A Family of Ring System-Based Structural Fragments for Use in Structure−Activity Studies: Database Mining and Recursive Partitioning

Ramaswamy Nilakantan,* David S. Nunn, Lynne Greenblatt, Gary Walker, Kevin Haraki,† and Dominick Mobilio

Cheminformatics Group, Wyeth Research, Pearl River, New York 10965

Received November 29, 2005

In earlier work from our laboratory, we have described the use of the ring system and ring scaffold as descriptors. We showed that these descriptors could be used for fast compound clustering, novelty determination, compound acquisition, and combinatorial library design. Here we extend the concept to a whole family of structural descriptors with the ring system as the centerpiece. We show how this simple idea can be used to build powerful search tools for mining chemical databases in useful ways. We have also built recursive partition trees using these fragments as descriptors. We will discuss how these trees can help in analyzing complex structure−activity data.

INTRODUCTION

Descriptors and Their Purpose. Molecular descriptors are used to facilitate quantitative structural comparison between molecules, such as in similarity calculations, SAR studies, etc. Descriptors can be of two broad types—single-value and multiple-value. When a molecule produces a single number as a descriptor characteristic of the entire molecule, it is a single-value descriptor. Examples include molecular weight, calculated logP, various connectivity indexes, and such. Sometimes a set of single-value descriptors is used to identify similar molecules from large databases. These descriptors are also used in structure−activity studies.

Molecular structures can also be decomposed into a set of fragment descriptors. In this case, a single molecule may produce several descriptors, and hence the term multiple-value descriptors. Multiple-value descriptors are typically used in structure−activity studies, similarity and dissimilarity calculations, database mining, and so on. Examples of these descriptors include augmented-atom fragments of Hodes et al.,[1] atom-pairs,[2] and topological torsions[3] from our laboratory, MACCS keys,[4] fingerprints from Daylight Chemical Information Systems,[5] Tripos,[6] BCI,[7] and many others. Willett[8] has reviewed some of the more commonly used descriptors in similarity calculations. Most of the fragment descriptors currently in use are small molecular fragments such as atom-centered fragments, small linear or branched fragments, atom-pairs, ring systems, etc. Thus they are fine-grained descriptors, with a typical molecule producing hundreds of them. Such a fine-grained description is sensitive to small differences in structure and is therefore suitable for similarity and dissimilarity calculations.

In an early paper[9] from our laboratory, we introduced the idea of using the ring system as a descriptor, primarily for database characterization. In later papers,[10−12] we expanded the idea to the ring cluster and ring scaffold and showed how these descriptors can be used for compound acquisition, novelty determination, clustering and browsing, and library design. Similar descriptors have also been proposed by Bemis and Murcko[13,14] and Xu.[15] Roberts et al.[16] have developed a library of predefined, chemically recognizable, structural descriptors for use in structure−activity studies in their Leadscope software. Xu and Johnson[17,18] have developed similar ideas and introduced the term *molecular equivalence number* to describe a set of codes developed from reduced representations of molecules. These equivalence numbers can be used to mine databases for structurally related families of compounds.

The present study builds further on these ideas. Here we extend our original concept and introduce a new family of ‘coarse-grained’ fragment descriptors. They are so-called because they are typically large pieces of molecules centered around ring systems. The purpose of these descriptors is to set up a new framework in which to carry out SAR studies, data mining for active analogues, pharmacophore discovery, etc. The descriptors used in this framework are intended to be chemically recognizable and generally large, fragments.

METHODS

Definitions. We define 12 different types of fragments. These fragments are described below. The abbreviated names of the descriptors are indicated in parentheses. Figure 1 illustrates the definitions with examples.

(1) Ring System (R): This fragment is obtained by separating the molecules into their constituent ring systems. All acyclic single-bonded appendages are dropped, but a single layer of double-bonded appendages is retained. Fused systems such as, for example, naphthalene, are considered to be a single ring system. Each molecule can produce more than one ring system.

(2) Ring Scaffold (RS): This fragment has been described by us in earlier publications.[11,12] It is derived by deleting all acyclic single-bonded appendages on ring systems and linkers connecting the ring systems. A single layer of double-bonded acyclic appendages is retained. All atom and bond types are retained. Each molecule produces a single ring scaffold.

* Corresponding author phone: (914)732-3773; fax: (914)735-3219.
† Retired from Wyeth Research.

