Journal of
CHEMICAL INFORMATION
AND MODELING

Article

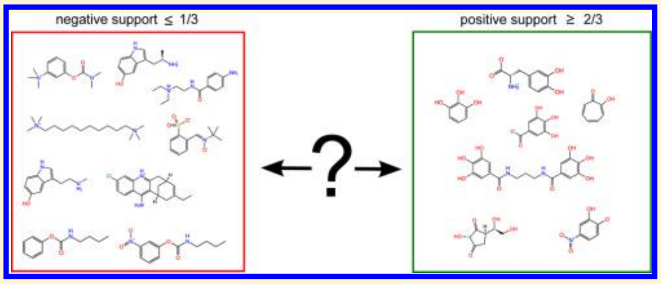
pubs.acs.org/jcim

Discriminative Chemical Patterns: Automatic and Interactive Design

Stefan Bietz, Karen T. Schomburg, Matthias Hilbig, and Matthias Rarey*

Center for Bioinformatics, University of Hamburg, Bundesstrasse 43, 20146 Hamburg, Germany

S *Supporting Information*



ABSTRACT: The classification of molecules with respect to their inhibiting, activating, or toxicological potential constitutes a central aspect in the field of cheminformatics. Often, a discriminative feature is needed to distinguish two different molecule sets. Besides physicochemical properties, substructures and chemical patterns belong to the descriptors most frequently applied for this purpose. As a commonly used example of this descriptor class, SMARTS strings represent a powerful concept for the representation and processing of abstract chemical patterns. While their usage facilitates a convenient way to apply previously derived classification rules on new molecule sets, the manual generation of useful SMARTS patterns remains a complex and time-consuming process. Here, we introduce SMARTSminer, a new algorithm for the automatic derivation of discriminative SMARTS patterns from preclassified molecule sets. Based on a specially adapted subgraph mining algorithm, SMARTSminer identifies structural features that are frequent in only one of the given molecule classes. In comparison to elemental substructures, it also supports the consideration of general and specific SMARTS features. Furthermore, SMARTSminer is integrated into an interactive pattern editor named SMARTSeditor. This allows for an intuitive visualization on the basis of the SMARTSviewer concept as well as interactive adaption and further improvement of the generated patterns. Additionally, a new molecular matching feature provides an immediate feedback on a pattern's matching behavior across the molecule sets. We demonstrate the utility of the SMARTSminer functionality and its integration into the SMARTSeditor software in several different classification scenarios.

INTRODUCTION

Molecule classification is a common problem in cheminformatics. Many applications that are related to toxicity prediction, structure−activity relationships, or substrate identification rely on the discovery of discriminative features to differentiate between two different molecule sets. The information inherent in these sets can be exploited to make predictions for molecules for which experimental information is not yet available. Finding discriminative features between the two sets can then be used to classify a new molecule as belonging to the one or the other class.

Such a discriminative feature may be of various kinds. In QSAR, mostly physicochemical properties such as molecular weight and logP or even 3D information such as shape are used.[1] While these descriptors often represent the complete molecule, a better discriminative feature might be found in a small but characteristic part of a molecule. Such cases can be assessed with topological molecular fingerprints, discriminative molecular substructures, or more abstract structural descriptors to which we will refer as discriminative chemical patterns. (For a differentiation between substructures and chemical patterns see one of our previous publications.[2]) Based on chemical structures, they are intended to explain why certain molecules from a *positive* set exhibit an effect and those from another *negative* set do not. Topological molecular fingerprints mostly describe molecules on the basis of a limited set of certain topological features[3] or predefined substructures[4] which, e.g., allows efficient storage and comparison for searching molecular databases. In contrast, molecular substructures generally facilitate the description of arbitrarily chemical moieties and are furthermore intuitive interpretable. Chemical patterns can be applied when a chemical feature cannot be sufficiently described by a molecular substructure and a more general or more specific characterization is required. Molecular pattern languages like MQL (molecular query language),[5] SLN (Sybyl line notation),[6] or SMARTS (SMILES arbitrary target specification)[7] support the formulation as well as the application of chemical patterns. While exploiting the information contained in such a discriminative chemical pattern is rather easy by using filters, its derivation is not trivial. Therefore, computational methods are required to support this task.

The automatic generation of discriminative chemical patterns is highly related to the problems of searching for frequent molecular substructures, common chemical patterns, or discriminative substructures. Based on the common computational representation of molecules, substructures, and chemical pattern as molecular graphs, these tasks are often tackled using subgraph mining algorithms. As subgraph mining constitutes a widespread technique in a multitude of application scenarios in computer science, a notable amount of solutions to this

Received: May 27, 2015
Published: August 13, 2015

DOI: 10.1021/acs.jcim.5b00323

problem has been developed.[8] Most of them are based on an iterative subgraph extension by either adding single nodes or joining previously generated subgraphs. Often, all solutions need to be identified from a highly complex search space. Hence, huge efforts have been directed at developing more efficient approaches, primarily by avoiding the detection of redundant substructures. In the field of cheminformatics, many applications benefit from these highly efficient methods. Frequent substructure mining techniques are, e.g., applied for molecule data set visualization,[9] clustering,[10,11] or efficient database searching.[12] However, a common problem of frequent substructure mining is its inherent characteristic that exact subgraph matching prevents the identification of similar properties resulting from chemically related elements. Therefore, Inokuchi proposes an approach for the creation of so-called generalized substructures, which differ from usual chemical substructures by the existence of nodes and edges describing multiple atom or bond types simultaneously.[13] The relations of all considered generalized labels are organized as a directed acyclic graph where basic atom and bond types form the leaves and all ancestors of a node constitute its alternative and more general labels. Similarly, Kazius et al.[14] use an extended chemical representation of the input molecules to enable the identification of chemical substructure patterns at different degrees of chemical detail. While usually atoms are represented by single nodes labeled with their element, this approach describes certain atoms by so-called atomic hierarchies. Using this elaborate chemical representation, the detection of frequent subgraphs rather results in chemical patterns than simple substructures.

Discriminative structural features are often discovered in a similar way. CASE[15] enumerates all linear fragments of a length between 3 and 12 atoms in an active and an inactive molecule set and evaluates their distribution across both sets. Fragments that predominantly occur in one set are classified as either activating or inactivating. MULTICASE uses the CASE descriptors in a hierarchical selection algorithm for generating a set of discriminative substructures.[16] MOLFEA supports the search for linear discriminative substructures allowing for frequency constraints for both a positive and negative molecule subset.[17] A first search identifies the set of substructures that fulfill a minimal frequency constraint with respect to the positive set. Afterward, all fragments that are too frequent in the negative set are removed during a second search.

The substructure mining tool MoSS can be used for the detection of frequent as well as discriminative substructures. The simultaneous usage of all substructure embeddings across all molecules allows for a direct detection of discriminative fragments and thus avoids the need of subsequently matching frequent substructures in the negative subset.[18] Several extensions to its first version including the usage of atomic wildcards, unified ring mining, and fuzzy chain matching have been developed.[19,20] Ting and Bailey introduce an approach for the identification of disconnected discriminative substructures based on the calculation of the maximal common edge set and a minimal hypergraph transversal algorithm.[21] LAST-PM has been developed for a fully automated derivation of generalized latent chemical patterns.[22] The underlying algorithm joins discriminative substructures by combining a common core with mutually deviating branches which results in a weighted graph. This can be further transformed into a SMARTS pattern containing alternative atom descriptors (combined by logical OR) including recursive SMARTS definitions. Compared to MoSS and the approaches by Inokuchi and Kazius et al., the automatic merge of alternative features supersedes a predefined selection of generalized descriptors. However, this also disables the user from applying custom-made descriptor sets and differentiating between chemical meaningful and rather disadvantageous generalizations. Furthermore, LAST-PM does not support the detection of cyclic patterns.

Frequent and discriminative substructures or patterns can be used as features for machine learning techniques like decision trees,[14] support vector machines,[17,22,23] or custom-made scoring schemes[24] to learn classification models for structure−activity relationships or toxicity prediction. Such applications are also closely related to the derivation of emerging patterns[25] which "are defined as itemsets whose supports increase significantly from one data set to another".[26] Originally, emerging patterns were introduced to cheminformatics by Auer and Bajorath using different molecular properties as items.[27] Following studies also applied this technique to substructure descriptors that could be, e.g., derived from fragmentation rules[28] or substructure mining.[29] As the application of such machine learning techniques allows for the detection of conjointly occurring substructures, they constitute a well-suited complement to substructure mining approaches, which often only support the detection of connected substructures. Further information on the application of graph mining in drug discovery and relevant substructure detection in the context of toxicology is given in reviews by Takigawa and Mamitsuka[30] and Lepailleur et al.[31]

Here, we present the integration of a new method for automatic discriminative molecular pattern generation into an already existing interactive pattern editor. The algorithm named SMARTSminer is based on known concepts from graph mining but tailored specifically to the applications in cheminformatics and modeling. Given two distinct molecule sets, SMARTSminer derives connected chemical patterns which can be used to differentiate between both sets. SMARTSeditor is one of a few interactive chemical pattern editors[2] based on SMARTS patterns. [SMARTSeditor and SMARTSminer are available at http://www.zbh.uni-hamburg.de/smartseditor.] Combining our intuitive visualization of chemical patterns[32] with the interactive editor and the automatic pattern generation renders these features applicable for nonexperts while covering multiple interesting use cases occurring in different molecular design scenarios.

■ METHODS

Editing Chemical Patterns. SMARTSeditor is an interactive editor for creating chemical patterns based on SMARTS. The first version, which is here extended with the SMARTSminer functionality, is described in Schomburg et al.[2] The depiction of patterns within SMARTSeditor is based on the SMARTSviewer concept.[32] The SMARTSeditor software as well as the newly developed SMARTSminer algorithm are based on the SMARTS language as defined by Daylight.[7] Apart from the consideration of disconnected SMARTS strings and SMARTS elements describing stereochemical features, SMARTSeditor supports the full variety of the SMARTS language. For molecule handling, parsing, and matching the SMARTS strings, the in-house NAOMI platform[33] is used.

