Journal of Computer-Aided Molecular Design, **15:** 497–520, 2001.
KLUWER/ESCOM

Similarity searching in large combinatorial chemistry spaces

Matthias Rarey[a,*] & Martin Stahl[b]
[a]*GMD-German National Research Center for Information Technology, Institute for Algorithms and Scientific Computing (SCAI), 53754 Sankt Augustin, Germany;* [b]*F. Hoffmann – La Roche AG, Pharmaceutical Division, Molecular Design and Bioinformatics, CH-4070 Basel, Switzerland*

Received 24 October 2000; accepted 16 March 2001

Key words: chemistry spaces, combinatorial libraries, COX-2 inhibitors, *de-novo* design, dopamine antagonists, feature trees, histamine antagonists, molecular similarity, similarity searching

Summary

We present a novel algorithm, called Ftrees-FS, for similarity searching in large chemistry spaces based on dynamic programming. Given a query compound, the algorithm generates sets of compounds from a given chemistry space that are similar to the query. The similarity search is based on the feature tree similarity measure representing molecules by tree structures. This descriptor allows handling combinatorial chemistry spaces as a whole instead of looking at subsets of enumerated compounds. Within few minutes of computing time, the algorithm is able to find the most similar compound in very large spaces as well as sets of compounds at an arbitrary similarity level. In addition, the diversity among the generated compounds can be controlled. A set of 17 000 fragments of known drugs, generated by the RECAP procedure from the World Drug Index, was used as the search chemistry space. These fragments can be combined to more than 10^{18} compounds of reasonable size. For validation, known antagonists/inhibitors of several targets including dopamine D4, histamine H1, and COX2 are used as queries. Comparison of the compounds created by Ftrees-FS to other known actives demonstrates the ability of the method to jump between structurally unrelated molecule classes.

Introduction

Tools for fast molecular similarity searching are firmly established as a rational approach to drug discovery [1–6]. The underlying idea of these methods is to find a descriptor for molecules which can be efficiently compared while preserving the physicochemical information necessary to identify biologically similar behavior. The best known examples for such descriptors are structural keys and molecule fingerprints representing the occurrence or absence of small fragments or paths in the molecular graph [7, 8]. Many of these methods have been successfully employed to search databases of existing or ‘virtual’, but explicitly formulated, molecules.

In recent years, computational methods to handle and select compounds from combinatorial libraries have gained importance [9]. The descriptors developed for database searching can still be applied to search within these spaces. However, it is difficult to arrive at similarity values for library molecules from calculations on their fragments, and, although there are efforts under way to speed up similarity calculations on combinatorial libraries [10], enumeration cannot be avoided. While enumeration is feasible for real combinatorial libraries with up to several hundred thousand molecules, it becomes impossible for virtual libraries that are several orders of magnitude larger. For searching in these large libraries, new descriptors and comparison algorithms are needed capable of handling libraries in their closed form rather than as explicitly enumerated molecular structures.

The idea of searching in large virtual combinatorial libraries can be carried one step further by performing similarity searching in more generally defined chemistry spaces. Chemistry spaces consist of several

*To whom correspondence should be addressed. E-mail: rarey@gmd.de thousand fragments that can be combined by a set of rules modeling simple chemical reactions. These spaces are still combinatorial in nature, since any fragment in a molecule can be replaced by a different fragment according to the same rules of synthesis. However, molecules from such a space contain a variable number of fragments and do not follow a unique synthesis scheme. A method to obtain such a combinatorial chemistry space is the RECAP procedure applying retrosynthetic rules to fragment databases of drug molecules [11]. This method is used here and will be discussed in more detail later.

So far, only few approaches aiming at searching in these large combinatorial spaces have been published. Schneider et al. [12] have employed an evolutionary search strategy in conjunction with the RECAP synthesis rules to generate molecular structures related to a query molecule in a software tool called TOPAS. As a descriptor, cross-correlation vectors of atomic properties over bond paths are used. The fitness function is the similarity of the molecule to a given query. TOPAS is able to search in large combinatorial spaces. An advantage of the search method is that it can be combined with arbitrary similarity descriptors. However, due to the nature of the genetic algorithm, which searches along paths of molecular structures, coverage of the search space is low. This results in a high probability of getting trapped in local minima. Douguet et al. [13] presented a genetic algorithm operating on SMILES strings [14] to generate molecules fulfilling a series of whole-molecule properties derived from QSAR analyses. Andrews and Cramer [15] used topomer shape similarity to identify plausible alternative hits related to known active compounds in a large virtual library consisting of several combinatorial sub-libraries. This approach is truly combinatorial in the sense that the search is performed on fragments that are then combined to complete molecules that are similar to the query. However, the method does not contain a generic search algorithm since constructed molecules always consists of two or three fragments only.

Here we present a novel method for similarity searching in combinatorial chemistry spaces, called *Feature Trees Fragment Space Search* (Ftrees-FS), that is unique in several respects. Being based on the feature tree descriptor [16], which describes a molecule by its major building blocks in a non-linear fashion, it can be directly applied to searching combinatorial spaces. In addition, the feature tree descriptor, validated on several inhibitor classes, has a high performance in selecting active compounds from data sets and shows little dependence on the two-dimensional structure of the compared molecules [17, 18]. The similarity search algorithm itself is based on a dynamic programming scheme. The method is therefore able to explicitly search for the most similar molecule without getting trapped in local minima. It handles arbitrary, even infinite, chemistry spaces in very short search times. For practical applications, we extended the method such that we can directly search for compounds at arbitrary similarity levels with respect to the query. This concept allows for generating compound sets that are structurally reasonably unrelated to the query but still contain plausible hits. Finally, we added concepts for directly creating structurally diverse sets of compounds. By means of test calculations on several sets of known inhibitors of various targets, we demonstrate that the method is able to generate compounds that are chemically distinct from the query compounds and at the same time closely related to other known inhibitors of the same target. This thorough validation allows the conclusion that the Ftrees-FS approach is capable of 'lead hopping', thereby supporting and augmenting the chemist's imagination.

In the following, we first give a brief introduction into the feature tree descriptor including two new concepts for describing shape properties as a feature. We then describe in more detail how chemistry spaces are defined and translated into feature tree fragment spaces. Third, the basic algorithm for comparing pairs of molecules in Ftrees-FS, the match-search algorithm, is outlined. Finally, we explain the dynamic programming scheme which is the key algorithm in Ftrees-FS, first for searching fragment pairs, then for searching multi-fragment collections. In the results section, Ftrees-FS is validated on various targets for which diverse sets of known inhibitors are available. For dopamine D4 and histamine H1 antagonists as well as COX-2 inhibitors, several examples of queries, retrieved compounds and related active compounds are shown. In order to demonstrate the wide applicability of the software, individual examples are shown for four further targets.

Materials and methods

The feature tree descriptor

The feature tree descriptor differs from most other descriptors used for similarity searching in that it is a non-linear representation of the molecule. Instead of

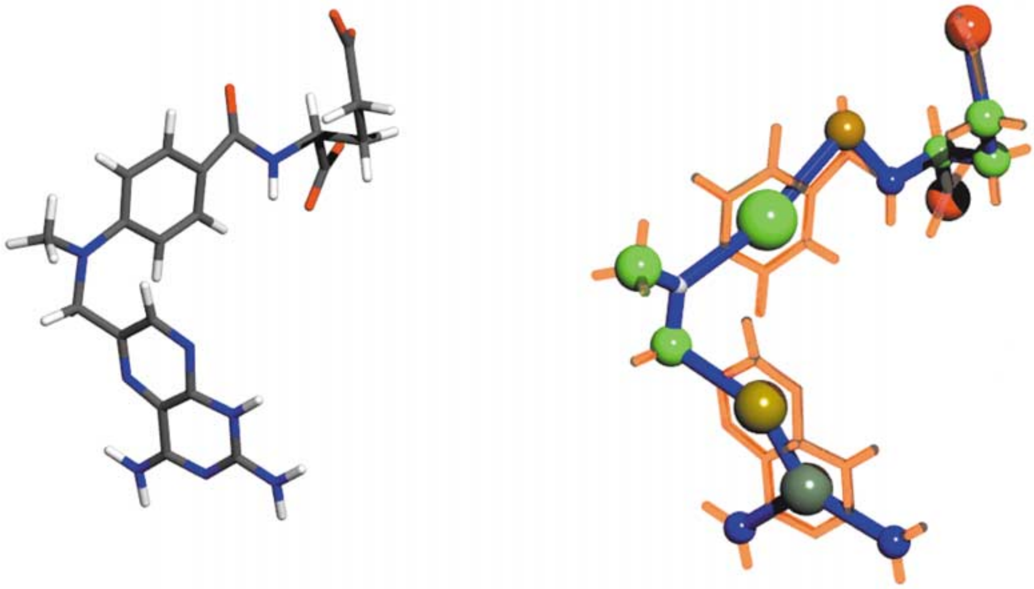


Figure 1. A drug molecule and its corresponding feature tree. The nodes of the feature tree represent building blocks of the molecule, the edges link the nodes according to the topology of the molecule. The nodes are labeled with features representing steric and physicochemical properties of the building block. Features are visualized by colors and size of the feature tree nodes.

employing a bit string or vector, the molecule is represented by a tree structure, whose nodes represent small fragments or building blocks of the molecule. Two nodes are connected by an edge in the feature tree if the corresponding building blocks are connected by a covalent bond or have atoms in common. The generation of a feature tree from a molecule is a simple procedure: Ring systems are decomposed into elementary rings. These elementary rings as well as all non-terminal, acyclic atoms together with adjacent terminal atoms form the set of feature tree nodes which are then connected according to the molecular graph of the molecule. An example of a molecule and its feature tree is shown in Figure 1.

The nodes of the feature tree are labeled with information on properties of the building blocks they represent. These properties are called features in this context. To use a feature in the comparison algorithm, three operations must be defined: a *generator*, which creates the feature from the building block, a *combinator*, which combines the features of two adjacent feature tree nodes two a single feature, and a *comparator*, which assigns a similarity value between 0 and 1 to a pair of features. Besides the fact that these operations must exist, there are no restrictions on a feature: it can be a simple number, a vector, or even a more complex object. In practice, we have used two linearly combined features, the first to describe the shape and size of the building block, called the shape descriptor, the second to describe its chemical properties, i.e., its ability to form intermolecular interactions, called the chemistry descriptor. For the chemistry descriptor, a simple profile of the interacting groups is created based on the FlexX interaction types [19, 20] as already described in the first feature tree paper [16]. The shape descriptor is the topic of the next section.

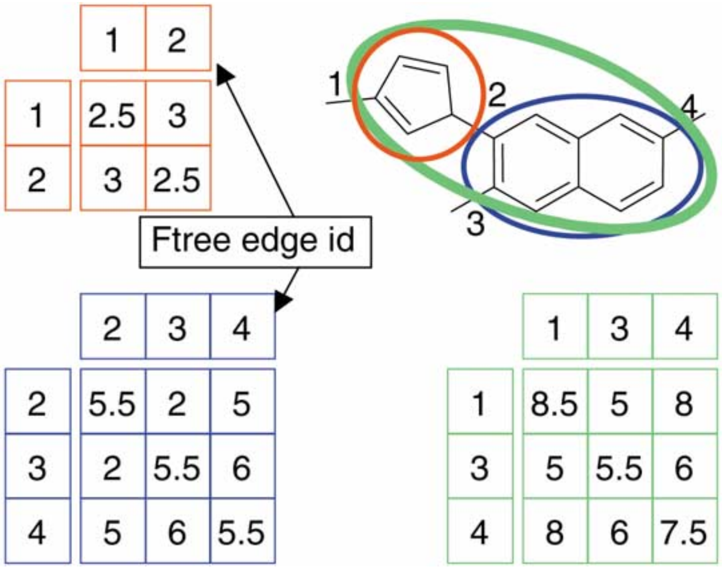
Novel shape descriptors

Since a feature tree is intended to be a conformation-independent description of the molecule, describing shape as a feature is not straightforward. So far, an approximated van der Waals volume has been used describing the size of a fragment rather than its shape. The similarity value for the volume term is

$$\begin{aligned} c_{\mathrm{vol}}(a,b) &= 2\min(vol(a), vol(b)) \Big/ (vol(a) \\ &\quad + vol(b)). \end{aligned}$$

For searching in chemistry spaces, we extended the shape descriptor by two optional elements, the RC- and the MPL-descriptor.