10.1021/ci050521b CCC: $33.50 © 2006 American Chemical Society
Published on Web 03/16/2006

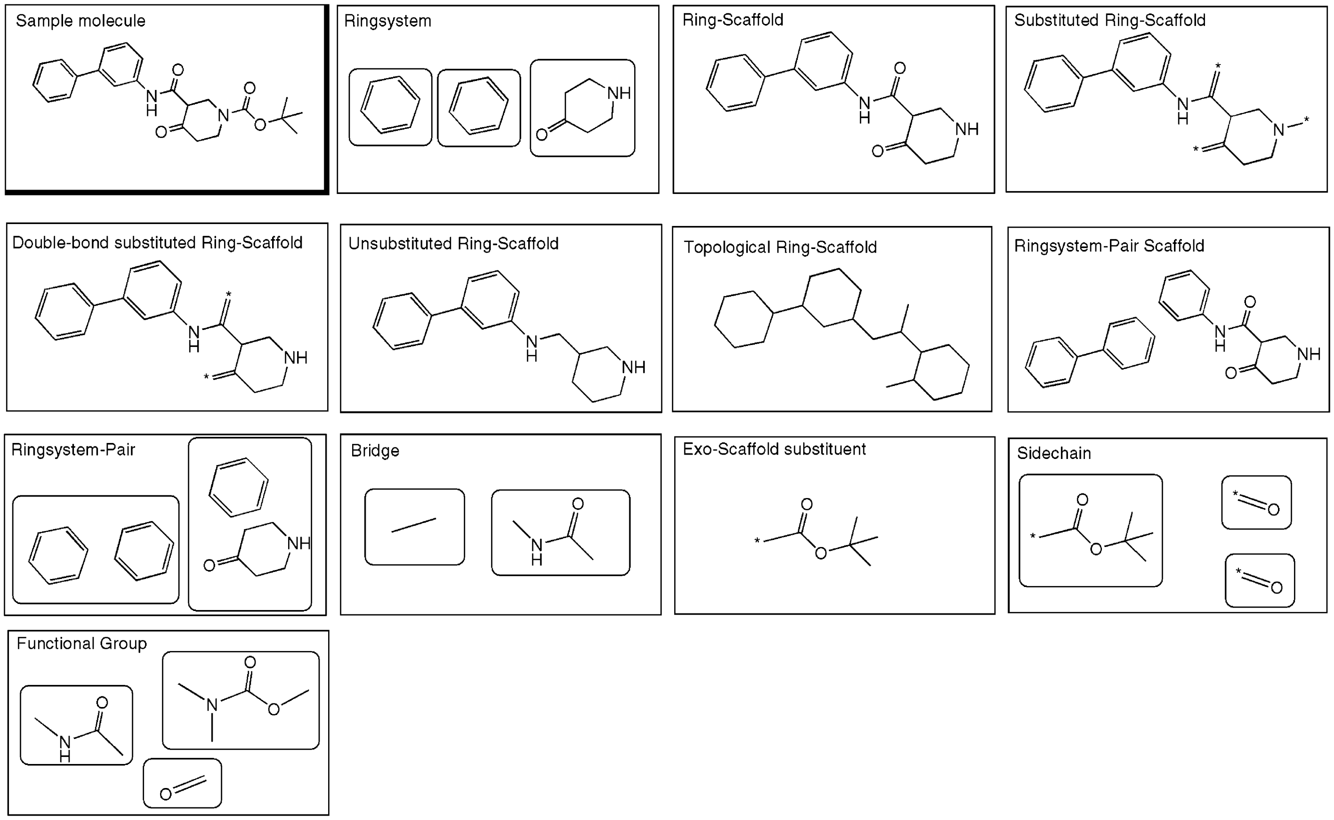

Figure 1. A sample molecule and the different fragments derived from it. This example serves as a schematic definition of the 12 fragment types that we have described in the text.

(3) Substituted Ring Scaffold (SRS): This is a variation of the ring scaffold wherein all substituent positions including double-bonded substituents are indicated by starred pseudoatoms. Each molecule produces a single substituted ring scaffold.

(4) Double-bond-substituted Ring Scaffold (DRS): This variation of the ring scaffold is obtained from the ring scaffold by suppressing the atom-types of double-bonded substituents on the scaffold. Each molecule produces a single double-bond-substituted ring scaffold.