Mining Chemical Patterns. Similar to most other subgraph mining techniques, SMARTSminer is based on an iterative extension of previous solutions by single elements. The usage of SMARTS strings for the representation of

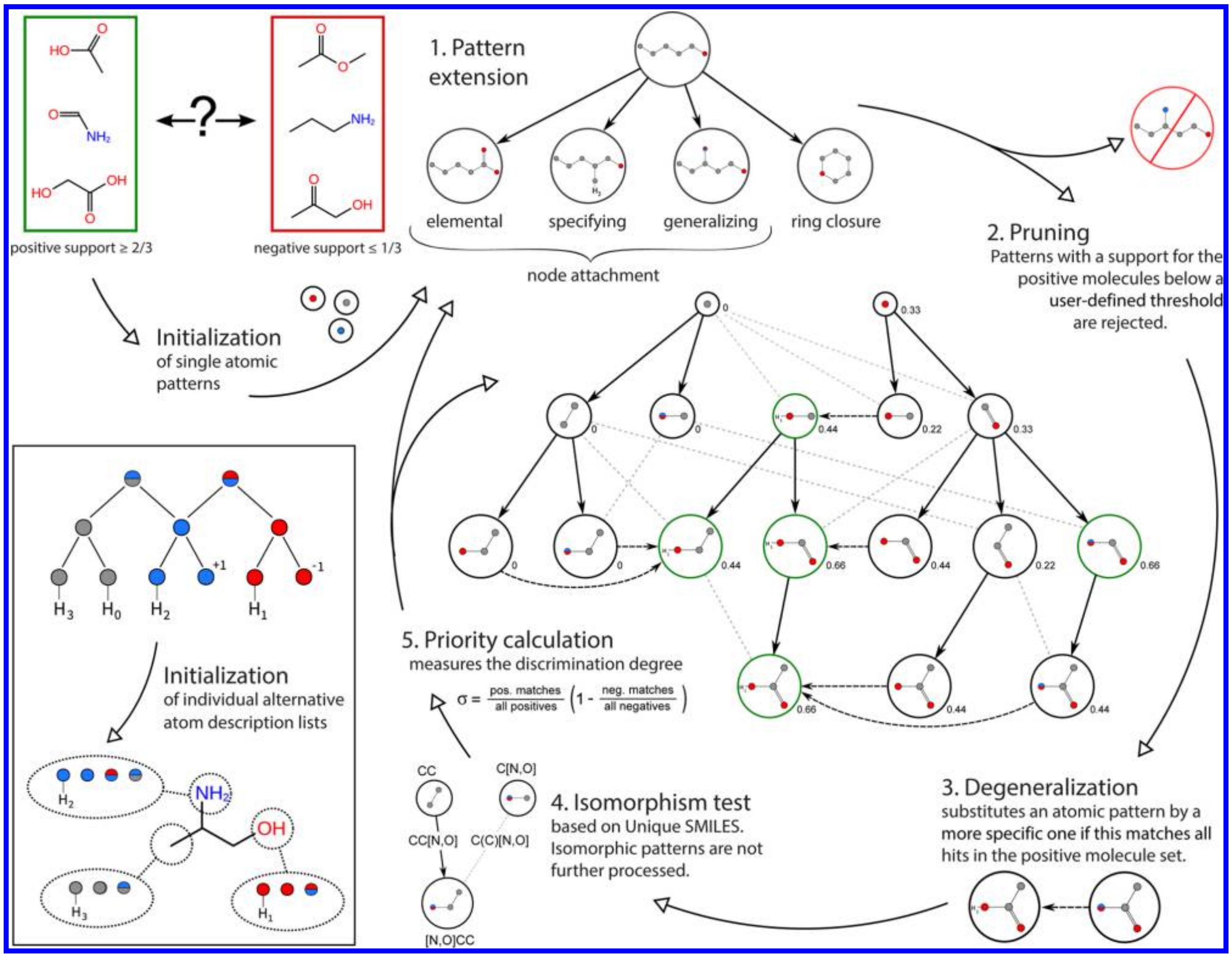

HO
O
O
NH2
HO
OH
O
positive support ≥ 2/3
←?→
O
O
NH2
O
OH
negative support ≤ 1/3
1. Pattern extension
elemental
specifying
generalizing
ring closure
node attachment
2. Pruning
Patterns with a support for the positive molecules below a user-defined threshold are rejected.
Initialization
of single atomic patterns
0
0.33
0
0
0.44
0.22
0.33
0
0
0.44
0.66
0.44
0.22
0.66
0.66
0.44
0.44
H3
H0
H2
H1
+1
-1
Initialization
of individual alternative atom description lists
H2
NH2
OH
H3
H1
5. Priority calculation
measures the discrimination degree
σ = pos. matches / all positives (1 - neg. matches / all negatives)
CC
C[N,O]
CC[N,O]
C(C)[N,O]
[N,O]CC
4. Isomorphism test
based on Unique SMILES. Isomorphic patterns are not further processed.
3. Degeneralization
substitutes an atomic pattern by a more specific one if this matches all hits in the positive molecule set.

Figure 1. Overview of the SMARTSminer algorithm. Two molecule sets together with a positive and a negative support threshold are given as input (upper left side). During the initialization phase, an individual list of alternative descriptors is prepared for each atom (lower left side). Larger patterns are generated in an iterative extension process composed of five steps (cycle at the right side). In the middle of the cycle, the resulting search tree is visualized: Each node represents a single pattern that is obtained during the recursive search. The numbers beside the nodes reflect their discrimination score σ. The solid arrows represent successfully completed recursive calls of the extension cycle, dashed arrows a degeneralization step, and dashed lines an unnecessary extension that is identified by the isomorphism test. The processing order of the nodes follows the priority search rules. Patterns that fulfill the required support values are highlighted in green.

discriminative chemical features motivates a graph-based buildup process that handles atomic descriptors as composable entities. The pattern generation starts from single atomic descriptors and adds further atom nodes in a backtracking-based search procedure. A most central aspect in the concept of the SMARTSminer algorithm constitutes the purpose of allowing for an immediate interactive examination and further processing of sensible preliminary solutions by the user. This is facilitated by a priority-driven search tree traversal which always selects the most discriminative pattern across all previously detected ones for the next extension iteration and therefore quickly identifies highly discriminative features. The discrimination behavior of a pattern is measured as the relative amount of molecules across the positive set P and the negative set N which contain the given pattern. We will refer to these values as a pattern's *positive* and *negative support* (s_P and s_N), respectively:

$$s_X(p) = \frac{|\{x \in X \mid p \sqsubseteq x\}|}{|X|} \qquad (1)$$

where X denotes either P or N and

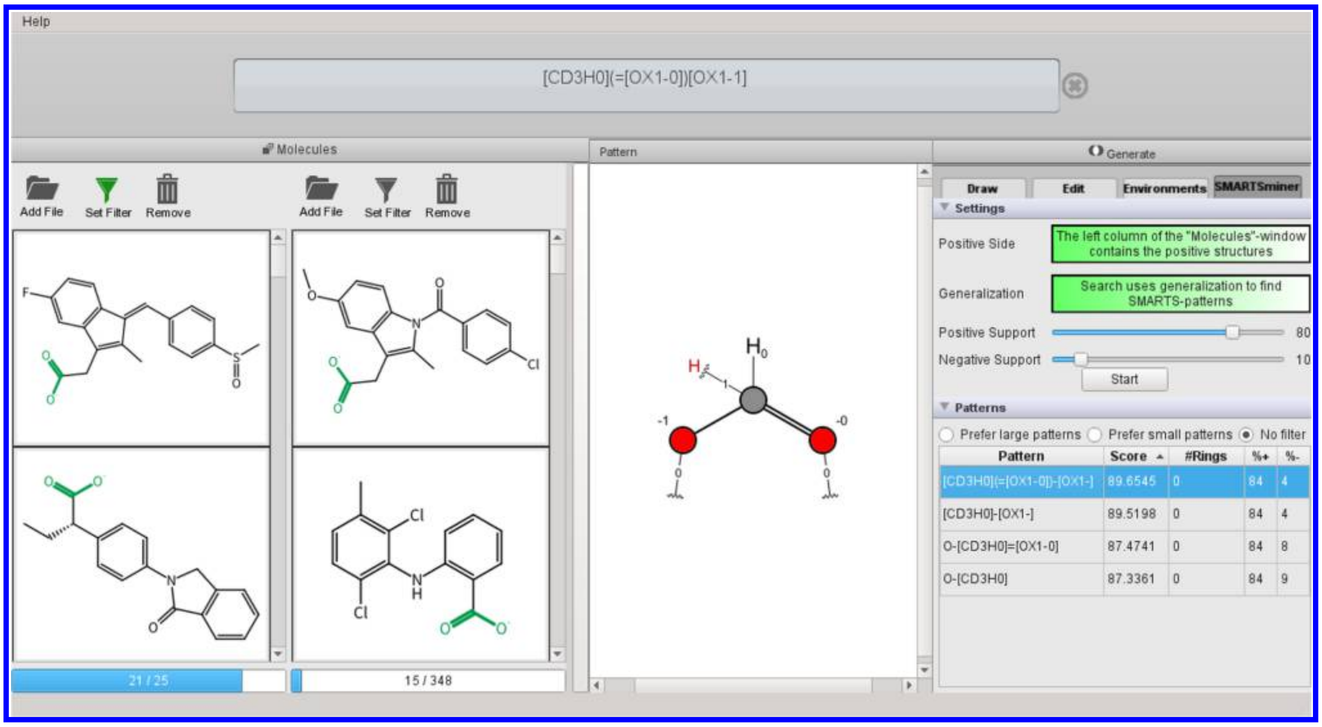
$\sqsubseteq$

an edge-induced subgraph isomorphism. Both features are further combined to a single discrimination score $\sigma(p)$:

$$\sigma(p) = \sqrt{s_P(p)\cdot(1 - s_N(p))} \qquad (2)$$

A σ-value of 1 corresponds to a perfect discrimination of both molecule sets where the respective pattern occurs in all molecules of the positive set. σ-values that are close to zero indicate either a low discriminative behavior or a low positive support. Theoretically, patterns with high negative and low positive support values also exhibit a discriminative character. SMARTSminer solely aims at discrimination patterns that are dominated by high positive support values. However, the SMARTSeditor user interface provides a convenient exchange of the molecule sets. Moreover, the user can define thresholds for a minimum required positive support and a maximum tolerated negative support. The former accelerates the pattern search by cutting search tree branches, while the latter can be used to reduce the number of valid solutions.

