	0	256	0

since an additional fragment can create a molecular symmetry that causes stereocenters to vanish. The assignment of stereocenters is only possible on the basis of molecules. Since the search method primarily supplies products, a final correspondence between query and molecule stereocenters can only be part of a postprocessing step.

The options to further process the search results are manifold. Products retrieved from a search provide open attachment points for a follow-up lead optimization. For direct use of the products, the attachment points can be saturated with hydrogen atoms to obtain explicit molecules. Moreover, the intermediate result in the form of a set of recombination trees is a valuable source to modify the corresponding FS. The recombination trees represent the connection patterns of fragments that lead to products following the query. With this information it is possible to modify the connection rules and,

dx.doi.org/10.1021/ci400107k | J. Chem. Inf. Model. 2013, 53, 1676–1688

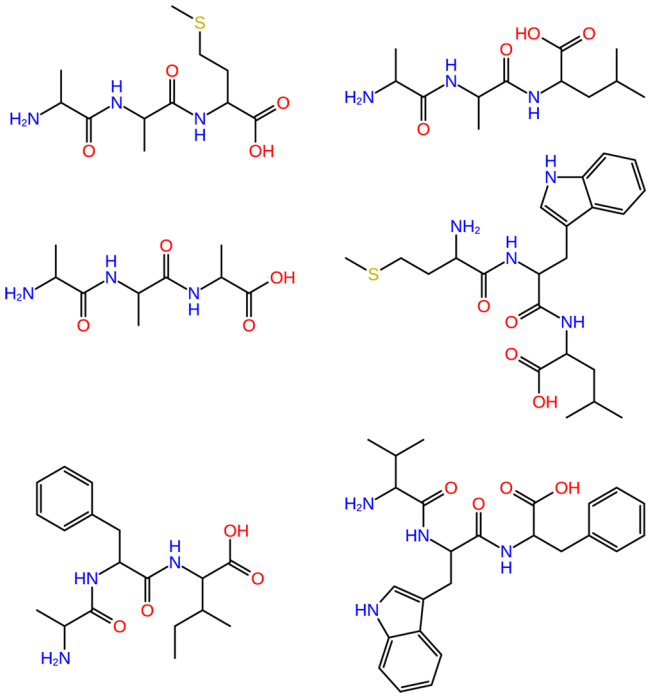


Figure 11. Examples of tripeptides retrieved from BRICS 4k (top), BRICS 20k (middle), and KnowledgeSpace (bottom).

thus, to focus or extend the underlying FS. The recombination tree might even allow the detection of common products encoded by multiple FS's.

■ ASSOCIATED CONTENT

Supporting Information

SMARTS strings referred to in the Experimental section. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

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

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Florian Lauck and Melanie Geringhoff for revising the manuscript.

■ REFERENCES

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

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

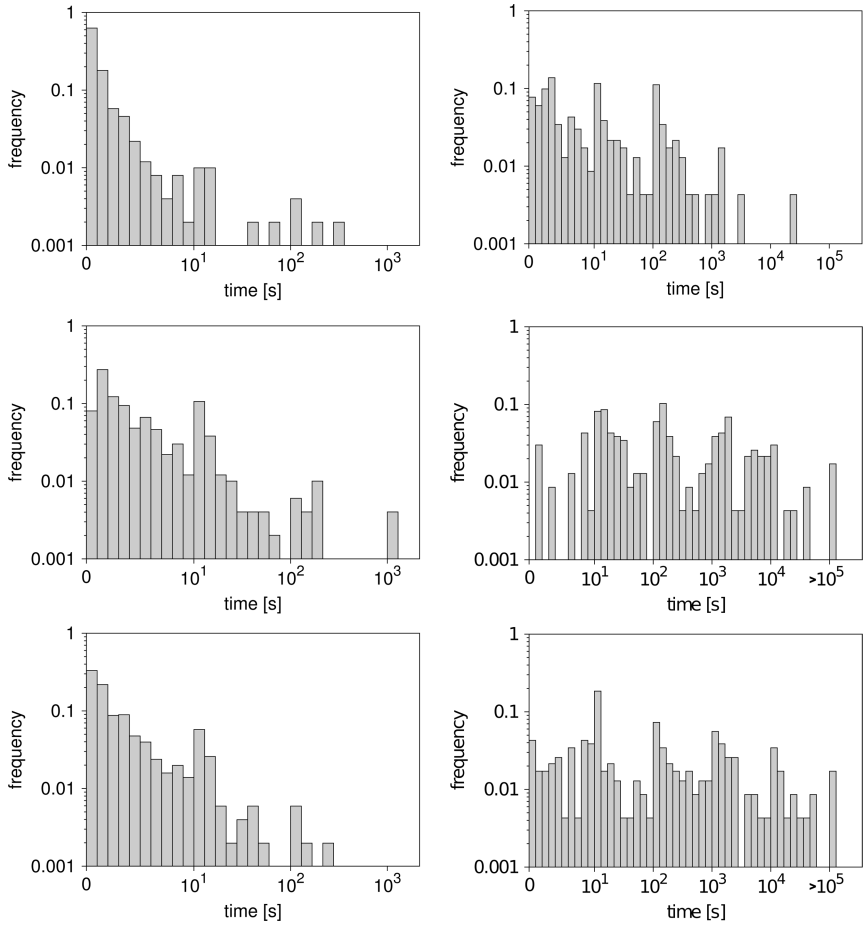


Figure 12. Search time histogram for 504 SMARTS pattern without (left) and 234 with (right) atomic environment definitions in BRICS 4k (top), BRICS 20k (middle), and KnowledgeSpace (bottom). The histogram scale is double logarithmic, the relative frequency is with regard to the number of patterns, and the times are given in seconds (s).