The RC-descriptor is simply the number of ring closures found in a building block. For combining features, the number of ring closures of two building blocks are added. The similarity value is calculated



	1	2
1	2.5	3
2	3	2.5

	2	3	4
2	5.5	2	5
3	2	5.5	6
4	5	6	5.5

	1	3	4
1	8.5	5	8
3	5	5.5	6
4	8	6	7.5

Figure 2. Two attached building blocks and its MPL matrices shown in red and blue. On the top and left of each matrix, the links corresponding to the columns and rows are indicated. In green, the MPL matrix for the combined building blocks is shown. It can be derived by adding path lengths for paths over the link bond (no. 2).

analogous to the volume term as

$$c_{rc}(a,b) = 2(\min(rc(a), rc(b)) + 1) \big/ (rc(a) + rc(b) + 2).$$

The MPL-descriptor represents the path lengths contained in a building block. If the building block has n outgoing bonds, the descriptor is an $n \times n$ matrix containing the path length between two outgoing bonds on its off-diagonal positions and the length of an outgoing bond to the most distant atom of the building block on its diagonal positions. An example for an MPL-matrix is given in Figure 2. For comparing two MPL-matrices A and B with dimensions n_A and n_B, we distinguish between two cases. If either n_A or n_B is 1, the path lengths to the most distant atom (diagonal elements) are compared, otherwise the maximum distance between two outgoing bonds (off-diagonal elements) are compared:

$$c_{mpl}(a,b) = 2(\min(mpl(A), mpl(B)) + 1) \big/ (mpl(A) + mpl(B) + 2)$$

$$mpl(X) = \begin{cases} \max_i (X[i,i]) : n_A = 1 \vee n_B = 1, \\ \max\limits_{\substack{i,j \\ i \neq j}} (X[i,j]) : \text{otherwise}. \end{cases}$$

To combine two MPL-matrices of adjacent building blocks to a single MPL-matrix (combinator operation), the bond linking the two building blocks must be identified. Then the elements of the combined MPL-matrix can be calculated by adding path lengths from the two outgoing bonds to the link bond as illustrated in Figure 2.

Finally, the overall shape similarity is defined as

$$c_{\text{shape}}(a,b) = c_{\text{vol}}(a,b) c_{rc}(a,b) c_{mpl}(a,b).$$

The MPL-descriptor is able to distinguish between elongated and compact building blocks. Moreover, it is able to distinguish between different substitution pattern at ring systems, for example between *ortho*-, *meta*-, and *para*-substituted rings. The disadvantage is however, that, due to its complexity and the large number of shape comparisons during a feature tree comparison, the performance drops by a factor of about three when this descriptor is used. We therefore decided to use the MPL-descriptor only in a selected example (COX-2, see Results) were this distinction is of importance.

Feature tree fragment spaces

So far, the feature tree descriptor has been used to represent individual molecules, but due to its modular structure, it can be extended in a straightforward manner to represent molecule fragments and chemistry spaces constructed thereof. The chemistry spaces used in combination with feature trees consist of a set of fragments, each fragment having at least one open valence, called a link. Each link is of a specific type defining how the link is used to connect to other fragments. For each pair of link types, a link compatibility matrix defines whether they can be connected or not. Properties of the connecting bond (bond type, length, etc.) and the connected atoms (hybridization and charge) after forming the bond are also predefined ('connect information'). Finally, a complete library definition contains information on how a link should be saturated in case that it is not used for a connection ('terminal information').

The chemistry space used here was generated from the World Drug Index (WDI) [21] according to the RECAP procedure by Lewell et al. [11]. The eleven retrosynthetic rules for fragmentation of the WDI database were also used as synthesis rules as described for the program TOPAS mentioned above [12]. For the present study, compatibility between links was slightly generalized. For example, there are no separate lists of alkyl fragments for the formation of ether bonds and bonds to nitrogen atoms in different functional groups. These lists have been combined to one class of alkyl substituents. Likewise, alkoxy groups for ether and ester formation as well as amines for urea, amide and sulfonamide formation were combined to one class. The twelve resulting link types together with their terminal information are shown in Table 1.

The fragmentation procedure of the WDI led to a set of 16 780 fragments containing 21 386 links in total. The majority of fragments have a single link (12 817 fragments), 3433 fragments have two, 443 fragments have three, and 87 fragments have more than three links. Table 1 contains the number of links found for each link type.

Further analysis of the space reveals that it is connected and infinite: *Connected* means that every pair of fragments of the space can theoretically occur in the same molecule. The space is *infinite*, because due to the link type combinations at the multi-link fragments, an infinite number of molecules can be constructed containing multiple copies of the same fragment. For practical purposes, these space properties are of little relevance, since the size of a drug molecule is limited. We therefore calculated the size of the space as a function of the longest chain of fragments in a molecule and as a function of the total number of fragments in a molecule. The resulting numbers are given in Table 2. The most realistic estimate for drug-like molecules is probably to limit the number of fragments to five, resulting in about 10^{18} molecules. Note that the search algorithm presented below is not limited to a maximum number of fragments.

In order to translate the chemistry space into a feature tree space, each fragment is converted to a feature tree in the same way as an individual molecule with one exception: The links, which are represented as dummy atoms in the fragments, are modeled as separate feature tree nodes, called link nodes. The link nodes contain the feature values of the corresponding terminal group of the link. If two fragments are connected, the feature tree of the pair of fragments can easily be created by removing the two corresponding link nodes and connecting the nodes adjacent to the link nodes as shown in Figure 3.

During the construction of a molecule from fragments, a link is either used to connect to another fragment or the corresponding terminal group is attached. Depending on the usage of the link, the features at the node adjacent to the link might differ. An example of this effect is shown in Figure 4. In order to handle both cases, we correct feature values as follows: Let l be the link node and a its neighboring node, let f be the feature in case that the terminal group is added and f' the feature in case that a different fragment is connected to a. The feature value for a is set to $f'(a)$ and is therefore correct if a fragment is connected to a. The feature value for l is set to $f(l) + f(a) - f'(a)$. Therefore, as long as the link node l is not separated from its neighboring node a, the feature value of l and a together is correctly set to $f(l) + f(a)$ (see also Figure 4).

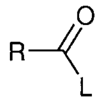
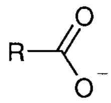
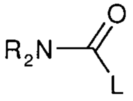
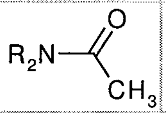
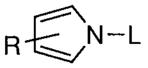
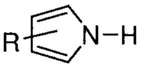
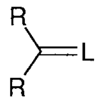
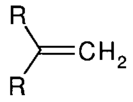
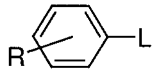
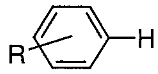
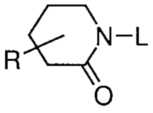
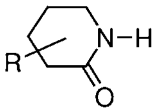
Comparing pairs of molecules

For comparing two molecules on the basis of their feature tree descriptors, an assignment of feature tree nodes, called a *matching*, must be found. Based on the matching, a similarity value can be calculated. First, a similarity value is assigned to each match by comparing the feature values of the matched feature tree nodes. Then, the similarity value for the molecule pair is the weighted sum of the match similarity values normalized by the size of the two molecules:

$$S_M(A,B) = \frac{\sum\limits_{m \in M} size(m) sim(m)}{u(size(A) + size(B)) + (1-u) \sum\limits_{m \in M} size(m)}$$

where A and B are the two molecules to compare, M is the set of matches, sim(m) is the similarity of a match and size(m) the size (number of non-hydrogen atoms) involved in match m. The parameter u allows to control the influence of the overall difference in size of

Table 1. Link types

link type	fragment	compatibe to	terminal	#links
carbonyl		amino [amide] alkoxy		2685
urea		amino [amide]		272
alkoxy	RO–L	carbonyl alkyl	RO–H	2379
sulfonyl	RSO_2–L	amino [amide]	RSO_2–CH_3	232
amino [amide]	R_2N–L	carbonyl urea sulfonyl	R_2N–H	3066
amino [amine]	R_2N–L	alkyl	R_2N–H	1530
amino [ammonium]	R_3N–L	alkyl	R_3N–H	103
aromatic amine		alkyl		919
alkyl	R–L	alkoxy amino [amine] amino [ammonium] aromatic amine lactame	R–H	4611
carben		carben		3843
aryl		aryl		1643
lactame		alkyl		103

Columns (f.l.t.r.): name of link type, corresponding fragment (L: link), compatible link types, number of occurrences in the WDI space.

the two molecules. Algorithms for comparing feature trees create a matching between the feature tree nodes optimizing the overall similarity value S_M. We have developed two different approaches, called the split-search and the match-search algorithm [16]. Here, we shortly describe the match-search algorithm since it is the basis for the new method for searching chemistry spaces.

The match-search algorithm is a dynamic programming scheme assigning sets of connected nodes, called subtrees, between two feature trees without gaps, i.e. there are no unmatched nodes located between matched nodes. The algorithm is illustrated in

Table 2. Number of molecules which can be constructed from the WDI space as a function of the longest path of fragments within the molecule (max. diameter, first row), and the maximum number of fragments within a molecule (second row).

	1	2	3	4	5
Max. diameter	1.68E+04	4.81E+07	8.91E+25	6.33E+43	7.82E+142
Max. # fragments	1.68E+04	4.81E+07	1.20E+11	7.46E+14	2.15E+18

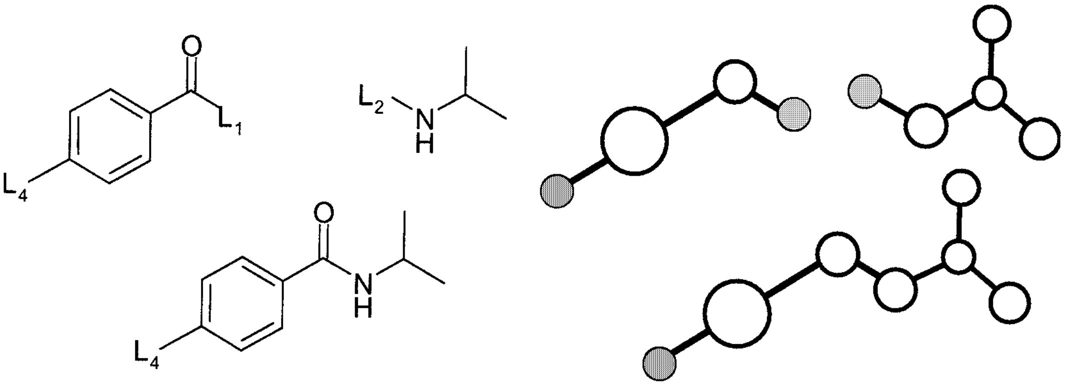


Figure 3. On the left, two fragments of a chemistry space and the corresponding connected fragment are shown. On the right, the corresponding feature trees are depicted.

Figure 5. Its central data structure is the dynamic programming matrix whose rows and columns are assigned to directed edges in the first and second feature tree, respectively. The information contained in one cell of this matrix is the best achievable similarity value by matching two subtrees starting with the so-called head nodes, the nodes that are pointed to by the two edges addressing the cell. The following steps are performed: (1) First, all matches starting with the head nodes and lying within given size limits, called extension matches, are enumerated. For each extension match, its similarity value and an estimate for the similarity value of the remaining part of the feature trees is calculated and a set of high similarity matches is selected. For each selected extension match, the following steps are performed. (2) To separate the extension match from the remainder of the feature tree, all edges leaving the match are cut. Each cut subtree of the first feature tree can now be combined with each cut subtree of the second feature tree. To find the best combination, the similarity value for each subtree combination is retrieved from the dynamic programming matrix. (3) Finding the best combination of the cut subtrees is a bipartite graph matching problem which can be solved efficiently. Based on the combination, an overall similarity value can be calculated by combining the similarity value of the extension match with those for the best combinations of cut subtrees. Finally, the best extension match is selected and the corresponding similarity value stored in the dynamic programming matrix. In step 2 of the match-search algorithm, the dynamic programming matrix is accessed to retrieve the similarity values for the cut subtrees. One way to implement this is to perform a recursive call of the match-search algorithm whenever the corresponding cell is not yet calculated. Alternatively, the directed edges of the feature trees can be processed in the order of increasing depth: The depth of a directed edge is defined to be the maximum length of a path starting with this edge as shown in Figure 6. Since in step 2 of the match-search algorithm only cells belonging to edges with smaller depths are accessed, no recursion is necessary.