Figure 1 illustrates the search procedure in closer detail with the aid of a simplified example. First, a preprocessing step



[CD3H0](=[OX1-0])[OX1-1]

Settings

Positive Side: The left column of the "Molecules"-window contains the positive structures

Generalization: Search uses generalization to find SMARTS-patterns

Positive Support: 80

Negative Support: 10

Start

Patterns

○ Prefer large patterns ○ Prefer small patterns ● No filter

Pattern	Score	#Rings	%+	%-
[CD3H0](=[OX1-0])-[OX1-]	89.6545	0	84	4
[CD3H0]-[OX1-]	89.5198	0	84	4
O-[CD3H0]=[OX1-0]	87.4741	0	84	8
O-[CD3H0]	87.3361	0	84	9

Figure 2. Left set contains ligands active against COX-1 and the right set ligands active against COX-2. In the 'SMARTSminer' tab on the right side, the automatic pattern generation can be triggered. The top button allows to switch which set shall be the positive and which the negative set. The button below allows to switch on and off abstract pattern features (vs substructures containing elemental atom descriptors only). The positive and negative support sliders allow to configure the search and restrict which results are shown to the users. The results are shown in the 'Patterns'-table below the SMARTSminer options. If a row is chosen here, the result is visualized in the drawing area and the matches in the two sets are calculated and highlighted.

analyzes the frequencies of single atomic patterns in both molecule sets. For each atom, the procedure detects the chemical element and whether it is aliphatic or aromatic and combines both features to a first atom description. Moreover, it identifies all matching atomic SMARTS patterns from a predefined pattern collection which additionally exhibit a frequency on the positive molecule set that exceeds the positive support threshold. This results in an individual list of alternative descriptions for each atom considering its specific molecular environment. The lists are stored in a hash map for the subsequent calculations.

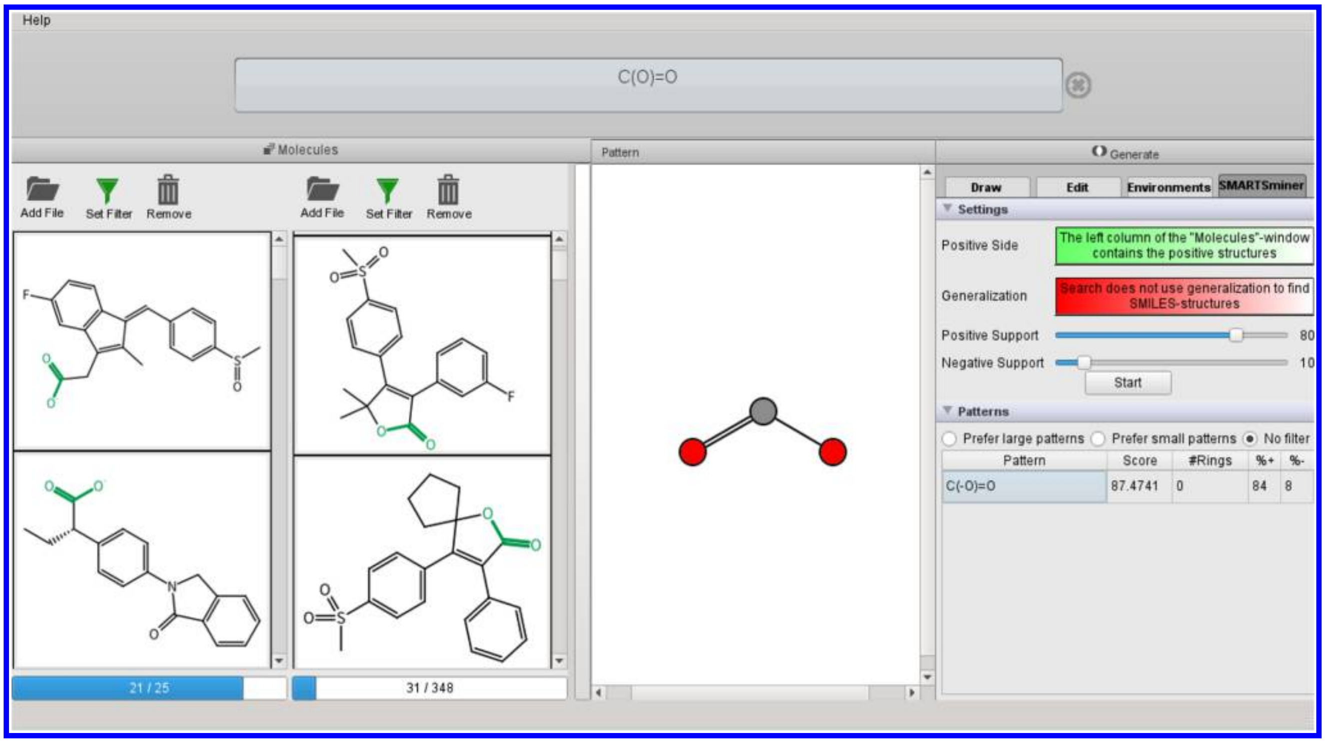
In principal, the predefined pattern collection can contain any syntactically valid SMARTS pattern describing a single atom. The only theoretical limitation is that the collection needs to be partially ordered with respect to the coverage of the patterns: In case pattern *A* matches all atom types which are also matched by pattern *B*, while *B* matches not all atom types matched by *A*, *B* is more specific and has to occur in the list before *A*. We will refer to a pattern as *specifying* if it describes a strict subset of all atom types that can be derived from an *elemental* pattern, which only characterizes an atom's chemical element and its aromaticity state. The term *generalizing* will be used for patterns which cover more than one elemental pattern. The ordering is required for a sensible pattern extension during the actual search process. In practice, a trade-off between meaningful chemical diversity and acceptable runtime is made by a manually created collection. Figure S1 (see the Supporting Information) illustrates the corresponding patterns and their dependencies. The ordered list can be obtained by generating a reverse topological sorting of the depicted graph.

During the creation of atom-specific description lists, the discrimination score σ is additionally calculated for all occurring single atomic patterns which are then processed in descending order with respect to their σ-score.

An iteration of the actual extension process is composed of five steps. First, all possible extensions for the currently highest scoring pattern from a set of extendable candidates are created. A possible extension consists either of an additional atomic node attachment including an additional bond description or an additional bond leading to a ring closure of the previous pattern. Atomic extensions can either add an elemental, a specifying, or a generalizing node to the pattern.

Second, for each of these extensions, the positive support for the resulting pattern is evaluated, and only those which exceed the user-defined threshold are maintained.

In the third step, each extended pattern is checked for a possible simplifying degeneralization: Whenever the set of underlying positive molecules of a certain pattern can also be described by a more specific pattern, the original pattern is accordingly adapted. At each pattern position, the initially stored description lists of its underlying atoms are compared, and the most specific pattern representing all underlying atoms is selected. This directly leads to the most specific representation for the set of underlying molecular substructures. The degeneralization step is essential for a feasible consideration of generalizing patterns as it eliminates those cases for which a general description yields no benefit compared to an equivalent more specific pattern. On the one hand this effectively reduces the number of search tree branches that need to be processed and thus improves the runtime behavior. On the other hand it also prevents the generation of



Positive Side: The left column of the "Molecules"-window contains the positive structures

Generalization: Search does not use generalization to find SMILES-structures

Positive Support: 80

Negative Support: 10

○ Prefer large patterns ○ Prefer small patterns ● No filter

Pattern	Score	#Rings	%+	%-
C(-O)=O	87.4741	0	84	8

Figure 3. COX-1/COX-2 discriminative substructure. SMARTSminer is started in substructure mode, as can be seen by the red button. Only one separating substructure is found, which hits 64% of the positive molecules (COX-1) and 8% of the negative molecules (COX-2).

overgeneralized patterns and brings more significant and precise patterns into focus.

In step four, a unique textual identifier is generated for every extended pattern and used for isomorphism detection against the reference set of all previously identified patterns. Duplicated patterns are removed, and new identifiers are added to the reference set. The generation of unique identifiers is based on the concept of Unique SMILES,[34] which has been extended by atomic SMARTS expressions for the given purpose. The canonization of this approach is limited to the reordering of a pattern's atomic descriptors and does not include a unification of those, as this is not necessary as long as the predefined atomic pattern collection does not include synonymous entities.

Finally, σ-scores are calculated for all remaining patterns which are then added to the candidate set. Patterns that fulfill the user-defined support values are additionally added to the solution set and instantly supplied for user interactions. The search process terminates, when no candidates are left for extension. Alternatively, it can be aborted by the user at any time.

Depending on the user-defined threshold, the amount and the similarity of the input molecules, the number of valid solutions can easily exceed a manually manageable amount. Therefore, a heuristic filtering procedure can be applied to reduce the solution set to a smaller selection of diverse patterns. It is left to the users discretion whether either smaller or larger patterns are preferred. Initially, the solutions are sorted with respect to, first, their discrimination score (decreasing), second, their number of bonds (either in- or decreasing), and, third, their number of ring closures (decreasing). Afterward, all patterns that are fully covered by at least one predecessor in the ordered solution list are removed. Any pattern B is treated as covered by another pattern A if the following condition applies