(5) Unsubstituted Ring Scaffold (URS): This variation of the ring scaffold is obtained by dropping all double-bonded acyclic attachments to the scaffold. Each molecule produces a single unsubstituted ring scaffold.

(6) Topological Ring Scaffold (TRS): This fragment is obtained by suppressing all atom and bond-types on the ring scaffold. Each molecule produces a single topological ring scaffold.

(7) Ring System-Pair Scaffold (RPS): This fragment is obtained by fragmenting the molecule into pairs of interconnected ring systems and then applying the definition of the ring scaffold to each pair. Each molecule can produce more than one ring system-pair scaffold. Molecules containing only one ring system produce no RPS fragments. Note that only ring systems connected directly by an acyclic bridge (i.e. without an intervening ring system) are considered.

(8) Ring System-Pair (RP): This fragment is obtained by dropping the linker on the ring system-pair scaffold, leaving two floating ring systems. Each molecule can produce more than one ring system pair. Molecules containing only one ring system produce no RP fragments.

(9) Bridge (B): This fragment is obtained from the ring scaffold by dropping all the atoms except the acyclic linker atoms between pairs of ring systems and the anchor atoms on the ring systems. Each molecule can produce more than one bridge.

(10) Exoscaffold Substituent (X): This fragment is a sort of negative image of the ring scaffold. All the ring scaffold atoms are dropped from the molecule to obtain the exoscaffold substituents. The attachment point of each substituent to the scaffold is indicated by a starred pseudoatom. Each molecule can produce several exoscaffold substituents.

(11) Side chain (SD): This fragment is similar to the exoscaffold substituent fragment. The only difference is that acyclic atoms doubly bonded to a ring atom are considered part of the side chain, rather than part of the ring system. Each molecule can produce several side chain fragments.

(12) Functional Group (FG): This fragment is defined by a set of structural rules.

- Bonds in heteroaromatic rings are retained.

- Bonds to heteroatoms are retained.

- Double and triple (as distinct from aromatic) bonds are retained.

- Single bonds between carbons that also bear double or triple bonds are retained.

- After cleaving all other bonds, isolated carbon atoms are removed.

- Carbons from aromatic rings are relabeled Ar. Thus phenolic OH and aliphatic OH generate different functional groups.

Note that in all the above fragments, stereochemistry is suppressed.

Table 1. Counts of Different Fragment-types in Our Database of Fragments

fragment type abbreviated name	count	fragment type abbreviated name	count
R	17704	RP	27083
RS	136469	B	6702
SRS	262966	X	24924
DRS	133779	SD	25768
URS	122079	FG	42915
TRS	54521	total	937720
RPS	82810		

Calculation and Storage of Fragments and Associated Data. For each fragment we calculate a structure-based unique code (hashcode). This hashcode is designed to provide rapid structure-based matching of fragments. We also calculate a structural complexity number, described in an earlier paper from our laboratory.[9] The complexity is defined as

$$C = B^2 - A^2 + A \qquad (1)$$

where C = complexity, A = number of non-hydrogen atoms, and B = number of bonds.

The actual number stored in the database, C' is calculated as

$$C' = \mathrm{abs}(B^2 - A^2 + A) + H/100 + S/10000 \qquad (2)$$

where H is the number of heteroatoms in the fragment, and S is the number of star atoms (pseudoatoms indicating substitution points).

The first term in eq 2 is a variant of eq 1, modified to handle acyclic structures. It can be easily shown that in the case of acyclics, the first term reduces to the number of bonds B, which is a crude indicator of the size of the fragment. When browsing a large number of fragments, it is convenient to place them in some intuitively reasonable order. We can achieve this by sorting the fragments by complexity.

The fragments and associated data are stored in Oracle databases. One table contains all the unique fragments and their associated structural data. Fragments are identified by their hashcode. For each fragment, identified by its hashcode, we also store its connection table, complexity as defined above, number of heteroatoms, and number of substituents. The latter refers to fragments where connection points to the rest of the molecule are marked. A separate table stores the hashcode and fragment-type of all the fragments in each compound. There is a separate record for each hashcode-fragment-type combination. Each such record contains the compound ID, component number (useful for multicomponent structures), the total number of components, hashcode, fragment type, fragment count (for multiple occurrences), SMILES[19] string, and molecule fraction. The molecule fraction is the number of non-hydrogen atoms in the fragment divided by the total number of non-hydrogen atoms in the molecule.