Fragment space search algorithms

Here we describe the main algorithm for searching for molecules similar to a given query molecule in chemistry spaces as defined above. The algorithm consists of two phases. In the first phase, a so-called edge-link similarity table is calculated. The table is addressed by the directed edges of the query feature tree and the

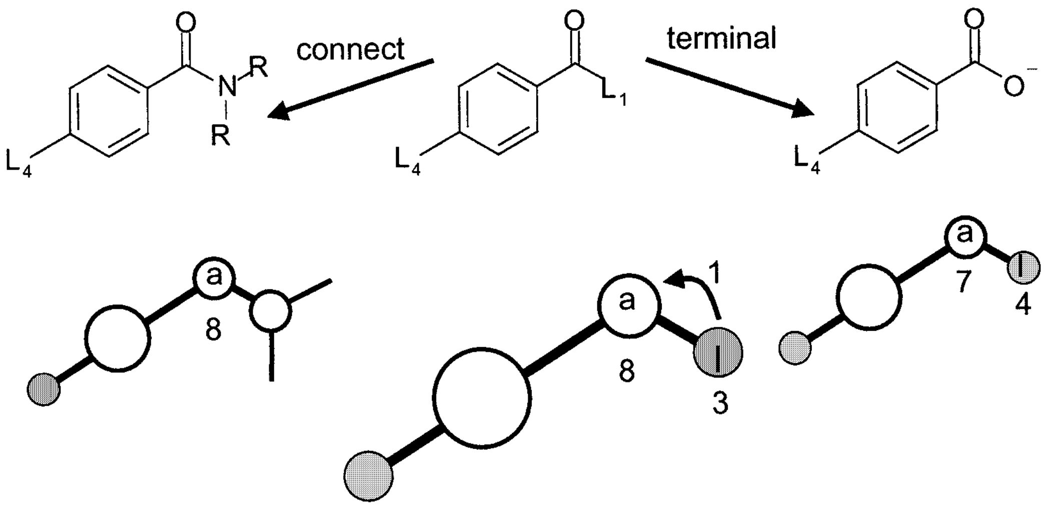


Figure 4. Link node correction: A link of a fragment (shown in the middle) can be either involved in the connection of two fragments (left) or it is terminal (right). The different environments influence feature values (numbers below nodes) for the neighboring node a. The difference in the feature value can be shifted towards the link node resulting in correct values for both cases.

link types of the chemistry space. For each edge-link type pair e, l, the table entry $T[e, l]$ contains a list of links of type l resulting in high similarity values if the subtree attached to a link with type l matches onto the subtree e is pointing to (see also Figure 7). The edge-link similarity table is the central data structure of the search algorithm. In the second phase, molecules are constructed by connecting fragments according to the lists of links stored in the edge-link similarity table. Before the generic case of searching for molecules consisting of multiple fragments is explained, the more simple case of searching for fragment pairs is described.

Pair-fragment search

In the first phase, we can create the edge-link similarity table cell by cell. For each combination of a directed edge e from the query and a link type l, we iterate through all links of type l in the chemistry space. For each link, we execute the recursive part of the match-search algorithm starting with a cut at edge e and at the link as shown in Figure 8 . This results in a similarity value for the fragment attached to the link and the part of the query attached to edge e. In the edge-link similarity table, the k links resulting in the highest similarity values are stored. In order to speed up the search, only fragments with a size in a range from 0.66 to 1.5 times the size of the query subtree are considered. In the second phase, all pair combinations of fragments implied by the links stored in the edge-link similarity table have to be built. For each pair of anti-parallel directed edges e, e' in the query molecule and all pairs l, l' of compatible link types, all combinations of links from the edge-link similarity table cells $T[e, l]$ and $T[e', l']$ are enumerated. For each combination, a similarity value is calculated from the similarity values calculated for the links. The combinations with highest similarity values are stored and the corresponding molecules can be generated by connecting the fragments attached to the combined links.

Multi-fragment search

To calculate the edge-link similarity matrix for the multi-fragment search, a dynamic programming scheme is employed. Recursion is avoided by processing the directed edges of the query in order of increasing depth as explained above for the match-search algorithm. Following this order, the matrix can again be computed cell by cell. As has been the case in the two-fragment search, the recursive part of the match-search algorithm is started for each pair of subtrees attached to a directed edge e of the query and a link of type l. The difference is that the subtree attached to the link might contain additional link nodes. While the match-search algorithm proceeds, a recursive call might come up matching an edge pointing to an additional link node l' with an edge e' from the query subtree as illustrated in Figure 9. A similarity value for this pairing can be calculated from the edge-link similarity table by maximizing the similarity value over all cells belonging to edge e' and link types which are

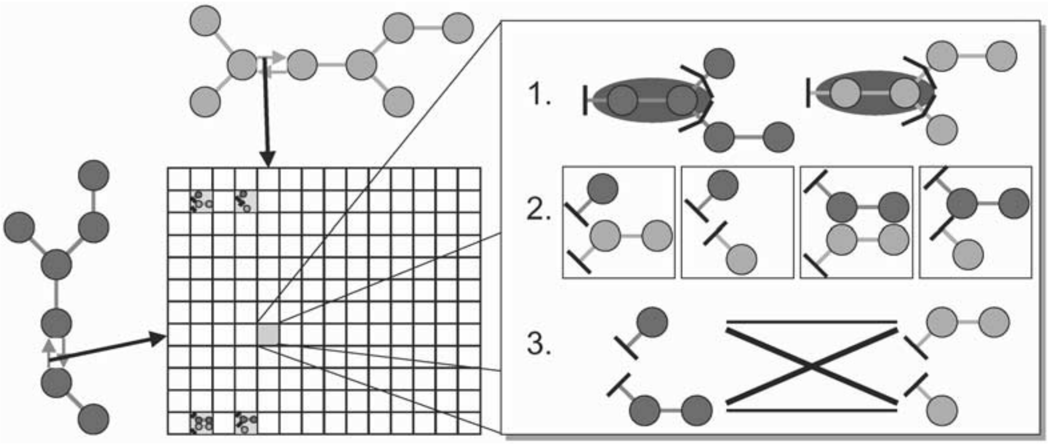


Figure 5. Illustration of the match-search algorithm. On the left, the dynamic programming matrix for comparison of two feature trees is shown. Each cell contains a similarity value as well as matching data for the best matching of the two subtrees that are pointed to by the directed edges addressing the cell. On the right, the three steps for computing this information is shown for one cell. First, a set of extension matches are computed, second, the pair-wise similarity data for all cut subtrees are extracted from the dynamic programming matrix, and third, a bipartite matching algorithm chooses the best assignment of cut subtrees.

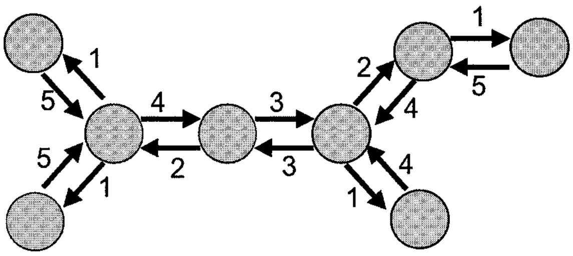


Figure 6. Feature tree with depth values shown at the directed edges.

compatible with l'. In doing so, we obtain the similarity value for the most similar fragment we can attach at link node l'. Since e' has a smaller depth than e, the table cells belonging to e' are already calculated.

After running the dynamic programming algorithm, the edge-link similarity table contains a very dense, recursive description of a solution space: For each edge in the query structure and each link type, it contains a list of links from the chemistry space yielding high similarity values if the attached subtrees were matched. In order to generate complete molecules, information is needed on how the subtree attached to the link was matched to the query. For each additional link the subtree contains, we therefore store the edge of the query matched to the edge pointing to the link as shown in Figure 7.

The most similar molecule can now be constructed recursively from the edge-link similarity table as follows: As in the pair-fragment search, we iterate over all query edges and link types. For each pairing of e, l and e', l' where e and e' are anti-parallel edges and l and l' are compatible links, we construct the most similar molecule fragment mapped to e and e' independently and connect them to a single molecule. For each directed edge, we take the first entry in the edge-link similarity table giving us the most similar fragment which could be matched to it. For each further link l'' it contains, the link type and the directed edge e'' of the query, to which the link l'' is matched, are stored in the table. Therefore, we can identify the most similar link matched to e'' type compatible to l'' in the table. If the fragment attached to it contains further links, we recursively proceed until no further open links are present. The recursive construction phase is shown in Figure 10.

In most applications, a list of similar molecules instead of a single molecule is required. We can generate these by always considering a list of links during the recursive construction phase instead of only the most similar one. If a fragment has more than a single further link, a list of similar fragments result for each link. In this case, we have to create all combinations of fragments in order to get the full set of molecules similar to the query. Note that this extension to molecule sets is heuristic. In other words, while the algorithm is able to create the most similar molecule it is not able to create the k-th similar one exactly.

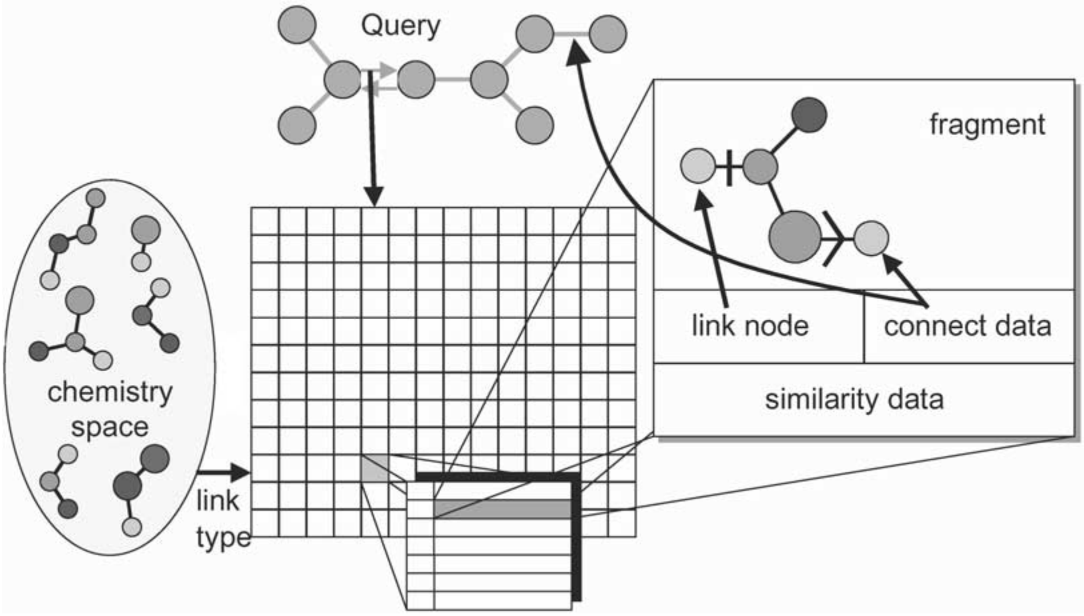


Figure 7. The edge-link similarity table is the central data structure for the fragment space similarity search. It is addressed by a directed edge of the query and a link type of the chemistry space. Each cell contains a list of links with the corresponding fragment. In addition, information about the similarity calculation (weighted similarity, overall size of matched subtrees, etc.) and about the matching of further link nodes (connect data) are stored.