$$P(A) \supseteq P(B) \wedge N(A) \subseteq N(B) \tag{3}$$

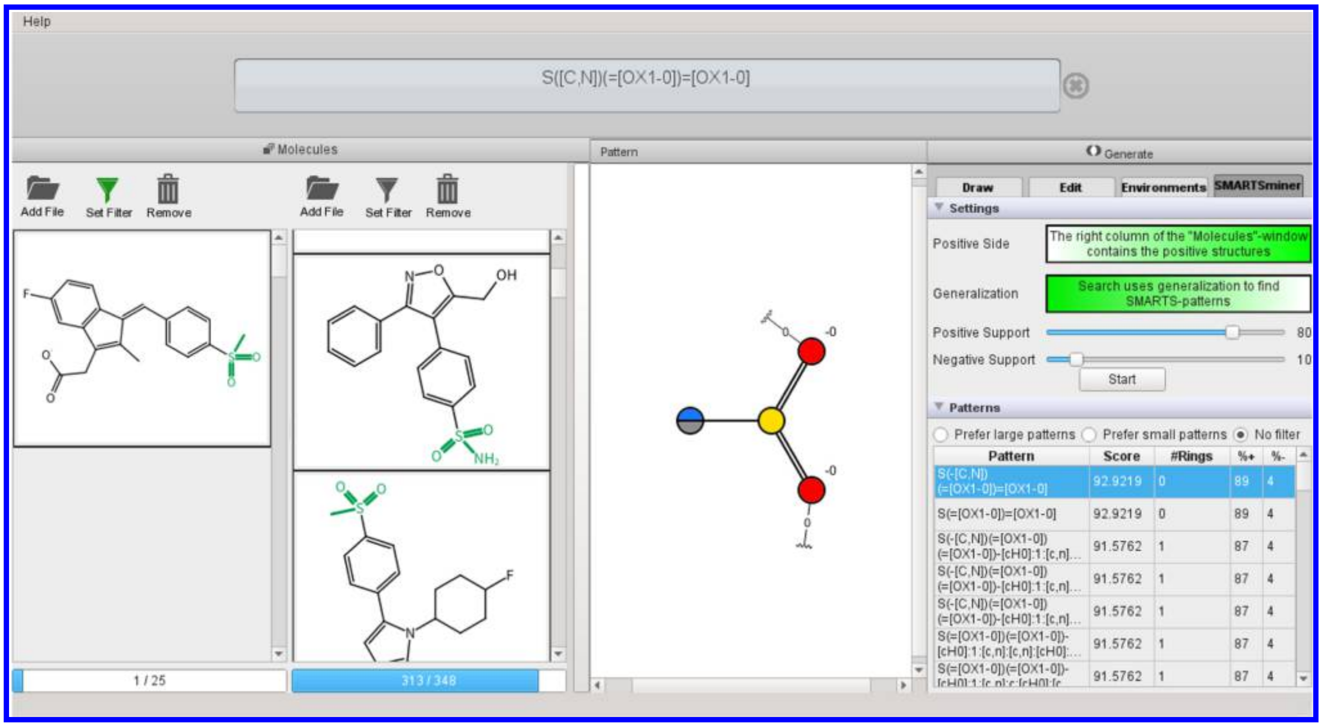
where $P(X)$ and $N(X)$ denote the sets of those molecules from the positive and the negative molecule set, respectively, which contain pattern X. Besides filtering, the user is further able to order the result list with respect to pattern size, number of rings closures, and discrimination score as well as positive and negative support.

SMARTSeditor Extensions. An important aim during the development of SMARTSminer was its smooth integration into the interactive SMARTSeditor software. For this purpose, the editor has been extended by a molecule matching feature (cf. Figure 2). The matching feature allows a user to load two molecule sets. These are depicted as 2D structure diagrams in two columns and can be browsed independently. Whenever a pattern is drawn in the editor, its matching substructures are highlighted, and the number of matching molecules across the complete set is shown. Furthermore, a filter button allows for a convenient selection of matching or nonmatching molecules from both sets.

The pattern generation functionality can be found in a separate tab named ’SMARTSminer’. This tab contains the method's parameters:

- Which set contains the molecules that the pattern/substructure shall match (positive set)
- Shall the results contain abstract pattern elements or only elemental substructures
- Up to which support thresholds shall the results be generated and displayed (minimal positive and maximal negative support)

Once a SMARTSminer search is started, all results fulfilling the parameters are displayed in a table as soon as they are found. As the search runs in a background process, the user can immediately work with already detected patterns during the search process. The results can be sorted by discrimination score δ, number of rings, or percent of matches in the positive



Positive Side: The right column of the "Molecules"-window contains the positive structures

Generalization: Search uses generalization to find SMARTS-patterns

Positive Support 80

Negative Support 10

○ Prefer large patterns ○ Prefer small patterns ● No filter

Pattern	Score	#Rings	%+	%-
S(-[C,N])(=[OX1-0])=[OX1-0]	92.9219	0	89	4
S(=[OX1-0])=[OX1-0]	92.9219	0	89	4
S(-[C,N])(=[OX1-0])(=[OX1-0])-[cH0]:1:[c,n]...	91.5762	1	87	4
S(-[C,N])(=[OX1-0])(=[OX1-0])-[cH0]:1:[c,n]...	91.5762	1	87	4
S(-[C,N])(=[OX1-0])(=[OX1-0])-[cH0]:1:[c,n]...	91.5762	1	87	4
S(=[OX1-0])(=[OX1-0])-[cH0]:1:[c,n]:[c,n]:[cH0]:...	91.5762	1	87	4
S(=[OX1-0])(=[OX1-0])-[cH0]:1:[c,n]:c:[cH0]:[c	91.5762	1	87	4

Figure 4. COX-2/COX-1 discrimination renders two patterns of different size with the same σ-score. Both hit 89% of the COX-2 pattern and only 4%, i.e., one of the COX-1 ligands. The filtering functionality enables a user to select the larger or the smaller pattern.

or negative molecule set. The result list can also be shortened by selecting either large or small patterns, as described above.

Once a result was chosen, it is loaded into the SMARTSeditor drawing section. Here it can be further modified, extended, and interactively adapted with the full SMARTSeditor functionality. The matching information in both sets is instantly updated as soon as the pattern is modified.

RESULTS AND DISCUSSION

In the following section, some use cases of the SMARTSminer functionality integrated within the SMARTSeditor software are illustrated on several examples. For most of the examples the DUD data set is used.[35] The data set has some known obstacles such as not clearly determined decoys.[36] For demonstrating the SMARTSminer functionality, this plays only a marginal role. In the following, we consider the classification of molecules into actives and decoys as ground truth.

Use Case 1: Active Molecules of Similar Targets. In drug design projects, selectivity on the one hand and polypharmacology on the other hand play a significant role. Finding molecules with selective binding profiles can be difficult if the targets are rather similar, such as belonging to the same enzyme family. One example are the isoenzymes cyclooxygenase-1 (COX-1) and cyclooxygenase-2 (COX-2), which are both prostaglandin-endoperoxide synthases. For the design of selective drugs, it might be helpful to find discriminative features.

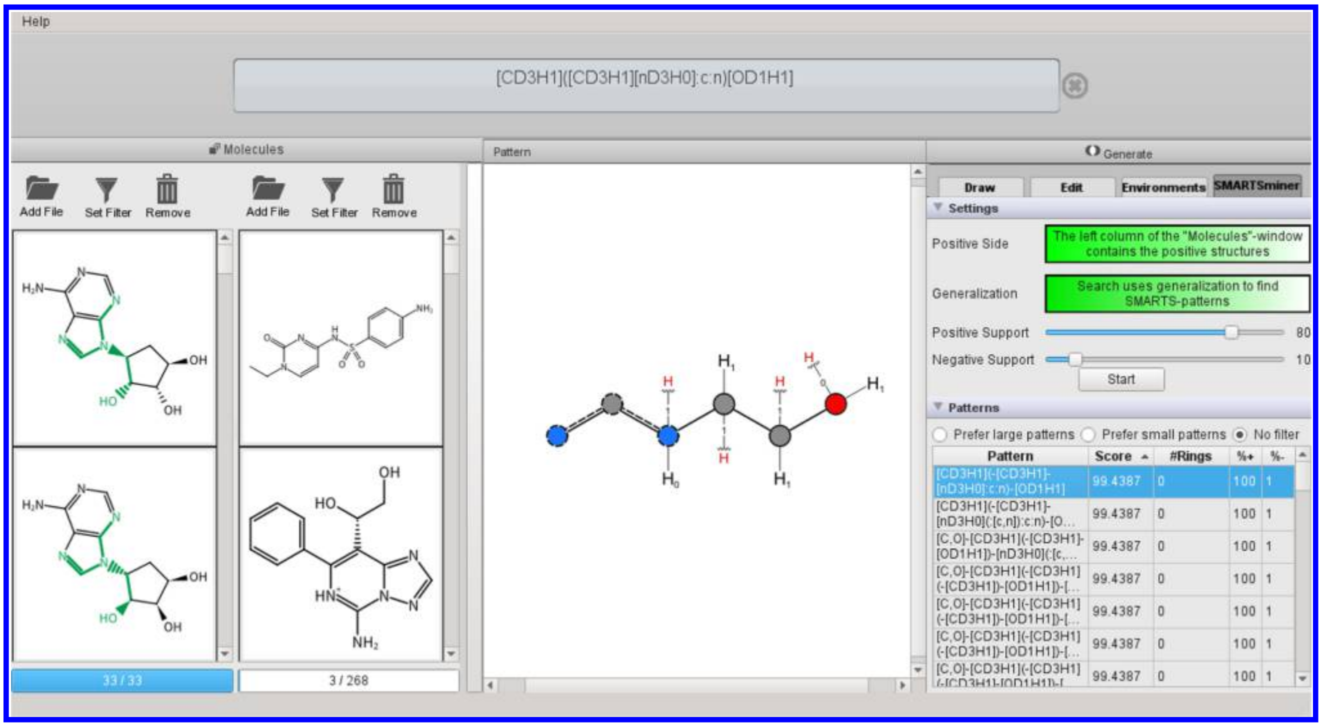
In Figure 2, the left molecule set contains ligands active against COX-1,[35] and the right molecule set contains ligands active against COX-2.[35] The automatic pattern generation identifies a carboxyl-group as a discriminative pattern: 84% of the COX-1 ligands contain it and only 4% of the COX-2 ligands. This result with the best score is found and displayed after 2 s in the table, while the exhaustive search of all possible solutions takes about 18 min (measured on a Suse 13.1 desktop with 4 Intel(R) Core(TM) i5-3570 CPU@3.4 GHz processors).

The matching parts are highlighted in the molecule sets. In this example, the set of COX-1 ligands is rather small with 25 ligands, and manual inspection would perhaps also lead a user to find the common carboxyl-group. However, it would be rather cumbersome to search the COX-2 set with 348 ligands manually.

Since the two sets from the DUD overlap in two molecules (identified with MONA[37]), a perfect separation is not possible in this case. Thus, it may be a good idea to further study experimentally if this carboxyl-group indeed is a feature which hinders ligands from binding to COX-2. The generated pattern for the carboxyl-group contains SMARTS expressions for each atom, e.g., the carbon atoms are described with the SMARTS "[CD3H0]" and one of the oxygen atoms with "[OX1-1]", showing that SMARTSminer selects pattern that are as specialized as possible.

In Figure 3, the pattern generalization is turned off so that only substructures can be found. Now, the simple substructure "C(-O) = O" is found - which still hits the same amount of ligands in the positive set but more (31 instead of 16) ligands from the negative set. The reason can be seen in the highlighting of the matches in the molecules of the negative set: Instead of specifically matching carboxyl-groups as the pattern does, the substructure also hits furanone substructures.