We have calculated fragments for our entire high-throughput screening set and for additional compounds that have sufficient quantity of available sample but have not been plated yet. Currently, our fragment database contains 732 103 distinct fragments and 937 720 different fragment−fragment-type combinations. Table 1 shows the details of the counts of the individual fragment-types.

RESULTS AND DISCUSSION

Coarse-Grained Similarity Searching. The precalculated descriptors can be used to identify analogues that are structurally similar to a given probe molecule by simply fetching all compounds that share a fragment with the probe. Similarity search done in this way is somewhat different from traditional similarity search. Here we use large chemically recognizable fragments as descriptors instead of much smaller, often chemically indeterminate fragments. We also do not calculate a numerical measure of similarity. All compounds that share a fragment with a probe molecule are assumed to be similar to it. We have already described this type of search using the ring system,[9] ring cluster,[10] and ring scaffold.[11] This can obviously be extended to the variants of the ring scaffold described above (viz., topological ring scaffold, substituted ring scaffold, double-bond substituted ring scaffold, unsubstituted ring scaffold), the ring system-pair scaffold, and various judiciously chosen combinations of these fragments. A single carefully constructed search can often substitute for a large number of complex substructure searches or even exceed the scope of any substructure search.

Fragment-based searches work very well when, for example, we use the ring scaffold as the descriptor and the scaffold dominates the molecule and its analogues. On the other hand, in simpler compounds, such as compounds with a single benzene ring (and no other rings), the scaffold does not always dominate the compound. The assertion that all compounds containing a benzene ring are similar to each other leads to somewhat unsatisfying results. Figure 2 shows examples of both these situations. The important caveat to keep in mind is that fragment-based database mining does not always work satisfactorily. It all depends on the structure of the probe and the contents of the database. However, when the method works, it can produce interesting results not obtained by traditional methods.

We describe below an example of how fragment-based searching could be used to identify analogues not easily discovered by conventional methods. This is a retrospective study on a set of compounds tested for their binding affinity to the $5HT_{1A}$ receptor. Since this is only for illustration, we restricted our search to the above set of compounds. We picked one of the more potent compounds as the probe and searched for other compounds that have the same topological ring scaffold (TRS). Figure 3 shows the probe compound, the equivalent topological ring scaffold, and some selected hits. All but the last compound met the criteria for potency. An examination of the structures shows that the hits span a variety of different ring system arrangements and types. Figure 4 shows an analysis of the hits. The four ring systems are labeled A through D. As can be seen from the figure, there is a single variant of A, three variants of B, four variants of C, and five variants of D. Also, it can be seen that pyridine at the D position is connected in three different ways, 2-, 3-, and 4-, in different analogues. There are also five variants of the bridge linking the D position, viz. amide, *N*-methylated amide, amidine, urea, and ester. The important point is that this method of identifying analogues has the potential to identify new scaffolds with new ring systems and new arrangements thereof. It should also be noted that some of