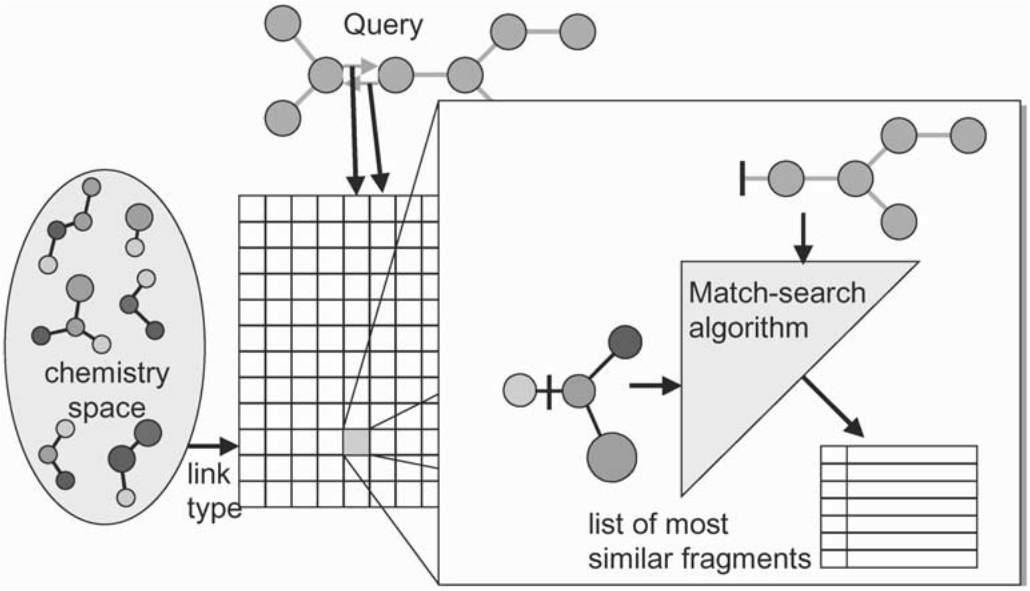


Figure 8. Pair-fragment search: For each cell, the recursive part of the match-search algorithm is executed for each link of the corresponding link type contained in the chemistry space. The match-search algorithm starts with the directed edge of the query addressing the cell and the edge leaving the link node.

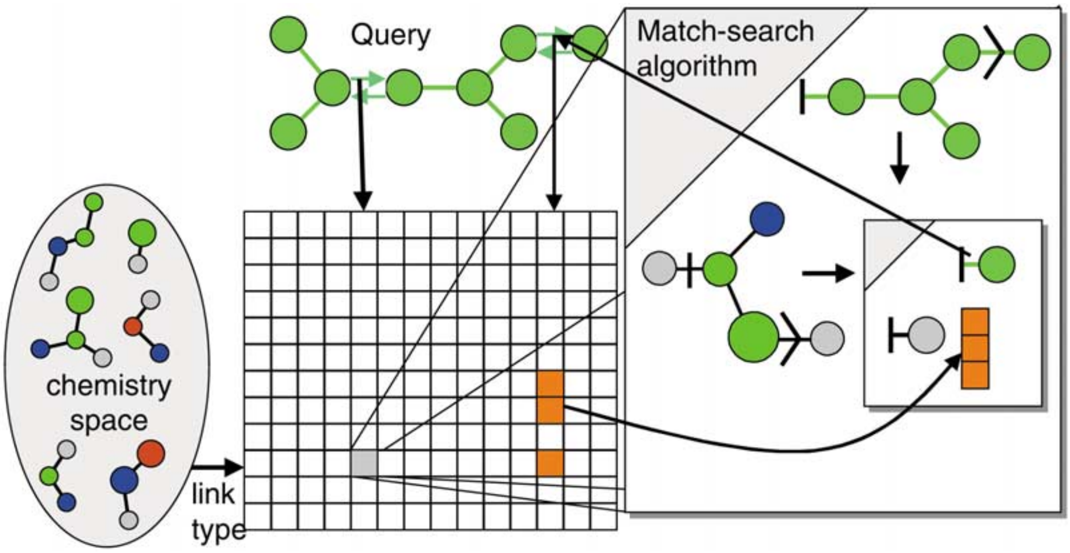


Figure 9. Multi-fragment search (1): The figure illustrates the recursive step within the match-search algorithm resulting in the dynamic programming scheme. During execution of the match-search algorithm, a recursive call containing a link node as input is answered by a lookup in the edge-link similarity table. The column is addressed by the directed edge of the query being the second parameter of the match-search algorithm call, the rows are selected to be all link types compatible with the cut link node (cells are highlighted in orange). The highest similarity value can be found by maximizing over these cells.

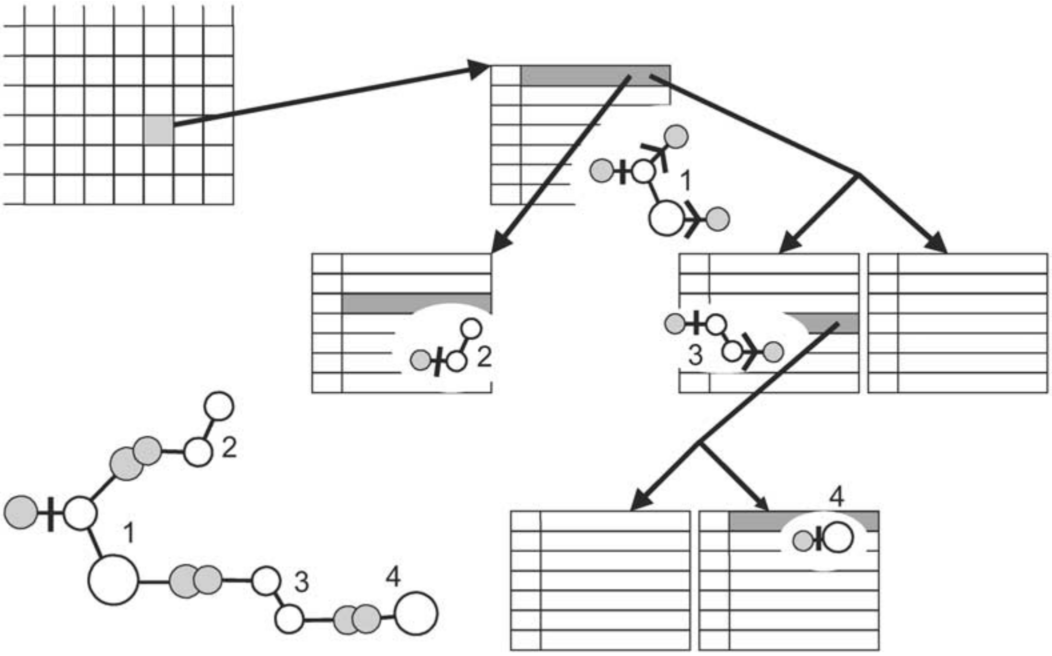


Figure 10. Multi-fragment search (2): A complete molecule (here shown as its feature tree) can be constructed from the edge-link similarity table by recursively following the connect information. In the upper right corner, a fragment with two further link nodes is shown. For each of the links, we can look up the edge-link similarity table (column: matched query edge, row: link-types compatible to the type of the link at the fragment) to find similar fragments which could be added. These fragments might themselves contain further links.

Solution set diversity and target similarity

For practical applications, we added two concepts to the similarity search algorithm, the first to achieve solution set diversity, the second to create molecules sufficiently different from the query structure.

Solution set diversity

Due to the usage of the limited set of fragments stored in the edge-link similarity table, it is likely that several very similar molecules will be constructed in the final phase of the algorithm. In fact, since a molecule is always associated with a mapping to the query structure, the same molecule can be constructed twice with slightly different mappings. Whenever a new molecule is created, we therefore apply one of the following diversity filters to avoid repetition:

- The maximum number of fragments in common is limited to k.
- The minimum number of different fragments is limited to k.
- The maximum similarity between two molecules in the data set is limited to k.

Before a newly created molecule is entered into the solution set, a list of all molecules contained in the solution set conflicting with respect to the diversity filter is created. If the new molecule is more similar to the query than the most similar conflicting one, it is entered into the solution set and the conflicting molecules are removed. Otherwise, the newly created molecule is rejected.

Target similarity

Since the search algorithm creates molecules from very large chemistry spaces, the resulting structures might be too similar to the query structure to be of interest in a drug design project. If the query molecule itself is contained in the space, the search algorithm will retrieve an identical copy or a very close analog from the space. In a drug design project, however, the aim of similarity searching is to find alternative active compounds which are structurally diverse from the query structure. We therefore have to search not for molecules with highest similarity but for molecules on a certain similarity level, called target similarity. The target similarity value is a trade-off between two competing goals, the search for biologically active compounds and the search for compounds structurally diverse from the query. The target similarity concept can be integrated into the search algorithm in a straightforward manner. During the dynamic programming algorithm in the first phase, links resulting in a similarity value with smallest deviation from the target similarity value are stored in the edge-link similarity table instead of those with highest similarity. During the construction of molecules, we can do the same for the final molecule list. Searching for molecules on a certain similarity level can in principle be done with every kind of similarity measure. In contrast to sequential similarity searching, here we apply the target similarity value already on the fragment level. We therefore end up with molecules whose dissimilarity is ‘spread’ over the whole molecule instead of molecules which are identical in one part of the structure and totally dissimilar in another part.

Computing times

All calculations were performed on a single 400 MHz, R12k processor of an SGI Origin 2000 computer. When searching the WDI fragment space, the process took less than 100 MB main memory in total. The match-search algorithm used for sequential database searching required about 9 ms per pair of feature trees. Our benchmark, a search with 397 queries in a database of 7925 compounds, could therefore be performed in about 8 h. Using the MPL-descriptor slows the computation down by approximately a factor of three resulting in about 27 ms per comparison. Computing times for searching the WDI fragment space with the multi-fragment search algorithm strongly depend on the size of the query molecule. A query with six rotatable bonds can be answered in less than a minute, queries with 10–14 rotatable bonds take about 3–4 min. Searching with very large queries (about 30 rotatable bonds) takes about 2–3 h. Therefore, for a typical drug molecule having less than 10 rotatable bonds, the WDI fragment space containing more than 10^{18} compounds can be searched with a hypothetical rate of 5×10^{13} compounds per millisecond or 2×10^{-14} milliseconds per compound.

Results and discussion

Determining the target similarity value

To provide a rationale for the way Ftrees-FS is applied in the following, it is necessary to discuss a number of general properties of chemical similarity measures employed in drug discovery processes. In practical applications of similarity searching with a single query

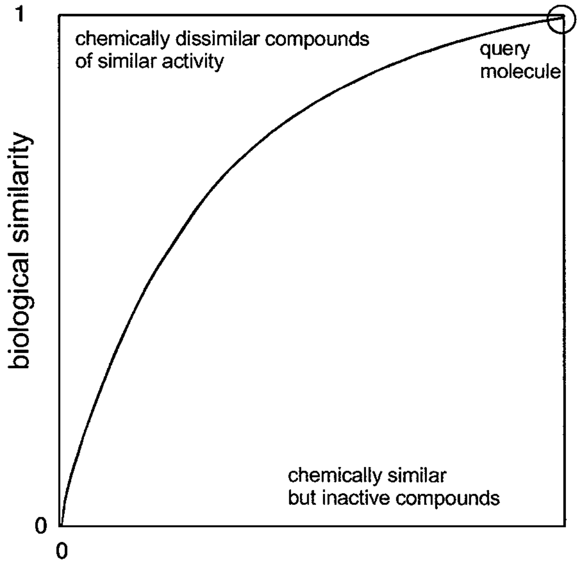


Figure 11. Hypothetical diagram of biological versus chemical similarity illustrating the behavior of a similarity measure that would be useful in drug discovery (curved line).

molecule, one aims at retrieving compounds of similar biological activity that are at the same time structurally as unrelated as possible to the query compound. The desire to simultaneously maximize ‘biological similarity’ and minimize ‘chemical similarity’ inevitably leads to a dilemma: On one hand, the similarity measure employed should describe biological activity as closely as possible, while on the other hand it must be based on chemical features, since the only given information is the chemical structure of a single compound. Compounds for which very high similarity values are calculated will therefore always be structurally related to the query, while at low similarity values there is little chance to find compounds with the same activity. The situation is schematically depicted in Figure 11 for the hypothetical situation that chemical and biological similarity can be expressed on absolute scales. The query molecule and compounds closely related to it are located at the upper right hand corner. The zone of interest for database searching is the upper left part of the diagram. The solution to the search dilemma is a compromise: A good similarity measure is able to extract some generally valid biological information from the query and can thus, while completely based on chemical features, classify structurally distinct compounds as still relatively similar. This behavior is illustrated by the curved line in Figure 11.