For now, the positive set contained the COX-1 ligands. By simply swapping the positive and the negative set, one can find a pattern for the COX-2 ligands which does not match the COX-1 ligands. In Figure 4, the results are shown. Two patterns with the same σ-score are found, a larger one, which is shown in Figure 4 loaded into the SMARTSeditor drawing area and a smaller one which contains only the sulfur and the two oxygen atoms of the larger pattern.



```
[CD3H1]([CD3H1][nD3H0]:c:n)[OD1H1]
```

Settings

Positive Side: The left column of the "Molecules"-window contains the positive structures

Generalization: Search uses generalization to find SMARTS-patterns

Positive Support: 80

Negative Support: 10

Patterns

○ Prefer large patterns ○ Prefer small patterns ● No filter

Pattern	Score	#Rings	%+	%-
[CD3H1](-[CD3H1]-[nD3H0]:c:n)-[OD1H1]	99.4387	0	100	1
[CD3H1](-[CD3H1]-[nD3H0](:[c,n]):c:n)-[O...	99.4387	0	100	1
[C,O]-[CD3H1](-[CD3H1]-[OD1H1])-[nD3H0](:[c,...	99.4387	0	100	1
[C,O]-[CD3H1](-[CD3H1](-[CD3H1])-[OD1H1])-[...	99.4387	0	100	1
[C,O]-[CD3H1](-[CD3H1](-[CD3H1])-[OD1H1])-[...	99.4387	0	100	1
[C,O]-[CD3H1](-[CD3H1](-[CD3H1])-[OD1H1])-[...	99.4387	0	100	1
[C,O]-[CD3H1](-[CD3H1](-[CD3H1]-[OD1H1]-[	99.4387	0	100	1

Figure 5. Separation of SAHH ligands versus SAHH decoys. The generated discriminative pattern is a part of the main scaffold of the actives. It hits all actives and only 3 of the decoys.

Use Case 2a: Separating Actives from Structurally Related Inactives Automatically. Another use case is the separation of molecules showing activity at a protein target from those which show none. By generating a discriminative pattern, molecules that are not tested so far can be predicted as active or inactive. In Figure 5, the left, the positive set contains actives for the *S*-adenosyl-homocysteine hydrolase (SAHH) target from the DUD. The right column contains the decoys. Here, SMARTSminer finds a pattern which hits all ligands and only 1%, i.e., 3 of the 268 decoys. The pattern covers part of the main scaffold of the actives with a part of the aromatic ring and a part of the five-membered ring. As can be seen in the Poseview[38] depiction of a SAHH-active site complexed with a ligand in Figure 6, the pattern contains parts of the interaction network hydrogen bonding partners of the ligands.

Use Case 2b: Separating Actives from Inactives Interactively. If the pattern generated by SMARTSminer does not separate the positive and the negative set completely, as can be observed in some of the cases of the DUD, the user can add more features to the pattern interactively. For demonstrating this, the actives and decoys of the target purine nucleoside phosphorylase (PNP) from the DUD are used. In Figure 7, the result of the automatic pattern generation is shown: 100% of the actives are matched by the pattern but also still 3% (25 molecules) of the decoys.

Now, the user can further extend the pattern interactively in order to eliminate the matching molecules of the negative set. Since SMARTSeditor supports almost the complete SMARTS language, the user can interactively add features which are not yet supported by SMARTSminer like recursions or bond query features. By adding two negated recursions to the pattern and a substituent of the six-membered ring connected by 'any' bond (see Figure 8), the pattern matches none of the negative set any more, thus providing a perfect separation of negative and positive molecules. The resulting pattern (SMARTS: [c,n;!$([c,n]S)]:1:[c,n]:[cH0]:2:[cH0](:[c,n;H1]:1):c(:n:[c;!$(c-, = [S,NH1])]:[nX2-0]:2) [O,NH2]) seems rather complex. However, by starting from the automatic pattern and adding features interactively using the SMARTSeditor functionality,

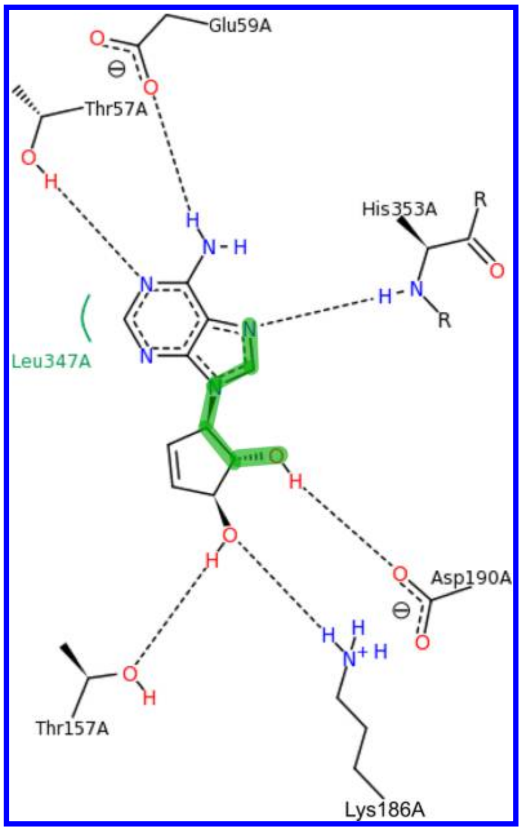
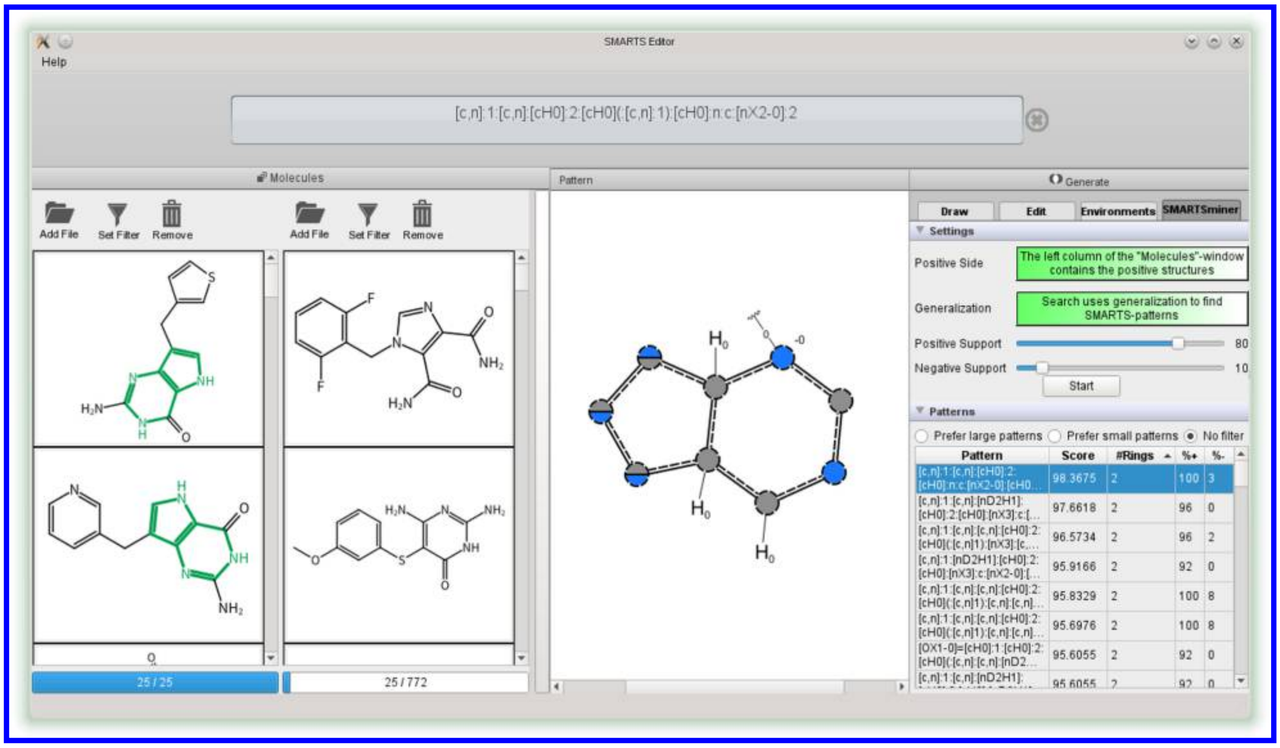


Figure 6. Poseview[38] depiction of the active site of PDB code 1a7a, SAHH complexed with an imidazopyrimidine ligand.



```
[c,n]:1:[c,n]:[cH0]:2:[cH0](:[c,n]:1):[cH0]:n:c:[nX2-0]:2
```

Positive Side: The left column of the "Molecules"-window contains the positive structures

Generalization: Search uses generalization to find SMARTS-patterns

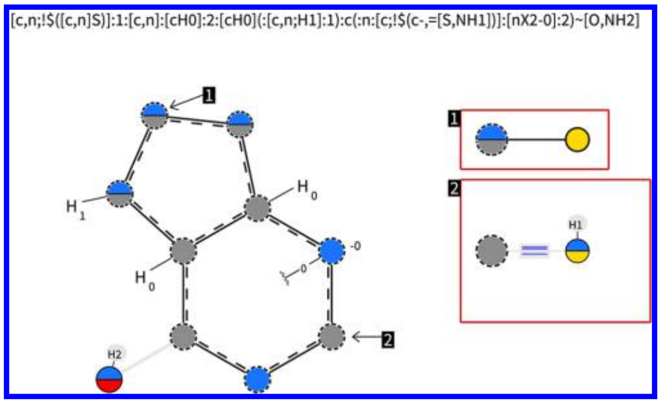
Positive Support: 80

Negative Support: 10

○ Prefer large patterns ○ Prefer small patterns ● No filter

Pattern	Score	#Rings	%+	%-
[c,n]:1:[c,n]:[cH0]:2: [cH0]:n:c:[nX2-0]:[cH0...	98.3675	2	100	3
[c,n]:1:[c,n]:[nD2H1]: [cH0]:2:[cH0]:[nX3]:c:[...	97.6618	2	96	0
[c,n]:1:[c,n]:[c,n]:[cH0]:2: [cH0](:[c,n]1):[nX3]:[c,...	96.5734	2	96	2
[c,n]:1:[nD2H1]:[cH0]:2: [cH0]:[nX3]:c:[nX2-0]:[...	95.9166	2	92	0
[c,n]:1:[c,n]:[c,n]:[cH0]:2: [cH0](:[c,n]1):[c,n]:[c,n]...	95.8329	2	100	8
[c,n]:1:[c,n]:[c,n]:[cH0]:2: [cH0](:[c,n]1):[c,n]:[c,n]...	95.6976	2	100	8
[OX1-0]=[cH0]:1:[cH0]:2: [cH0](:[c,n]:[c,n]:[nD2...	95.6055	2	92	0
[c,n]:1:[c,n]:[nD2H1]:	95.6055	2	92	0