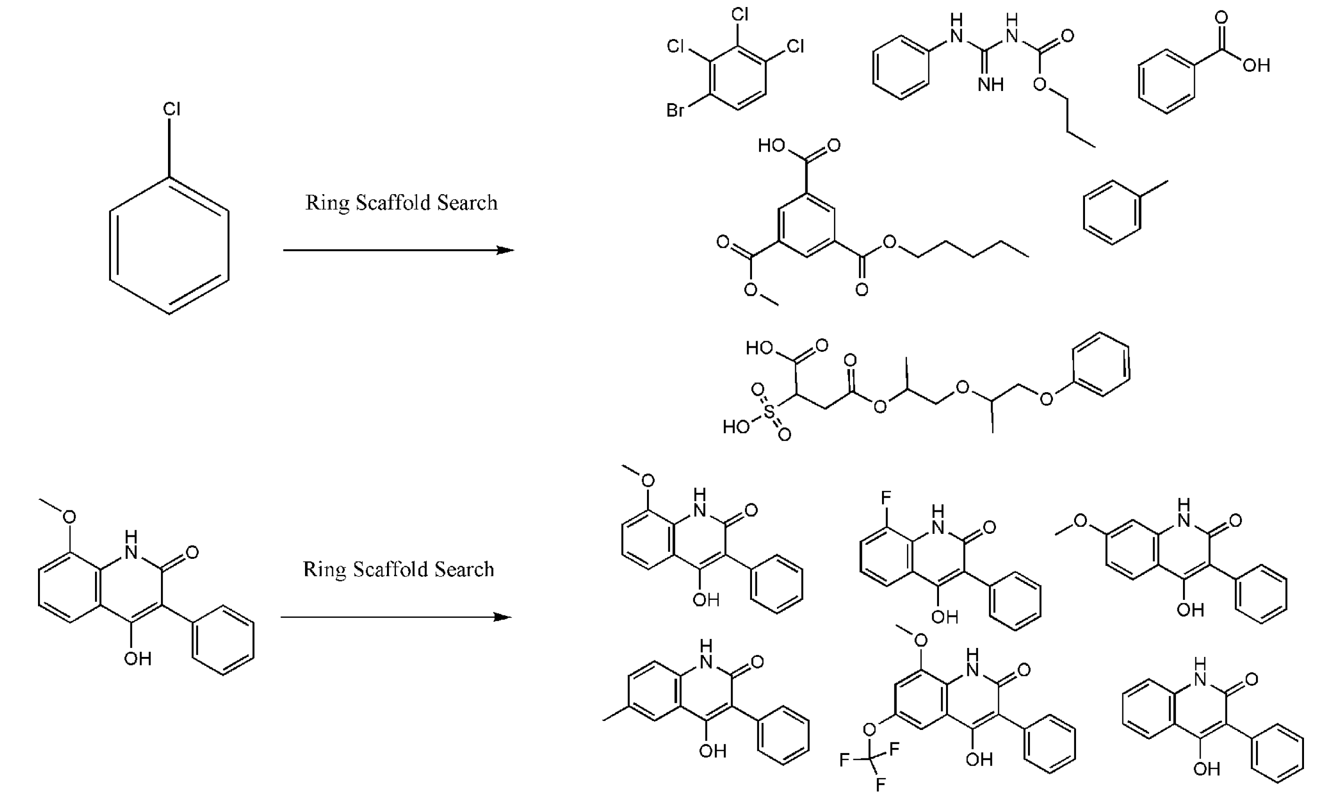

Figure 2. Two examples of a search for compounds with the same ring scaffold as the probe molecule. In the first case (top), the probe scaffold is a simple benzene ring, and the resulting hits are structurally quite diverse. In the example on the bottom, since the scaffold is not a simple one-ring structure, the resulting hits are not so structurally diverse and might be thought of as members of one chemical series.