As a point in case, consider the results of a database ranking experiment with the dopamine D4 antagonist **1** as a query (Figure 12). A database consisting of 54 more dopamine D4 antagonists from the literature [22] and 7528 representative WDI compounds was ranked according to feature tree similarity to the query. The database ranking can be considered successful, since most D4 antagonists are among the top 5% of the database. A closer look at the 2% top ranking compounds reveals that the highest ranking compounds are indeed structurally very similar to the query molecules and structural diversity increases as the calculated similarity to the query decreases. The similarity value decreases not linearly, but drops quickly in the top region. At the same time, the density of active compounds also decreases quickly. There is thus an optimum search region, or similarity level, where the probability of finding active compounds is still high and structural diversity is large enough to provide the chance of discovering new chemical classes with the same biological activity.

Ftrees-FS can be used to directly generate sets of compounds such that their calculated similarity values to the query compound approach a defined target value. For most applications, best performance was obtained with a target similarity value of 0.90, leading to sets of structures in the range of 0.87 and 0.9, depending on the size of the compounds and diversity restrictions imposed on the solution set. In the database ranking shown in Figure 12, compounds in this similarity range rank very high, but can be structurally quite different from the query (e.g., compound

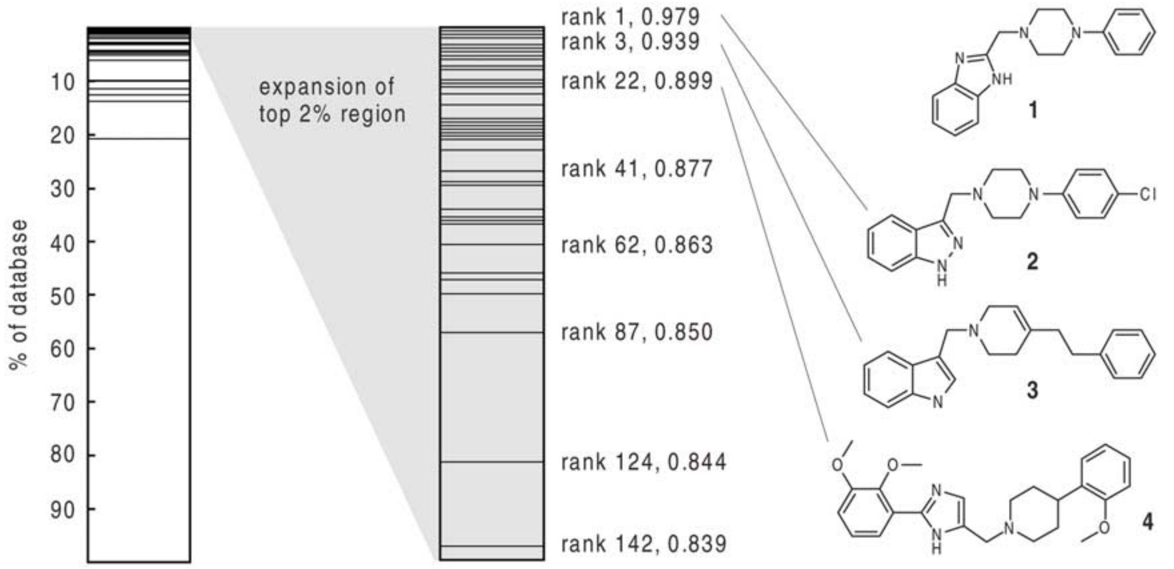


Figure 12. Results of a database ranking experiment of 7528 WDI compounds and 54 dopamine D4 antagonists with antagonist **1** as a query.

4 on rank 22). Another rationale for the target similarity value of 0.9 can be derived from Figure 13. It shows histograms of pairwise compound similarity among the D4 antagonists and between the WDI subset and the D4 antagonists. At a similarity level of 0.9, the probability of finding inactive compounds (WDI compounds with high calculated similarity to a D4 antagonist) is still relatively low, while the probability density of finding a related active compound reaches a maximum. Since the number of molecules that can potentially be generated with fragment spaces is enormously high, it is not advisable to set the target similarity to values below 0.9, where the rate of false positives increases quickly.

Dopamine D4 antagonists

The set of D4 antagonists served as a first test case for the feature tree fragment space software. As described in the methods section, the search algorithm is able to uniquely identify the most similar compound that can be generated to any given query. To illustrate this property, the 55 dopamine D4 antagonists were fragmented retrosynthetically according to the RECAP synthesis rules and the fragments added to the WDI fragment space. The 55 antagonists were then used as query molecules at a target similarity of 1.0. With few exceptions, the highest ranking compound was identical to the query structure. The exceptions were cases where either two fragments were indistinguishable on feature tree level (substitution patterns on benzene rings or regioisomers of heterocycles) or where the query structure had accidentally been entered in a different protonation state than the corresponding fragment.

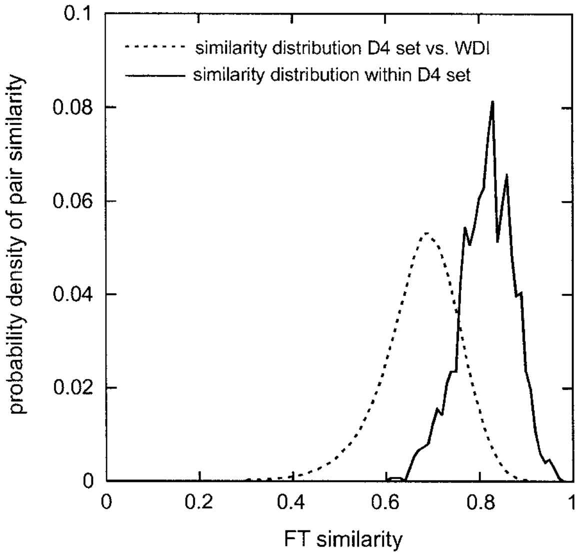


Figure 13. Pair-wise similarity distribution for a set of dopamine D4 antagonists and WDI compounds according to the feature tree similarity measure.

Figure 14 gives a visual impression of compounds generated at different similarity target values. The query molecule used is the low molecular weight D4 antagonist **5**, whose feature tree representation is a linear chain of five nodes. The six top ranking compounds differing in at least two fragments are plotted.

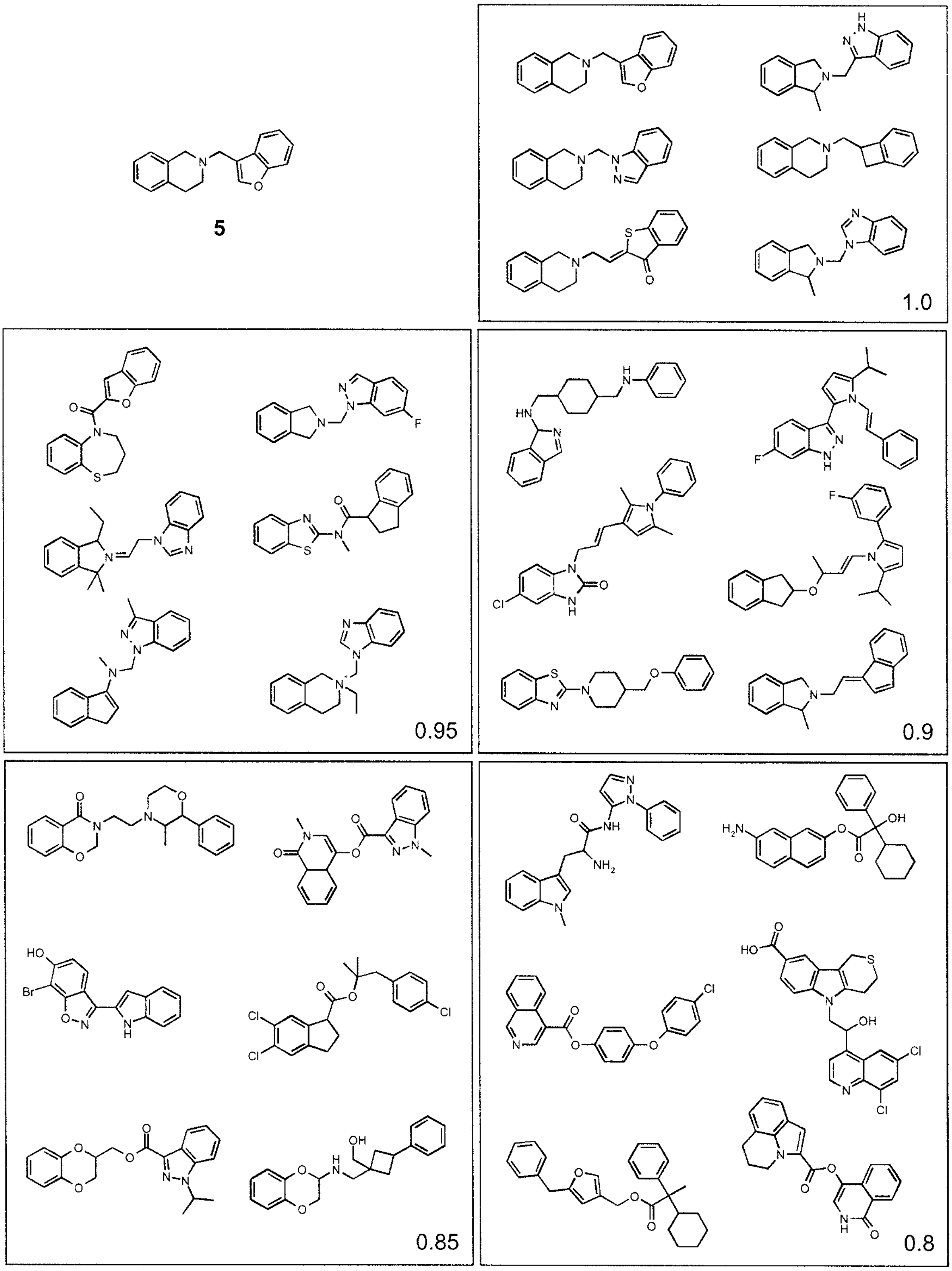


Figure 14. Sets of 6 top-ranking compounds generated with Ftrees-FS at various similarity levels (bottom right corner of each box) and with compound **5** as a query.