Figure 7. Separation of PNP ligands versus PNP decoys. The automatic pattern generation finds a discrimination of 100% hits in the positive set and only 3% in the negative set.



```
[c,n;!$([c,n]S)]:1:[c,n]:[cH0]:2:[cH0](:[c,n;H1]:1):c(:n:[c;!$(c-,=[S,NH1])]:[nX2-0]:2)~[O,NH2]
```

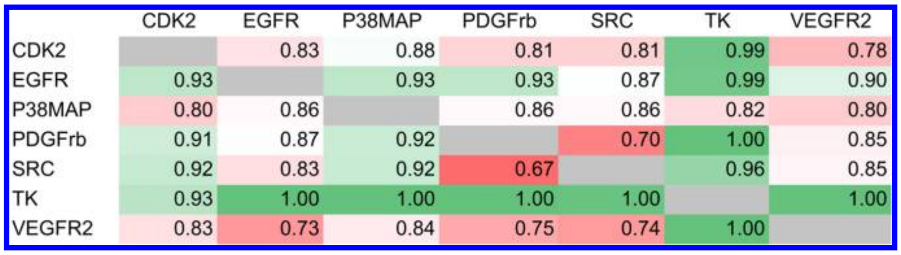
Figure 8. Pattern separating PNP actives and decoys perfectly. It contains two recursions and one substituent connected by 'any' bond, which are features not included in the automatic pattern generation. Visualization created with SMARTSviewer.[32]

even a less experienced user can easily create such complex patterns. The SMARTSeditor user concurrently sees which change induces which matching result in the molecule set and can therefore increase the selectivity of a pattern step by step.

Use Case 3: Selectivity of Kinase Ligands. The DUD data set contains ligands for several kinase targets: cyclin-dependent kinase (CDK2), epidermal growth factor (EGFR), P38 mitogen activated protein (P38 MAP), platelet derived growth factor receptor kinase (PDGFrb), tyrosine kinase SRC (SRC), thymidine kinase (TK), vascular endothelial growth factor (VEGFr2). Many kinase inhibitors are highly promiscuous. Finding separating patterns and analyzing how well the ligands of the different kinases can be classified can support the design of selective inhibitors.

Figure 9 shows a heatmap containing discrimination scores for the separation of various sets of active kinase ligands against each other. Cases for which the automatically generated pattern is able to separate the two sets perfectly are colored in dark green. A rather unsuccessful separation is marked in dark red, as shown in the example of SRC vs PDGFrb ligands.

A poor σ-score can have two reasons: Either the pattern does not cover a large part of the molecules of the positive set or it also hits parts of the negative set. In the SRC/PDGFrb example, a combination of both is the reason: only 63% of the



	CDK2	EGFR	P38MAP	PDGFrb	SRC	TK	VEGFR2
CDK2		0.83	0.88	0.81	0.81	0.99	0.78
EGFR	0.93		0.93	0.93	0.87	0.99	0.90
P38MAP	0.80	0.86		0.86	0.86	0.82	0.80
PDGFrb	0.91	0.87	0.92		0.70	1.00	0.85
SRC	0.92	0.83	0.92	0.67		0.96	0.85
TK	0.93	1.00	1.00	1.00	1.00		1.00
VEGFR2	0.83	0.73	0.84	0.75	0.74	1.00	

Figure 9. Kinase selectivity study. The table shows a heatmap and the σ-scores of the discriminative patterns which can be obtained automatically by SMARTSminer. Rows denote the positive set and columns the negative set.

positive set and 29% of the negative set are covered. In the Supporting Information, the set coverage for all kinase cases can be found. According to Drugbank[39] SRC and PDGFrb share at least one ligand (Dasatinib DB01254). An analysis of duplicates in the active ligand files of the DUD shows that SRC and PDGFrb share 55 actives. Therefore, the deficient separation of both sets is not surprising. This analysis further reveals that SRC and EGFR share 40 identical active ligands and EGFR and PDGFrb share 18 ligands. Thus, a perfect separation of these sets is not possible.

In the case of thymidine kinase (TK), the actives can be separated almost perfectly from the other kinase ligands in both directions, either if these actives are the positives or the negatives. In Figure 10, the automatically generated patterns are shown. The annotation of which pattern separates which kinase target shows that for separation against EGFR, P38MAP, and PDGFrb, the same pattern is found: A pattern of the nitrogenous base of the thymidine kinase inhibitors. For the separation against SRC kinase inhibitors, a small pattern consisting of a hydroxyl group connected to a short aliphatic chain is sufficient enough. For VEGFR, a part of the aromatic system of the nitrogenous base is coded into the pattern. For separation against CDK2 actives, a rather large pattern consisting of a combination of the SRC and VEGFR patterns is generated. This pattern is the only one which does not cover all positives (86%).

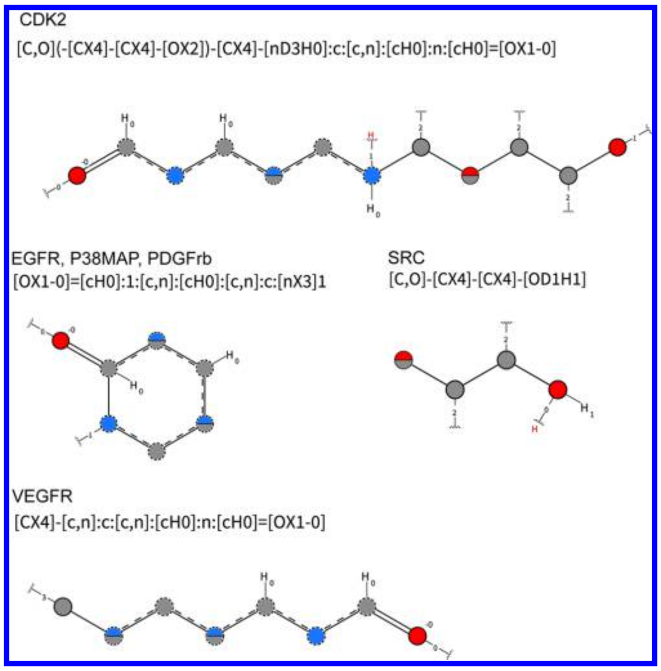

CDK2
[C,O](-[CX4]-[CX4]-[OX2])-[CX4]-[nD3H0]:c:[c,n]:[cH0]:n:[cH0]=[OX1-0]
EGFR, P38MAP, PDGFrb
[OX1-0]=[cH0]:1:[c,n]:[cH0]:[c,n]:c:[nX3]1
SRC
[C,O]-[CX4]-[CX4]-[OD1H1]
VEGFR
[CX4]-[c,n]:c:[c,n]:[cH0]:n:[cH0]=[OX1-0]

Figure 10. Discriminative patterns for separating thymidine kinase actives against CDK2, EGFR, P38MAP, PDGFrb, SRC, and VEGFR actives. Visualization created with SMARTSviewer.[32]

If an inhibitor of a special kinase shall be designed selectively, it may be of interest to separate one kinase inhibitor set against actives of all other kinase classes. The results of the separation of one kinase active molecule sets against the actives of all other kinase sets is shown in Figure 11. The patterns are rather diverse, and the score ranges from 0.7 for VEGFR to 1 for TK. The pattern generated for P38MAP is the largest with two rings.

Use Case 4: Frequent Substructure versus Frequent Pattern. Identification of a common scaffold of molecules is of interest in many applications and can help with the classification of a set of molecules. With our search strategy, a frequent substructure of a set of molecules can be identified by omitting pattern features and searching for subgraphs of a positive set only. In the case of the DUD ligands for thymidine kinase, a pyrimidine substructure is found (see Figure 12A). However, by using the pattern feature functionality, also frequent patterns can be searched - at least to the extent to which the algorithm is able to use pattern features. Since the pattern features allow a more general description of structures, the frequent patterns can be larger, i.e. covering larger parts of the molecules than frequent substructures, which in some cases allows to better describe a common scaffold of a set of molecules. In the case of thymidine kinase, indeed, SMARTSminer detects a pattern matching larger substructures, which is shown in Figure 12B. Note that six aromatic atoms are part of the pattern; however, the potential ring is not closed in order to accommodate ring systems with five- and six-membered rings.

Use Case 5: Analysis of Benchmark Data Sets. Besides the application in molecular design, it becomes clear that SMARTSminer is also very helpful in creating and analyzing validation data sets like DUD. Test cases in which actives and decoys can be separated by a relatively small pattern to a high degree are, for example, less suited for validating topology-driven similarity measures.

One example within the DUD which was especially interesting is the example of the mineralo-corticoid receptor. Here the simple pattern '[N,n]', i.e. any nitrogen atom, is sufficient to separate actives and decoys with a σ-score of 0.86:413 of 520 decoys contain a nitrogen but only one of the actives. Similar cases can be observed: In the case of SAHH *S*-adenosyl-homocysteine hydrolase (SAHH) actives and decoys, the simple pattern 'A-N' hits 0 of the actives but 223 of the 268 decoys, thus has a σ-score of 0.91.

Analyzing all molecule sets of the 40 DUD targets with respect to the smallest pattern that separates the actives from the decoys led to the following results: Using a threshold combination of 80/20 (80% hits in the ligand set and 20% hits in the decoy set), for one target a single-atomic pattern, for 17 targets a pattern of two atoms (one bond), and for eight targets a pattern of three atoms (two bonds) is found. For ten targets, a pattern of three to five bonds is found, and only four target data sets cannot be separated at this threshold setting. Using a 90/10 threshold combination, still for six targets a pattern containing two atoms is found, for five targets a pattern of three atoms and for 13 targets patterns of three to six bonds are found. A list containing the shortest patterns for all DUD targets is given in the Supporting Information.