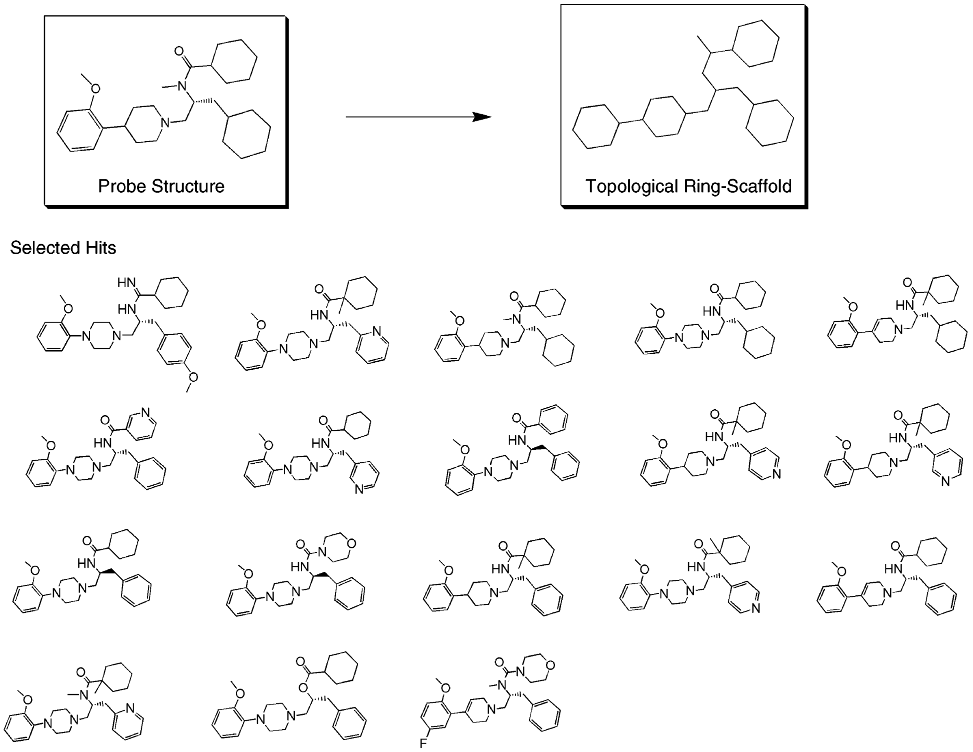

Figure 3. This figure shows a database mining example where a search is carried out for compounds with the same topological ring scaffold as the probe molecule. The probe molecule, its topological ring scaffold, and 18 hits (including the probe itself) are shown.

the analogues are not very similar to the probe as perceived by conventional similarity search methods. Table 2 shows the similarity scores of the analogues in Figure 3 against the probe molecule. Three different methods were used to calculate the similarity, atom-pair,[2] topological torsion,[3] and a fingerprint method using Tripos UNITY fingerprints.[6] There are 6 hits with an atom-pair similarity score of <60 (a reasonable lower cutoff for the atom-pair descriptor), 11 hits with a topological torsion similarity score of <55 (a reasonable cutoff for the topological torsion descriptor), and 7 hits with a fingerprint similarity score of <70 (a reasonable cutoff for the fingerprint descriptor).