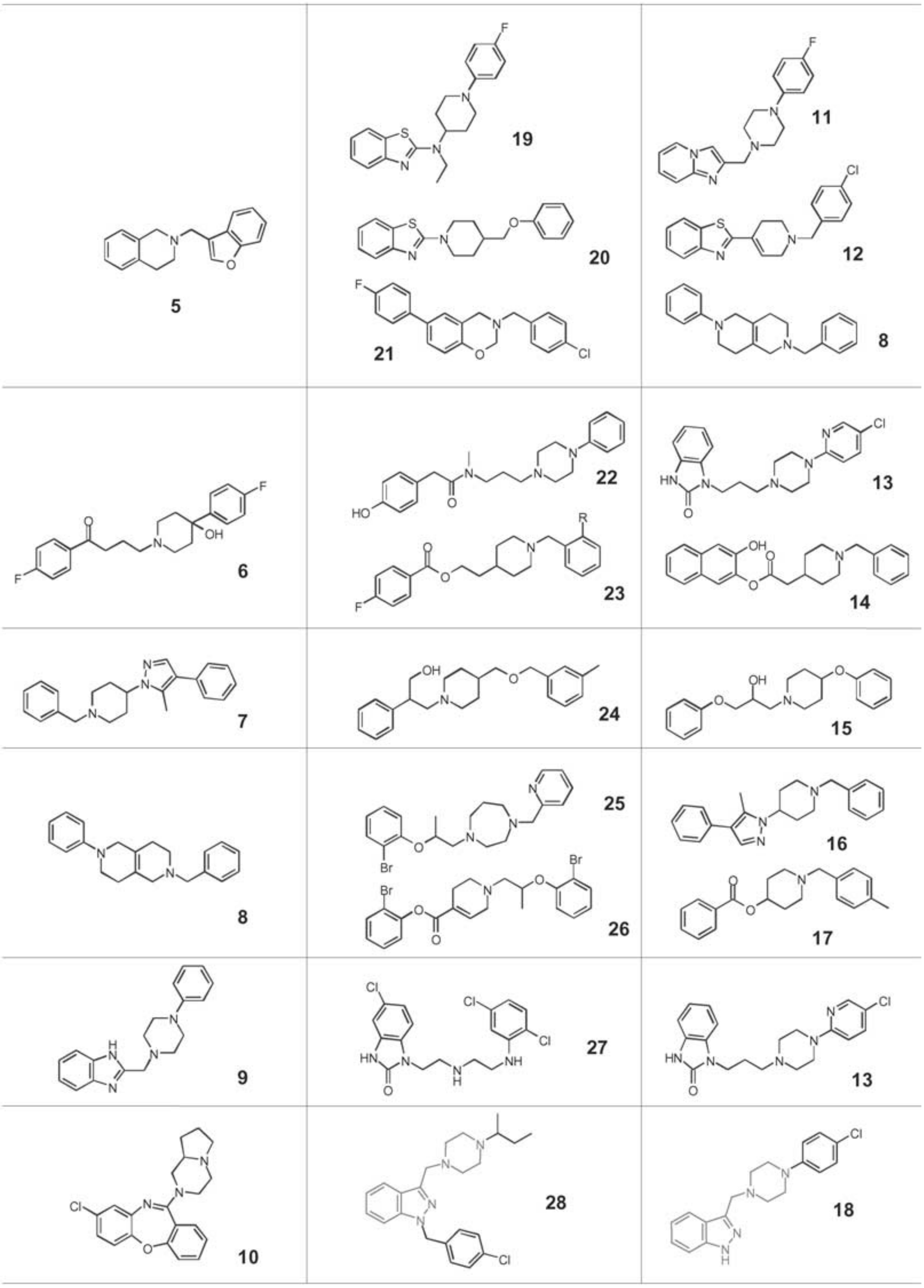


Figure 15. Query results for various dopamine D4 antagonists. Left column: query, middle column: plausible hit, right column: known antagonist related to the plausible hit.

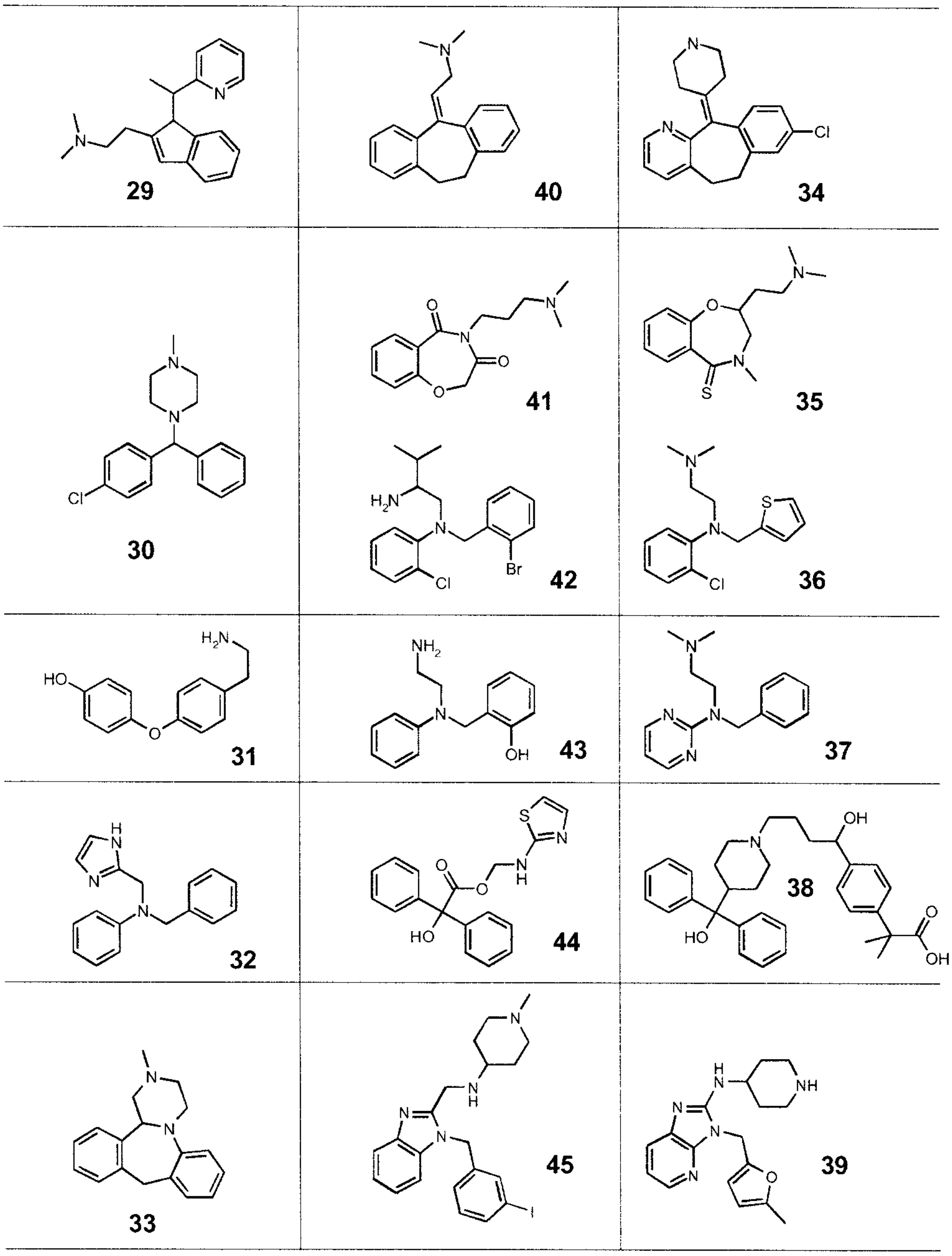


Figure 16. Query results for various histamine H1 antagonists. Left column: query, middle column: plausible hit, right column: known antagonist related to the plausible hit.

At similarity levels 1.0 and 0.95, all compounds follow the ring-linker-ring template of the query. At a level of 0.90, more rings appear, some bicyclic structures are broken up into separate rings, and more branched molecules are generated. This structural variability is of key importance in the search for new classes of active compounds. At even lower similarity levels, more functional groups appear and compounds of increasing molecular weight are constructed. Compounds generated at lower similarity levels generally tend to be larger, because there are more ways of ‘tuning’ feature tree similarity by adding substituents than by replacing them.

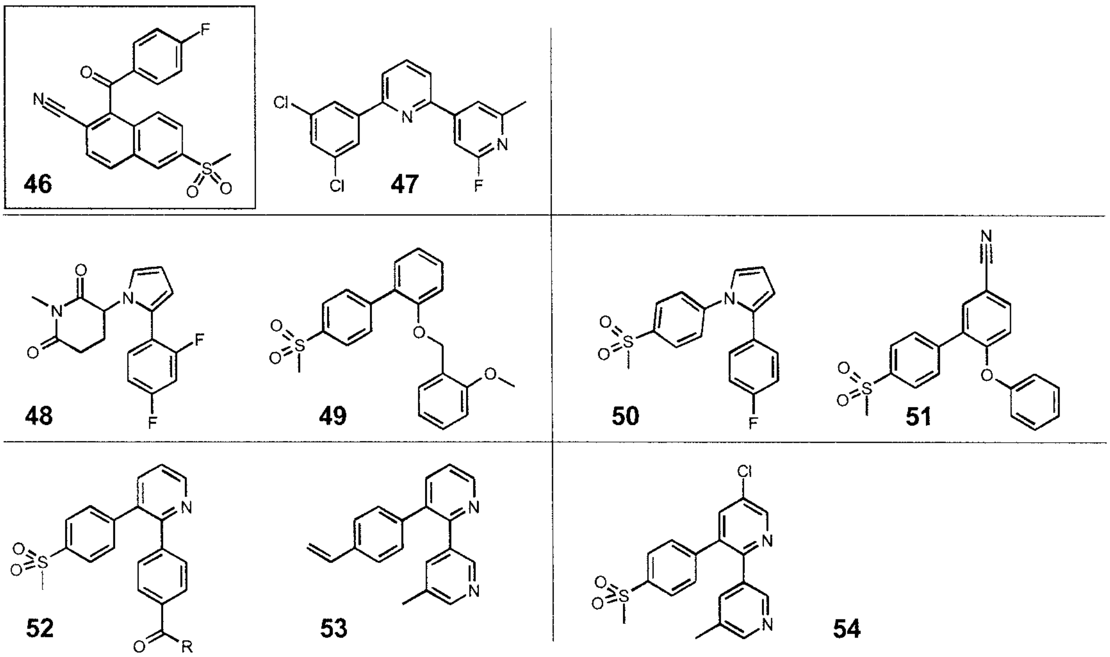


Figure 17. Query results for COX-2 inhibitor **46** that illustrate the use of the MPL-descriptor (details see text). Compounds **50**, **51** and **54** are known inhibitors.

With several D4 antagonists as query structures, the combined chemistry space of WDI and D4 antagonist fragments was searched. For each of the query molecules (left column in Figure 15), 400 compounds were generated at a similarity level of 0.9, again with the restriction that each pair of compounds differ in at least two fragments to increase structural diversity in the solution set. Several generated compounds are shown in Figure 15 together with related known D4 antagonists. Considering the enormous size of the search space, it is not surprising that in no case one of the 55 known antagonists was generated even though all corresponding fragments were available. However, many of the generated compounds are structurally closely related to known antagonists and are therefore very plausible hits.

D4 antagonists share a number of common features that become obvious from looking at the known actives **1–18** in Figures 12 and 15. With one exception, they are linear compounds with aromatic rings at both termini and a tertiary amine group at variable positions along the main chain. The tertiary amine nitrogen atoms are most likely protonated under physiological conditions, but in the query molecules they were kept neutral. The generated compounds also share these properties, with the exception that the nitrogen atom is in two cases (**19**, **20**) bound to an aromatic ring, leading to a lower pK_a value. Such 'flaws' in the designed compounds are beyond the scope of fast similarity searching. It should be noted that within a node in a feature tree, no directionality is stored, i.e., compound **19** and an analog in which the piperidine nitrogen and tertiary carbon atom exchange their positions are identical. With haloperidol **6**, one of the longest known dopamine antagonists, as a query, compounds with a phenylpiperazine moiety (such as **22**) are generated, a common fragment of modern dopamine antagonists. An especially interesting case of 'lead hopping' is observed with query structure **10**. One of the created compounds is **28**, containing a large fragment of the totally unrelated antagonist **18** as a substructure. The chemical features of **28** are arranged such that they match those of the query compound 'pharmacophore', but with a completely different backbone leading to a new chemical series.

Histamine H1 antagonists

While the dopamine D4 searches were performed with a fragment space to which fragments of 55 known antagonists were deliberately added, results discussed in the following were obtained with the 17 000 fragment WDI space alone and thus represent a test of the method *and* the fragment space.