(3) Read, R. C.; Corneil, D. G. The graph isomorphism disease. *J. Graph Theor.* **1977**, *1*, 339−363.
(4) Gati, G. Further annotated bibliography on the isomorphism disease. *J. Graph Theor.* **1979**, *3*, 95−109.
(5) Ullmann, J. R. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31−42.
(6) Attias, R. DARC substructure search system: a new approach to chemical information. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 102−108.
(7) Heyman, J.; Karasinskia, E.; Giles, P. CAS information services for medicinal chemists. *Drug. Inf. J.* **1982**, *16*, 185−190.
(8) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Model.* **1998**, *38*, 983−996.
(9) Cordella, L.; Foggia, P.; Sansone, C.; Vento, M. Performance evaluation of the VF graph matching algorithm. *International Conference on Image Analysis and Processing, 1999. Proceedings*; IEEE Computer Society: Washington, DC, 1999; pp 1172−1177.
(10) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal.* **2004**, *26*, 1367−1372.
(11) Yan, X.; Yu, P. S.; Han, J. Substructure similarity search in graph databases. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*; ACM: New York, NY, 2005; pp 766−777.
(12) Golovin, A.; Henrick, K. Chemical Substructure Search in SQL. *J. Chem. Inf. Model.* **2009**, *49*, 22−27.
(13) Blum, L. C.; Reymond, J.-L. 970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732−8733.
(14) Irwin, J.; Shoichet, B. ZINC-a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.
(15) 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: Amsterdam, The Netherlands, 2008; Vol. 4, Chapter 12, pp 217−241.
(16) Bohacek, R. S.; McMartin, C.; Guida, W. C. The art and practice of structure-based drug design: A molecular modeling perspective. *Med. Res. Rev.* **1996**, *16*, 3−50.
(17) Fink, T.; Bruggesser, H.; Reymond, J.-L. Virtual Exploration of the Small-Molecule Chemical Universe below 160 Da. *Angew. Chem., Int. Ed. Engl.* **2005**, *44*, 1504−1508.
(18) Reymond, J.-L.; van Deursen, R.; Blum, L. C.; Ruddigkeit, L. Chemical space as a source for new drugs. *Med. Chem. Commun.* **2010**, *1*, 30−38.
(19) 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.
(20) 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.
(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) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with feature trees. *J. Chem. Inf. Model.* **2009**, *49*, 270−279.
(23) Markush, E. Patent US 1506316.
(24) Simmons, E. S. Markush structure searching over the years. *World Pat. Inf.* **2003**, *25*, 195−202.
(25) Downs, G. M.; Barnard, J. M. Chemical patent information systems. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 727−741.
(26) Cosgrove, D. A.; Green, K. M.; Leach, A. G.; Poirrette, A.; Winter, J. A system for encoding and searching Markush structures. *J. Chem. Inf. Model.* **2012**, *52*, 1936−1947.
(27) Gillet, V. J.; Downs, G. M.; Ling, A.; Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 8. Reduced chemical graphs and their applications in generic chemical structure retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126−137.
(28) 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.
(29) Holliday, J. D.; Lynch, M. F. Computer storage and retrieval of generic chemical structures in patents. 16. The Refined Search: An Algorithm for Matching Components of Generic Chemical Structures at the Atom-Bond Level. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1−7.
(30) 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.
(31) Rarey, M.; Stahl, M. Similarity searching in large combinatorial chemistry spaces. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 497−520.
(32) 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.
(33) 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 Special thanks to Neil R. Taylor for his help in preparation of the manuscript. *Angew. Chem., Int. Ed. Engl.* **2000**, *39*, 4130−4133.
(34) 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.
(35) 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.
(36) 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.
(37) 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.
(38) 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.
(39) Fischer, J.; Lessel, U.; Rarey, M. LoFT: Similarity-Driven Multiobjective Focused Library Design. *J. Chem. Inf. Model.* **2010**, *50*, 1−21.
(40) Domine, D.; Cedric, M. Method for fast substructure searching in non-enumerated chemical libraries. US Patent Application US 2007/0260583 A1, 2007
(41) Ehrlich, H.-C.; Volkamer, A.; Rarey, M. Searching for Substructures in Fragment Spaces. *J. Chem. Inf. Model.* **2012**, *52*, 3181−3189.
(42) *Daylight Theory Manual*, version 4.9; Daylight Chemical Information Systems Inc.: Aliso Viejo, CA, 2008.
(43) Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. SYBYL line notation (SLN): A versatile language for chemical structure representation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71−79.
(44) Proschak, E.; Wegner, J. K.; Schueller, A.; Schneider, G.; Fechner, U. Molecular query language (MQL)—A context-free grammar for substructure matching. *J. Chem. Inf. Model.* **2007**, *47*, 295−301.
(45) Ehrlich, H.-C.; Rarey, M. Systematic benchmark of substructure search in molecular graphs—from Ullmann to VF2. *J. Cheminf.* **2012**, DOI: 10.1186/1758-2946-4-13.
(46) Detering, C.; Claussen, H.; Gastreich, M.; Lemmen, C. KnowledgeSpace—a publicly available virtual chemistry space. *J. Chem. Inf. Model.* **2010**, *2* (Suppl 1), O9.
(47) Schomburg, K.; Ehrlich, H.-C.; Stierand, K.; Rarey, M. From structure diagrams to visual chemical patterns. *J. Chem. Inf. Model.* **2010**, *50*, 1529−1535.
(48) Enoch, S. J.; Madden, J. C.; Cronin, M. T. D. Identification of mechanisms of toxic action for skin sensitisation using a SMARTS pattern based approach. *SAR QSAR Environ. Res.* **2008**, *19*, 555−578.

(49) 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.