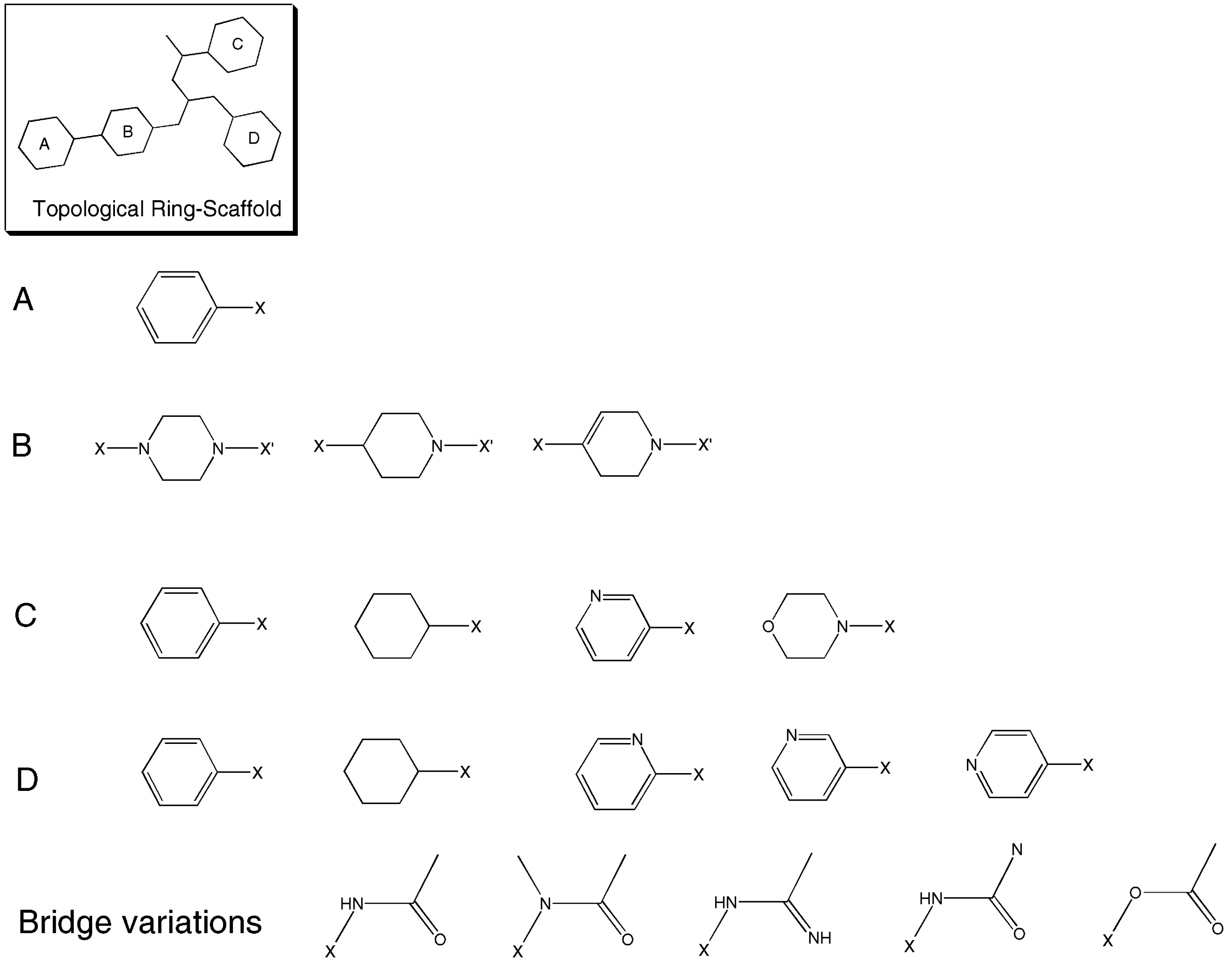

Figure 4. This figure shows an analysis of the hits obtained in the topological ring scaffold search shown in Figure 3. The four ring systems are labeled A through D. It can be seen that there is a single variant of ring A, three variants of ring B, four of ring C, five of ring D, and five variations of the inter-ring system bridge.

Table 2. Similarity Values (on a 0−100) Scale of the Probe Molecule to All the Hits from the Topological Ring Scaffold Search Shown in Figure 3[a]

s.no	atom-pair	topological torsion	fingerprint
1	**58.2**	**38.1**	**65.8**
2	67	57.1	**68.6**
3	74.8	**44.4**	83.7
4	80.5	**54.1**	70.8
5 (probe)	100	100	100
6	**40.8**	**32.8**	**65**
7	62.9	**41**	71.1
8	**40.8**	**32.8**	**67.5**
9	67.6	60.3	82.3
10	68.2	60.3	80.6
11	63.1	**41**	75.5
12	**55.5**	**31.1**	**62**
13	68.4	60.3	85.1
14	66.5	57.1	73.4
15	65	65.6	96.8
16	61.3	**43.1**	**67.8**
17	**58.8**	**34.4**	**62.4**
18	**51.4**	**43.8**	79.6

[a] The similarity values shown in boldface are those that fall below the threshold commonly used for that descriptor, 60 for atom-pair, 55 for topological torsion, and 70 for fingerprint. Note that these cutoffs have been derived empirically from experience.

It is important to understand the limitations of this type of coarse-grained search. Some topological ring scaffolds are extremely general and therefore very common. For example, the TRS for a molecule containing just one six-membered ring system will hit all molecules containing exactly one six-membered ring system, regardless of the atom-types and hybridization, generally not a very useful list. If there are two or more ring systems in the molecule, and the bridges interconnecting them are not extremely common, there will be fewer hits, and there is the potential of finding some interesting analogues.

The above example illustrates the usefulness of this type of search. While these results could have been obtained by a properly formulated substructure search, it is unlikely that one would run the required search without advance knowledge of the contents of the database. The use of a fragment database eliminates the need for designing a substructure search query, a potentially time-consuming task. It is also possible to make creative use of multiple fragment types to identify novel analogues. For instance, one could query the database to find combinations of ring systems, or ring system pairs, or particular bridge-ring system combinations, and so on.

Building a Recursive Partition Tree with Coarse-Grained Descriptors. It is possible to analyze structure−activity relationships using so-called recursive partition (RP) trees. These are ordered structures where sets of compounds that are structurally related and have similar biological activities are clustered together. It is important to note that such trees are generally not perfect, i.e., compounds that are not structurally or biologically too similar may also end up together. This is because the relationship between structure and biological activity is usually extremely complex and there is no perfect structure−activity model.

Recursive partitioning is a procedure wherein a set of compounds (or objects in general) is successively split or partitioned into two sets—those that have a descriptor and those that do not. This is done recursively until some chosen stopping criterion is reached, or we reach down to the individual compounds. The descriptor that is chosen for each split could be decided based on a variety of criteria. The general guideline is that the final tree should be as ordered as possible, that is, similar compounds with similar activities should tend to cluster together.