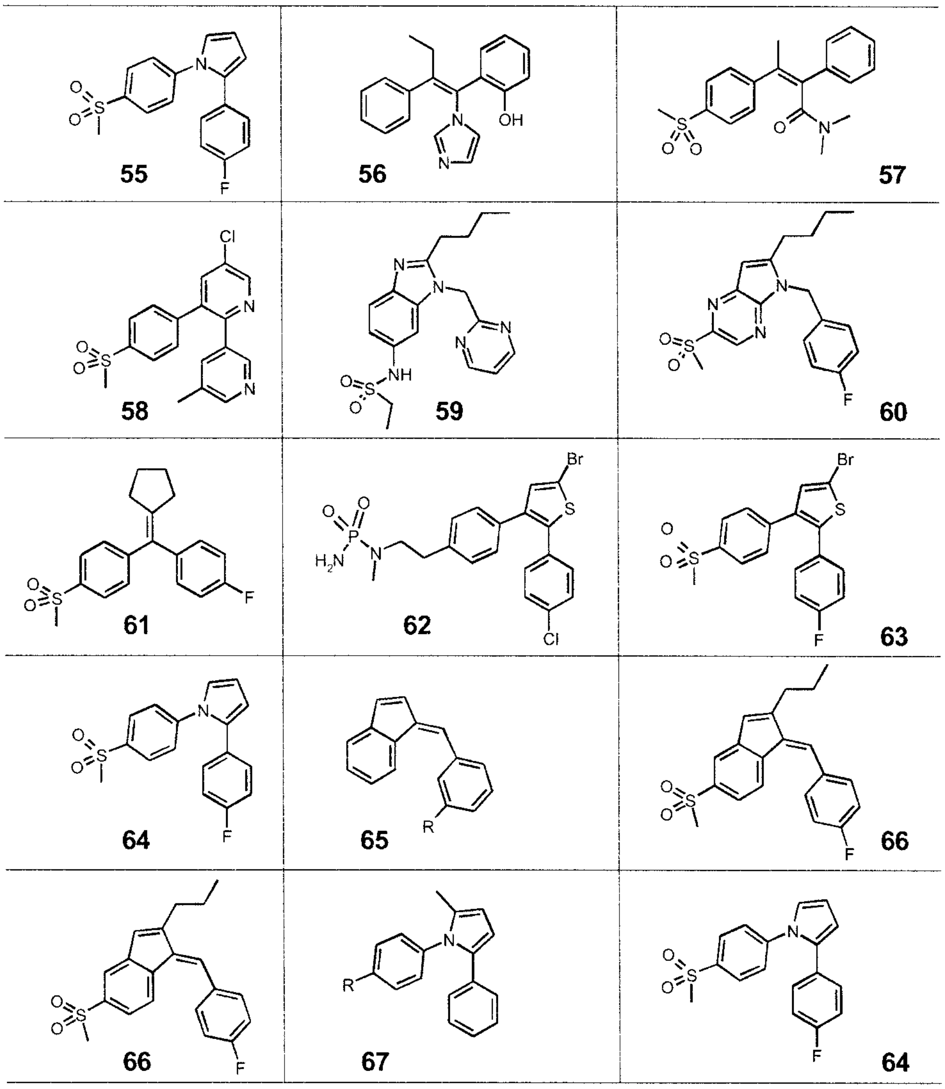


Figure 18. Query results for various COX-2 inhibitors. Left column: query, middle column: plausible hit, right column: known inhibitor related to the plausible hit.

A total number of 55 histamine H1 antagonists was collected from the literature [23]. Five of these compounds were used as queries at a similarity level of 0.9 with the same parameter settings as used in the D4 antagonist study, whereas the remainder served to assess the plausibility of the structures designed by the feature trees software. Representative results are shown in Figure 16. As can be seen, close analogs of different classes of known H1 antagonists are generated. Again, the key to move from one structural class to another is the variation of ring patterns, linker lengths and connectivities. In some cases the algorithm 'dares' to change the overall topology of the molecule as in the step from **31** to **43**, or omit a feature of the query structure as in going from **30** to **41**. In the other examples, similar to the D4 study above, the generated structures fulfill a common pharmacophore in which two aromatic rings and a (positively charged) nitrogen form a triangle. These properties are generally conserved at a similarity level of 0.9.

Cyclooxygenase-2 inhibitors

Figures 17 and 18 show results for a number of cyclooxygenase-2 (COX-2) inhibitors. All inhibitors shown here were retrieved from recently published reviews [24–26]. Many known COX-2 inhibitors are

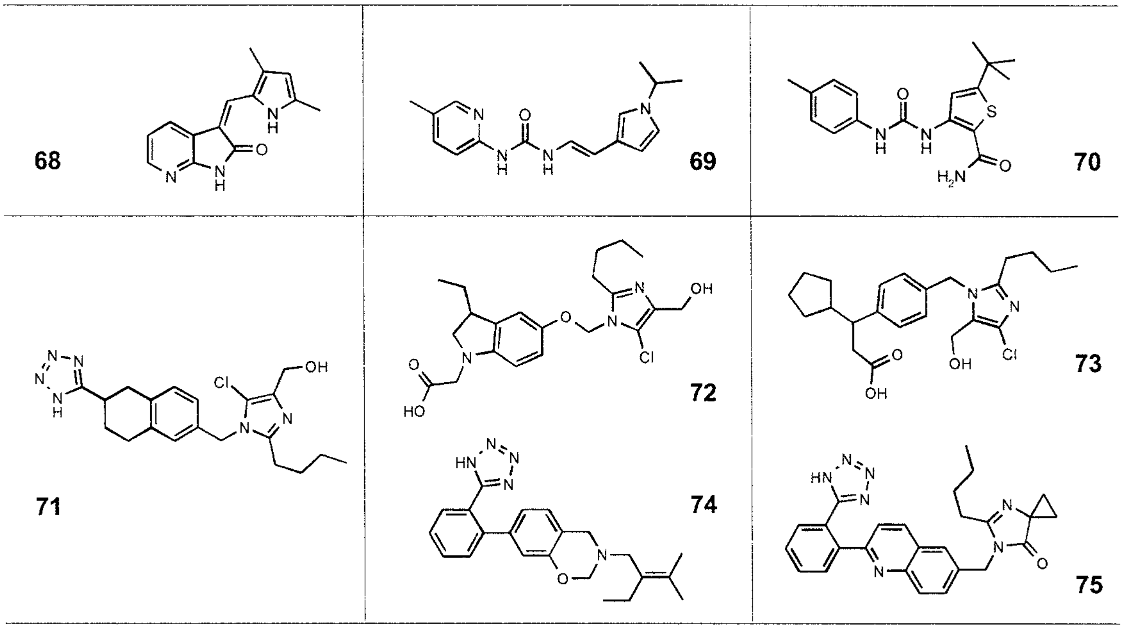


Figure 19. Query results for a tyrosine kinase inhibitor **68** and an angiotensin II antagonist **71**. Left column: query, middle column: plausible hit, right column: known active related to the plausible hit.

1,2-diaryl heterocycles such as **50** or **54**. When used as query structures at similarity levels above 0.9, Ftrees-FS generates alternative compounds of this general type, most of which are known as COX-2 inhibitors. At a similarity level of 0.9, Ftrees-FS can jump between this and other COX-2 inhibitor classes. For example, using compound **46** as a query, the solution set is dominated by compounds of type **47**, which would fall in the diaryl heterocycle class if the substitution pattern was *ortho* rather than *meta* on the central pyridine ring. Knowing only the query structure, one can already deduce that *ortho*-disubstituted compounds would be more likely hits. The form of the query compound can successfully be utilized by incorporating the MPL-descriptor into the similarity value. In doing so, one displaces structures of type **47** from the list of generated compounds. Instead, compounds of type **48** and **49** appear, analogs of which are known inhibitors (**50**, **51**). When a pyridyl fragment with aromatic link atoms in positions 2 and 3 is manually added to the WDI space, compounds **52** and **53** are in the solution set, which are closely related to inhibitor **54**. We have generally found that the path length descriptor leads to little performance increase in standard database ranking, but can be very valuable when employed in fragment space calculations.

Further examples of compounds designed by Ftrees-FS starting from known COX-2 inhibitors are given in Figure 18. All examples demonstrate 'jumps' between the *vic*-diaryl series and other inhibitor classes. For example, starting with **64** Ftrees-FS generates a large set of compounds with the general indene structure **65**. A known COX-2 inhibitor of this class is compound **66**. Conversely, using **66** as a query, one finds back to representatives of the *vic*-diaryl class such as **64**.

More applications: kinase inhibitors, angiotensin II antagonists, serine protase inhibitors

Query results for four more molecules of different biological activity are shown in Figures 19–21. The Indolin-2-one **68** (Figure 19) is a tyrosine kinase inhibitor [27]. Using **68** as a starting structure, Ftrees-FS locates a number of acyclic urea structures such as **69**. Such urea compounds have recently been reported as inhibitors of p38 MAP kinase [28]. Although quite unrelated in their conformational and physicochemical properties, structures **68** and **69** are actually close analogs of each other on feature tree level. The central carbonyl group, which most probably forms a hydrogen bond to the same NH group in the hinge region of kinases, is common to both structures.

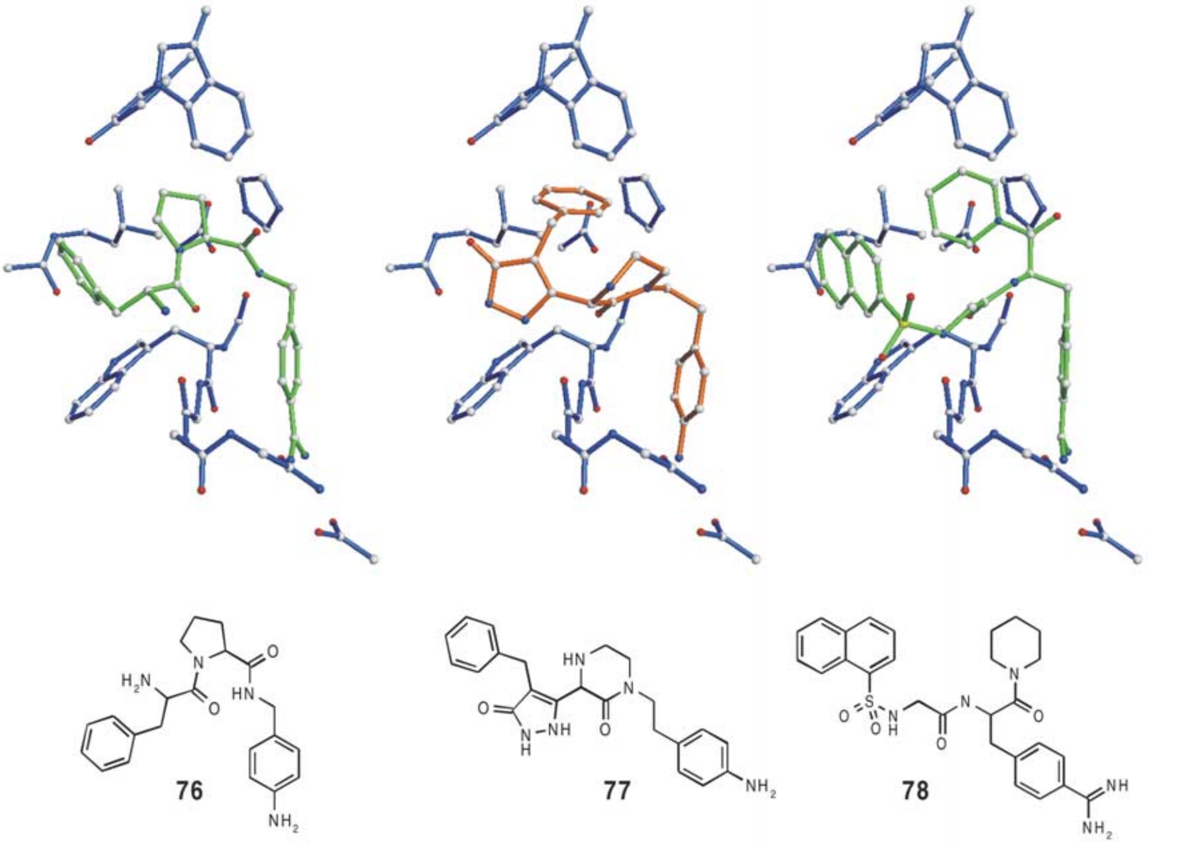


Figure 20. With thrombin inhibitor **76** as a query, compound **77** is generated, which according to modeling studies should display a binding mode related to NAPAP **78** in the S1 region. For modeling purposes, the PDB structure 1ca8 was chosen. The 3D structures of both **76** and **77** are models, whereas the NAPAP conformation was taken from PDB structure 1dwd.

Searches with the angiotensin II antagonist **71** result in compounds **72** and **74**. Compound **72** demonstrates that the feature trees similarity measure is able to recognize a carboxylate group as a substitute of the tetrazole moiety, a property that we have not observed with most other fast chemical similarity measures. Due to space limitations, only compound **73** is shown here as an antagonist related to the designed compound **72**. However, there are several known antagonists of type **73** with varying chain lengths and substitution patterns in the carboxylate region that make **72** indeed a reasonable alternative [29]. Compound **74** demonstrates that the feature tree matching of query and solution molecule does not in all cases reflect a common pharmacophore, but can still lead to new ideas. In the known antagonists **71** and **75**, approximately the same spatial arrangement of the two five-membered heterocycles is achieved with aromatic spacers of different length. The spacer of compound **75** is mimicked by **74**, but the feature tree match overlays the pyrane ring of **74** and the imidazole ring in **71**.