Thus, for three-quarters of this data set (for threshold 80/20), a pattern as long as three bonds, i.e. four atoms (if no ring is formed) is sufficient for classification. Therefore, the DUD data set as being developed for docking benchmark studies should probably not be used for topological fingerprint measures. Certainly, the data set is inappropriate for all approaches applying machine learning on topological features.

Use Case 6: Characterization of Chemical Reaction Centers. As a final use case, we want to discuss the possibilities to describe reaction centers with discriminative patterns. Given a set of molecules which are subject to a certain chemical reaction and another set of molecules which are not, a discriminative pattern can be used to describe the structure of

DOI: 10.1021/acs.jcim.5b00323
J. Chem. Inf. Model. 2015, 55, 1535−1546

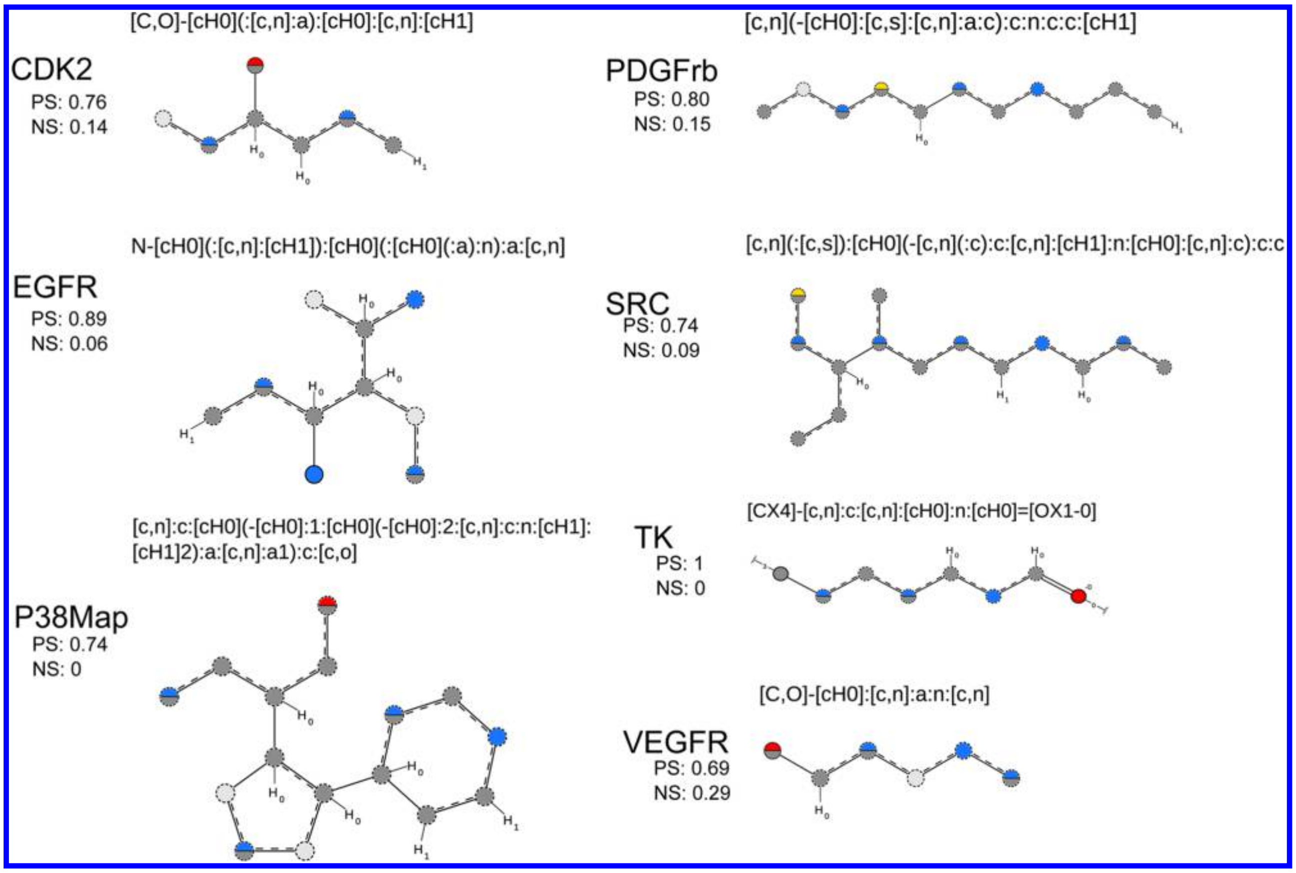

[C,O]-[cH0](:[c,n]:a):[cH0]:[c,n]:[cH1]
CDK2
PS: 0.76
NS: 0.14
[c,n](-[cH0]:[c,s]:[c,n]:a:c):c:n:c:c:[cH1]
PDGFrb
PS: 0.80
NS: 0.15
N-[cH0](:[c,n]:[cH1]):[cH0](:[cH0](:a):n):a:[c,n]
EGFR
PS: 0.89
NS: 0.06
[c,n](:[c,s]):[cH0](-[c,n](:c):c:[c,n]:[cH1]:n:[cH0]:[c,n]:c):c:c
SRC
PS: 0.74
NS: 0.09
[c,n]:c:[cH0](-[cH0]:1:[cH0](-[cH0]:2:[c,n]:c:n:[cH1]:[cH1]2):a:[c,n]:a1):c:[c,o]
P38Map
PS: 0.74
NS: 0
[CX4]-[c,n]:c:[c,n]:[cH0]:n:[cH0]=[OX1-0]
TK
PS: 1
NS: 0
[C,O]-[cH0]:[c,n]:a:n:[c,n]
VEGFR
PS: 0.69
NS: 0.29

Figure 11. Pattern separating all kinase ligands against all other kinase actives of the DUD. The SMARTS string, the SMARTS visualization, the positive support (PS), and the negative support (NS) are shown. Visualization created with SMARTSviewer.[32]

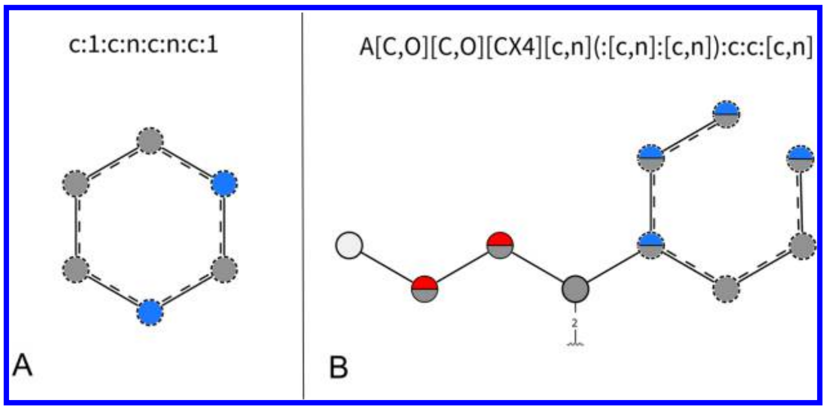


Figure 12. A) MCS of the thymidine kinase actives as designated by the search functionality using only subgraph features. B) Most common pattern of the thymidine kinase actives as designated by the search functionality using pattern features. Visualization created with SMARTSviewer.[32]

the reaction center. Enoch et al.[40] exemplary designed SMARTS patterns for different reaction mechanisms which allow for filtering molecules with skin sensitization potential. While Enoch et al. designed the SMARTS pattern with chemical expert knowledge manually, one could also use the automatic discriminative pattern generation. Clearly, the manually designed patterns contain more chemical knowledge. However, their manual design is cumbersome which is probably the reason why there are yet rather few chemical patterns describing reaction centers.

As an example, we took the molecules described in Enoch et al. as being subject to Schiff base reactions as the positive set and those molecules which are not leading to skin sensitization as the negative set. The result of the automatic pattern generation is shown in Figure 13. This pattern hits 28 of the 29 positives and seven of 31 negatives. By interactively adding one negated recursion in order to exclude carbonyl group hits, the pattern hits the same amount of positives but only one of the negatives.

Enoch et al. propose 8 different patterns for Schiff base reactions. Next to reactive oxygens, these patterns also contain reactive sulfur centers. Since the molecules of the set do not contain sulfur atoms, however, the SMARTSminer patterns do not contain any either. Thus, here chemical knowledge applied by Enoch et al. adds meaningful content to the patterns which cannot be found by the automatic algorithm.

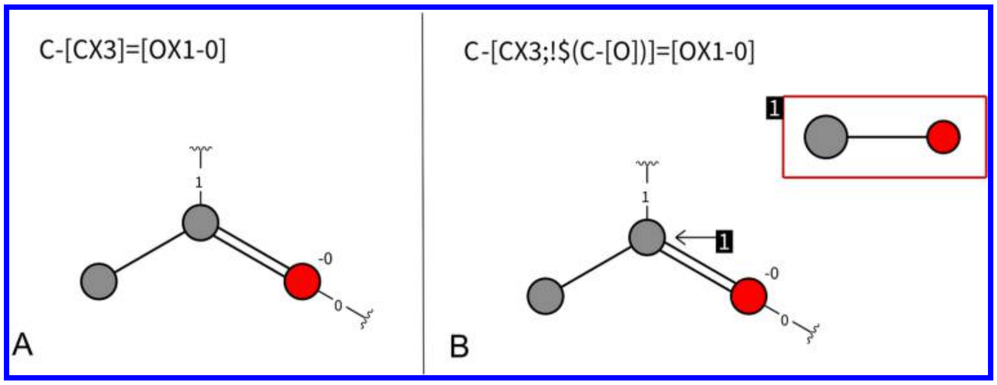


Figure 13. Pattern describing a Schiff base reaction center as occurring in the molecules listed in Enoch et al.[40] A) Automatically generated pattern. B) Pattern A extended with recursion to exclude some false positives. Visualization created with SMARTSviewer.[32]

Limitations and Outlook. The current embedding of the SMARTSminer algorithm within the SMARTSeditor software puts the focus on interactivity. The algorithm is tuned to create good results quickly for the sake of slower run times for an exhaustive search. Certainly, the method has more potential concerning the running time of an exhaustive search.

Concerning the SMARTSminer pattern generation, there are currently two main limitations to name. First, generalized bond patterns have not been included so far. While this is a mostly technical issue, a more complicated feature would be the automatic generation of recursive atomic environment specifications. Furthermore, SMARTSeditor and SMARTSminer do not support stereochemistry descriptors yet. Using SMARTS as representation of the discriminative patterns also contains some limitations: In the syntax there is no way to represent e.g. chains of variable length. These limitations consequently also hold for SMARTSminer.