Compounds generated with Ftrees-FS can quickly be evaluated when a target 3D structure is available. An example with thrombin as a target is given in Figure 20. Thrombin inhibitor **76** containing the well-known D-Phe-Pro motif [30] was chosen as the query structure. Visual inspection of the solution set revealed compound **77** as an interesting candidate. It was manually modeled into the active site of thrombin as shown in Figure 20, resembling the X-ray structure of NAPAP **78** complexed with thrombin in the way the S1 moiety is connected to the remainder of the molecule. The step from **76** to compound **77** is analogous to the effort of many groups to reduce the peptidic nature of thrombin inhibitors by cyclization [31].

The prioritization of compounds designed by Ftrees-FS has been done manually in the preceding examples, but the availability of a target 3D structure opens the way for convenient automated analysis

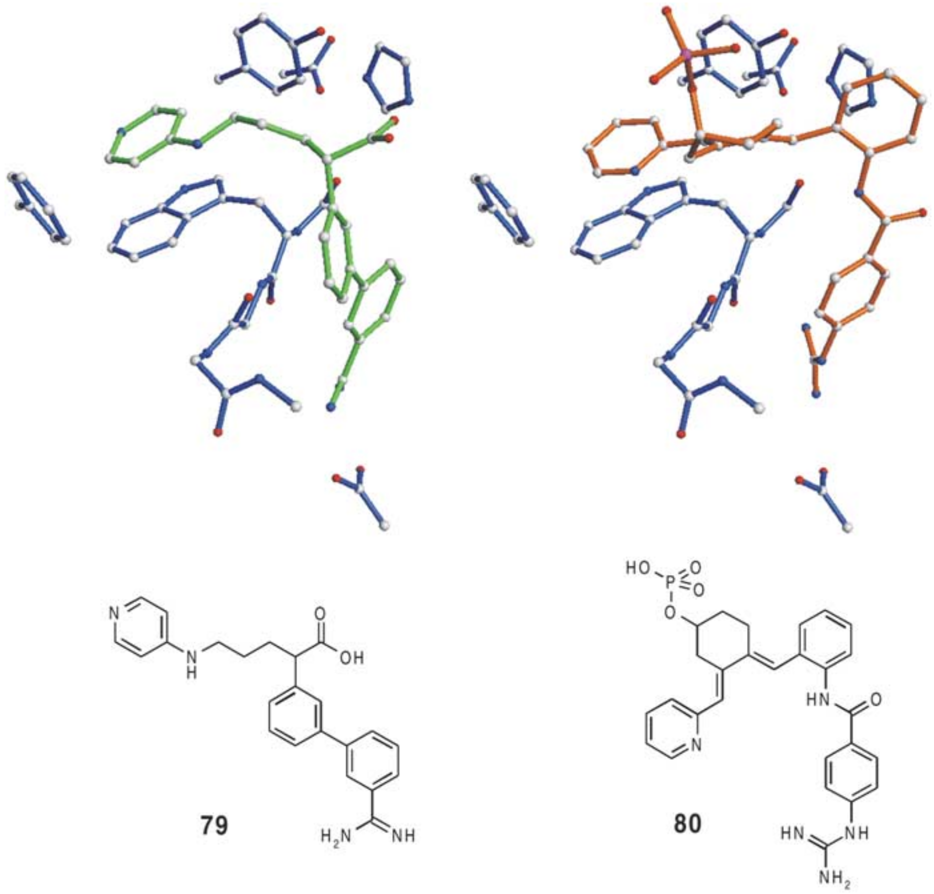


Figure 21. Factor Xa inhibitor **79** (X-ray structure in the complex in green) was used as a query to generate a solution set of 400 compounds with Ftrees-FS, which were subsequently docked into the active site with FlexX. A plausible hit resulting from this study is **80**, whose proposed binding mode (force-field minimized FlexX docking solution) is depicted in brown.

through docking calculations. We illustrate this with the example in Figure 21. The factor Xa inhibitor **79** was extracted from PDB [32] structure 1xka and used as a query compound. All 400 solutions generated by Ftrees-FS were then docked into the factor Xa active site by means of FlexX [19, 20, 33, 34]. For each compound, rank 1 solutions were written to disk and visually assessed. Several compounds displayed binding modes typical of known factor Xa inhibitors [35]: A basic group binding to the S1 pocket, an aromatic ring situated in the aromatic 'cage' formed by Tyr 99, Phe 174 and Trp 215, and an arc-shaped linker between these two groups. One of these compounds is **80**, which is certainly a 'lead hop' away from **79**. Although the structure itself is not a suitable candidate for synthesis (the phosphate group should be replaced by a carboxylate group, the pyridine nitrogen is not at the *para* position), the structure is a good starting point for further modeling studies that could lead to novel factor Xa inhibitors.

Conclusions and outlook

We have presented a novel algorithm for similarity searching in large combinatorial chemistry spaces implemented in the software tool Ftrees-FS. The algorithm is a generic search algorithm in the sense that it is not limited in the number of fragments a solution molecule consists of or the topology in which the fragments are connected to each other. In contrast to most existing methods for similarity searching, the algorithm, based on a dynamic programming scheme, searches within the combinatorial space directly instead of enumerating molecules from the chemistry space. Therefore the algorithm is several orders of magnitude faster than existing approaches for similarity searching. The Ftrees-FS search algorithm is based on the feature tree descriptor that has proven to be a highly effective in the search for similar active compounds in previous applications.

Ftrees-FS is able to exactly extract the molecule that is most similar to a given query. This property may be useful to search for analogs of an active compound that can be synthesized with a specific combinatorial reaction. On the other hand, the concept of target similarity allows to generate diverse sets of compounds at lower similarity levels that can serve as a pool of suggestions in the search for alternative lead compounds. In a thorough validation on several groups of active compounds, we have demonstrated the ability of Ftrees-FS to generate plausible ‘hit’ structures that are structurally distinct from query compounds. The WDI fragment space is an ideal database for such searches, since it implicitly contains the wisdom of several generations of medicinal chemists in synthesizing bioactive compounds. However, it also inherently contains the danger of repetition, which can only be avoided by systematic addition of new alternative fragments.

The set of solution structures generated by Ftrees-FS can either be visually inspected for potential new chemical templates or can be used as input of a docking tool when a target 3D structure is available. This procedure has a number of advantages over traditional *de novo* design based on target 3D structures alone: The search for new structures is guided by a known active compound, problems with bridging various regions of interest in the active site do not occur, the proposed molecules are synthetically feasible, and the process is very fast. If no 3D structure is available, the solution structures could be filtered through a pharmacophore hypothesis to weed out unlikely candidates.

Acknowledgements

MR thanks Roche Bioscience and GMD for funding his research stay in Palo Alto and all colleagues at Roche Bioscience for providing a productive research environment. MS thanks his colleagues at Roche Basel and Roche Bioscience for supporting his sabbatical stay in Palo Alto and for providing the stimulating atmosphere needed to perform this work. We are indebted to Man-Ling Lee and Gisbert Schneider for providing SMILES lists of the WDI RECAP fragments.

Software availability

The fragment space extension to Feature Trees Ftrees-FS described in this paper will be made available as soon as documentation etc. can be provided by the authors. Information about the status can be found at http://cartan.gmd.de/Ftrees.

References

1. Johnson, M.A. and Maggiora, G.M., Concepts and Applications of Molecular Similarity. Wiley, New York, 1990.
2. Dean, P.M., Molecular Similarity in Drug Design. Chapman & Hall, London, 1995.
3. Downs, G.M. and Willett, P., in Lipkowitz, K.B. and Boyd, D.B. (eds), Reviews in Computational Chemistry, Vol. 7. VCH, New York, 1996.
4. Good, A.C. and Mason, J.S., in Lipkowitz, K.B. and Boyd, D.B. (eds), Reviews in Computational Chemistry, Vol. 7. VCH, New York, 1996.
5. Willett, P., J. Chem. Inf. Comput. Sci., 38 (1998) 983.
6. Kubinyi, H., in Kubinyi, H., Folkers, G. and Martin, Y.C. (eds), 3D QSAR in Drug Design: Ligand Protein Interactions and Molecular Similarity, Vol. 9–11. Kluwer/ESCOM, Dordrecht, 1998.
7. Daylight Software Manual, Daylight Inc., Mission Viejo, California, USA.
8. MACCS II, MDL Information Systems Inc., San Leandro, California, USA.
9. Leach, A.R. and Hann, M.M., Drug Discovery Today, 5 (2000) 326.
10. Barnard, J.M., J. Chem. Inf. Comput. Sci., 37 (1997) 59.
11. Lewell, X.Q., Judd, D.B., Watson, S.P. and Hann, M.M., J. Chem. Inf. Comput. Sci., 38 (1998) 511.
12. Schneider, G., Lee, M.-L., Stahl, M. and Schneider, P., J. Comput. Aid. Mol. Des., 14 (2000) 487.
13. Douguet, D., Thoreau, E. and Grassy, G., J. Comput. Aid. Mol. Des., 14 (2000) 449.
14. Weininger, D., J. Chem. Inf. Comput. Sci., 28 (1988) 31.
15. Andrews, K.M. and Cramer, R.D., J. Med. Chem., 43 (2000) 1723.
16. Rarey, M. and Dixon, J.S., J. Comput. Aid. Mol. Des., 12 (1998) 471.
17. Matter, H. and Rarey, M., in Jung, G. (Ed.), Combinatorial Organic Chemistry. Wiley-VCH, New York, NY, 1999.
18. Stahl, M., Rarey, M. and Klebe, G., in Lengauer, T. (Ed.), Bioinformatics: From Genomes to Drugs. VCH, Weinheim, 2000.
19. Rarey, M., Kramer, B., Lengauer, T. and Klebe, G., J. Mol. Biol., 261 (1996) 470.
20. Rarey, M., Wefing, S. and Lengauer, T., J. Comput. -Aid. Mol. Des., 10 (1996) 41.
21. WDI: World Drug Index.
22. Sanner, M.A., Exp. Opin. Ther. Patents, 8 (1998) 383.
23. Aslanian, R. and Piwinski, J.J., Exp. Opin. Ther. Patents, 7 (1997) 201.
24. Carter, J.S., Exp. Opin. Ther. Patents, 8 (1997) 21.
25. Friesen, R.W., Brideau, C., Chan, C.C., Charleson, S., Deschenes, D., Dubé, D., Ethier, D., Fortin, R., Gauthier, J.Y., Girard, Y., Gordon, R., Greig, G.M., Riendau, D., Savoie, C.,

Wang, Z., Wong, E., Visco, D., Xu, L.J. and Young, R.N., Bioorg. Med. Chem. Lett., 8 (1998) 2777.

26. Kalgutkar, A.S., Exp. Opin. Ther. Patents, 9 (1999) 831.

27. García-Echeverría, C., Traxler, P. and Evans, D.B., Med. Res. Rev., 20 (2000) 28.

28. Boehm, J.C. and Adams, J.L., Exp. Opin. Ther. Patents, 10 (2000) 25.

29. Chakravarty, P.K., Exp. Opin. Ther. Patents, 5 (1995) 431.

30. Murray, C.W., Auton, T.R. and Elridge, M.D., J. Comput. Aid. Mol. Des., 12 (1999) 503.

31. Wiley, M.R. and Fisher, M.J., Exp. Opin. Ther. Patents, 7 (1997) 1265.

32. Bernstein, F.C., Koetzle, T.E., Williams, G.J.B., Meyer, J., E. F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. and M., T., J. Mol. Biol., 112 (1977) 535.

33. Rarey, M., Kramer, B. and Lengauer, T., J. Comput. Aid. Mol. Des., 11 (1997) 369.

34. Rarey, M., Kramer, B. and Lengauer, T., Bioinformatics, 15 (1999) 243.

35. Al-Obeidi, F. and Ostrem, J.A., Exp. Opin. Ther. Patents, 9 (1999) 931.