The greatest perspective for future development of the SMARTSeditor software and the SMARTSminer pattern generation is the support of disconnected patterns. So far, only connected patterns are found by the automated pattern generation. However, often molecules share functional parts which are connected to different scaffolds. For these cases, extending the pattern mining algorithm to finding disconnected patterns carries great promise.

CONCLUSION

With SMARTSminer we present an efficient algorithm for calculating discriminative chemical patterns between two sets of molecules in an interactive environment. The algorithm is embedded into the SMARTSeditor software for graphical chemical pattern design. Based on the SMARTS language, which contains well-defined pattern elements, SMARTSminer automatically generates discriminative patterns requiring a molecule set of one class and a molecule set of another class which shall be separated from the first one as the only input. By integrating the pattern generation into the interactive SMARTSeditor, the user gains several opportunities: The generated patterns can be inspected since they are visualized comprehensively. Their matches in the provided molecule sets can be browsed and highlighted within the molecules. Furthermore, several filter and score settings provide the user with the possibility to adapt the method according to the use case. Additionally, the full SMARTSeditor functionality can be used to extend or modify the generated patterns.

We have shown the applicability on several use cases covering the separation of actives versus decoys, kinase classification, analysis of data sets, and the characterization of chemical reaction centers.

Clearly, SMARTSminer has the potential for further extension of its pattern generation functionality. Nevertheless, it already proves to be a highly useful tool for experts and even scientists only rudimentary familiar with the SMARTS language. To our knowledge, it is a pioneer tool in combining discriminative chemical pattern analysis with an interactive editor.

ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.5b00323.

> Default collection of atomic SMARTS expressions (Figure S1) and a detailed result overview concerning Use case 3 (Kinase separation); smallest calculated discriminative chemical patterns for all DUD targets (PDF)

AUTHOR INFORMATION

Corresponding Author

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

Author Contributions

S. Bietz developed the SMARTSminer algorithm, K. T. Schomburg supported the integration into SMARTSeditor, M. Hilbig developed the functionality of the molecule sets within SMARTSeditor, and M. Rarey initiated and supervised the project.

Notes

The authors declare the following competing financial interest(s): The authors declare a potential financial interest in case the SMARTSminer software is licensed for a fee to nonacademic institutions in the future.

ACKNOWLEDGMENTS

We would like to thank Lars Wetzer and David Seier for their contributions to the SMARTSeditor implementation.

REFERENCES

(1) Nikolova, N.; Jaworska, J. Approaches to Measure Chemical Similarity–a Review. *QSAR Comb. Sci.* **2003**, *22*, 1006–1026.

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

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

(4) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *J. Chem. Inf. Model.* **2002**, *42*, 1273−1280.

(5) Proschak, E.; Wegner, J. K.; Schüller, A.; Schneider, G.; Fechner, U. Molecular Query Language (MQL) - A Context-Free Grammar for Substructure Matching. *J. Chem. Inf. Model.* **2007**, *47*, 295−301.

(6) Homer, R. W.; Swanson, J.; Jilek, R. J.; Hurst, T.; Clark, R. D. SYBYL Line Notation (SLN): A Single Notation to Represent Chemical Structures, Queries, Reactions, and Virtual Libraries. *J. Chem. Inf. Model.* **2008**, *48*, 2294−2307.

(7) James, C. A.; Weininger, D. *Daylight Theory Manual, version 4.9*; Daylight Chemical Information Systems, Inc.: Laguna Niguel, CA, 2011.

(8) Han, J.; Cheng, H.; Xin, D.; Yan, X. Frequent Pattern Mining: Current Status and Future Directions. *Data Min. Knowl. Disc.* **2007**, *15*, 55−86.

(9) de Graaf, E.; Kosters, W.; Kok, J.; Kazius, J. In *Research and Development in Intelligent Systems XXIV*; Bramer, M., Coenen, F., Petridis, M., Eds.; Springer London: London, 2008; pp 267−280.

(10) Seeland, M.; Girschick, T.; Buchwald, F.; Kramer, S. In *Machine Learning and Knowledge Discovery in Databases*; Balcázar, J. L., Bonchi, F., Gionis, A., Sebag, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2010; Vol. *6323*; pp 213−228.

(11) Seeland, M.; Johannes, A. K.; Kramer, S. Structural Clustering of Millions of Molecular Graphs. Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC'14. New York, New York, USA, 2014; pp 121−128.

(12) Yan, X.; Yu, P. S.; Han, J. Graph Indexing: A Frequent Structure-based Approach. Proceedings of the 2004 ACM SIGMOD international conference on Management of data - SIGMOD '04. New York, New York, USA, 2004; pp 335−346.

(13) Inokuchi, A. Generalized Substructures from a Set of Labeled Graphs. Fourth IEEE International Conference on Data Mining (ICDM'04). Brighton, UK, 2004; pp 415−418.

(14) Kazius, J.; Nijssen, S.; Kok, J. Substructure Mining Using Elaborate Chemical Representation. *J. Chem. Inf. Model.* **2006**, *46*, 597−605.

(15) Klopman, G. Artificial Intelligence Approach to Structure-Activity Studies. Computer Automated Structure Evaluation of Biological Activity of Organic Molecules. *J. Am. Chem. Soc.* **1984**, *106*, 7315−7321.

(16) Klopman, G. MULTICASE 1. A Hierarchical Computer Automated Structure Evaluation Program. *Quant. Struct.-Act. Relat.* **1992**, *11*, 176−184.

(17) Helma, C.; Cramer, T.; Kramer, S.; De Raedt, L. Data Mining and Machine Learning Techniques for the Identification of Mutagenicity Inducing Substructures and Structure Activity Relationships of Noncongeneric Compounds. *J. Chem. Inf. Model.* **2004**, *44*, 1402−1411.

(18) Borgelt, C.; Berthold, M. Mining Molecular Fragments: Finding Relevant Substructures of Molecules. *2002 IEEE International Conference on Data Mining, 2002. Proceedings* **2002**, 51−58.

(19) Hofer, H.; Borgelt, C.; Berthold, M. R. Large Scale Mining of Molecular Fragments with Wildcards. *Intelligent Data Analysis* **2004**, *8*, 495−504.

(20) Meinl, T.; Borgelt, C.; Berthold, M. Fragments with Fuzzy Chains in Molecular Databases. Proc. 2nd Int. Workshop on Mining Graphs, Trees and Sequences (MGTS 2004). Pisa, Italy, 2004; pp 49−60.

(21) Ting, R. M. H.; Bailey, J. Mining Minimal Contrast Subgraph Patterns. 6th SIAM international conference on data mining (SDM 2006). Bethesda, Maryland, USA, 2006; pp 638−642.

(22) Maunz, A.; Helma, C.; Cramer, T.; Kramer, S. In *Machine Learning and Knowledge Discovery in Databases*; Balczar, J., Bonchi, F., Gionis, A., Sebag, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin Heidelberg, 2010; Vol. *6322*, pp 353−368.

(23) Deshpande, M.; Kuramochi, M.; Wale, N.; Karypis, G. Frequent Substructure-based Approaches for Classifying Chemical Compounds. *IEEE Transactions on Knowledge and Data Engineering* **2005**, *17*, 1036−1050.

(24) Dominik, A.; Walczak, Z.; Wojciechowski, J. In *Adaptive and Natural Computing Algorithms*; Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2007; Vol. *4431*, pp 772−781.

(25) Please note the different usage of the term *pattern* in comparison to *chemical pattern* which we use here in the sense of an abstract (connected) molecular substructure.

(26) Dong, G.; Li, J. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA, 1999; pp 43−52.

(27) Auer, J.; Bajorath, J. Emerging Chemical Patterns: A new Methodology for Molecular Classification and Compound Selection. *J. Chem. Inf. Model.* **2006**, *46*, 2502−2514.

(28) Sherhod, R.; Judson, P. N.; Hanser, T.; Vessey, J. D.; Webb, S. J.; Gillet, V. J. Emerging Pattern Mining to Aid Toxicological Knowledge Discovery. *J. Chem. Inf. Model.* **2014**, *54*, 1864−1879.

(29) Poezevara, G.; Cuissart, B.; Crmilleux, B. In *Foundations of Intelligent Systems*; Rauch, J., Ra, Z., Berka, P., Elomaa, T., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2009; Vol. *5722*, pp 45−55.

(30) Takigawa, I.; Mamitsuka, H. Graph Mining: Procedure, Application to Drug Discovery and Recent Advances. *Drug Discovery Today* **2013**, *18*, 50−57.

(31) Lepailleur, A.; Poezevara, G.; Bureau, R. Automated Detection of Structural Alerts (Chemical Fragments) in (Eco) Toxicology. *Comput. Struct. Biotechnol. J.* **2013**, *5*, 1−8.

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

(33) Urbaczek, S.; Kolodzik, A.; Fischer, J. R.; Lippert, T.; Heuser, S.; Groth, I.; Schulz-Gasch, T.; Rarey, M. NAOMI: On the Almost Trivial Task of Reading Molecules from Different File Formats. *J. Chem. Inf. Model.* **2011**, *51*, 3199−3207.

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

(35) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.* **2006**, *49*, 6789−6801.

(36) Schneider, N.; Hindle, S.; Lange, G.; Klein, R.; Albrecht, J.; Briem, H.; Beyer, K.; Claußen, H.; Gastreich, M.; Lemmen, C.; Rarey, M. Substantial Improvements in Large-Scale Redocking and Screening using the Novel HYDE Scoring Function. *J. Comput.-Aided Mol. Des.* **2012**, *26*, 701−723.

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

(38) Stierand, K.; Rarey, M. Drawing the PDB: Protein-Ligand Complexes in Two Dimensions. *ACS Med. Chem. Lett.* **2010**, *1*, 540−545.

(39) Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Woolsey, J. DrugBank: A Comprehensive Resource for in silico Drug Discovery and Exploration. *Nucleic Acids Res.* **2006**, *34*, D668−D672.

(40) Enoch, S.; Madden, J.; Cronin, M. Identification of Mechanisms of Toxic Action for Skin Sensitisation using a SMARTS Pattern based Approach. *SAR QSAR Environ. Res.* **2008**, *19*, 555−578.