The use of decision trees in studying structure−activity is not new. Hawkins et al.[20] and Rusinko et al.[21] reported on the use of RP to analyze SAR in large heterogeneous compound sets. The basic approach has been modified in many different ways in different laboratories (van Rhee et al.,[22] van Rhee,[23] Godden et al.,[24] Miller,[25] Blower et al.,[26] Cho et al.,[27] DeLisle and Dixon[28]). Refinements include different descriptors, splitting criteria, combining RP with methods such as simulated annealing,[26] evolutionary programming,[28] etc. In the pharmaceutical area, recursive partitioning has been used mostly for predictive modeling of the biological activity of compounds, (i.e., use a training data set to create the tree and then use the tree to make predictions outside the training set). However, some groups have used RP for other purposes such as automatic identification of pharmacophores (Chen et al.[29]), discriminating drugs and nondrugs (Wagener and Geerestein[30]), etc.

In our implementation of RP, we used the entire set of fragments stored in our Oracle database as descriptors. In the present version, we handle only binary biological data, i.e., compounds are either active or inactive. We first place the entire set of compounds at the root and find the best fragment to split them into two sets, viz. those that contain the descriptor and those that do not. The choice of splitting fragment is made by a ‘minimum entropy’ method. The average entropy associated with a particular split is calculated by using the Shannon entropy formula and is given by the expression below.

$$\text{average entropy} = \sum_b \left(\frac{n_b}{n_t}\right) \times \left[\sum_c -\left(\frac{n_{bc}}{n_b}\right) \log_2 \left(\frac{n_{bc}}{n_b}\right)\right]$$

where n_b = number of instances in branch b, n_t = total number of instances in all branches, and n_{bc} = number of instances in branch b of class c.

The above general expression can be interpreted in our particular situation as

$$\text{average entropy} = \frac{h_t}{t} \times \left[\left(\frac{A_h}{h_t}\right) \log_2 \left(\frac{A_h}{h_t}\right) - \left(\frac{I_h}{h_t}\right) \log_2 \left(\frac{I_h}{h_t}\right)\right] + \frac{n_t}{t} \times \left[-\left(\frac{A_n}{n_t}\right) \log_2 \left(\frac{A_n}{n_t}\right) - \left(\frac{I_n}{n_t}\right) \log_2 \left(\frac{I_n}{n_t}\right)\right]$$

where t = the total number of compounds at the node being split, h_t = the total number of compounds containing the fragment, n_t = the total number of compounds not containing the fragment, A_h = the total number of active compounds containing the fragment, I_h = the total number of inactive compounds containing the fragment, A_n = the total number of active compounds not containing the fragment, and I_n = the total number of inactive compounds not containing the fragment.

The idea here is to create a split such that compounds are more ordered (lower entropy), or, in other words, similar compounds with similar activities are clustered together.

The splitting continues until either (a) there are no more descriptors to split the compound set by, or (b) there are 10 or fewer compounds in the set. These terminal nodes where no further splitting occurs are the so-called ‘leaves’ of the tree. We chose the number 10 somewhat arbitrarily, but the idea is that it is small enough that a chemist can visually examine and analyze the compound set easily without any further splitting. Leaves containing 40% or more of active compounds are defined, again somewhat arbitrarily, as ‘significant’ leaves. We have written a Web application for generation and display of the tree and its contents; details are given in the Appendix. The application is designed for the use of medicinal chemists or molecular modelers seeking to perceive patterns or establish structure−activity rules. They can run the application from their desktop and browse through the leaves to get a quick idea of structure−activity relationships in the compound set. We illustrate this with an example.

We applied the RP method to a set of ∼3000 compounds evaluated in a $5HT_{1A}$ receptor binding assay as described by Childers et al.[31] Compounds whose K_i value met a predetermined threshold for potency were assigned a score of 1, and all others 0.

We ran the RP program and obtained the tree shown in Figure 5. There were a total of 90 significant leaves. The contents of one significant leaf are shown in Figure 6. At the top of the figure, the fragments common to every compound in the leaf are shown. These are taken from the splits along the path starting from the root of the tree to the leaf. Beneath the picture of each fragment, a ratio of the form *a/b* is given, where *b* is the total number of compounds containing the fragment and *a* is the number of active compounds containing the fragment. This ratio refers to the entire set of compounds used in the generation of the tree and gives the user an idea, albeit only qualitative, of the importance of that fragment in determining biological activity. Next, the structures of the compounds in the leaf are shown along with their activities in parentheses (*A* for active and *I* for inactive). Also shown is a diversity number (on a scale of 0 to 1) of the set of compounds in the leaf. This is calculated as 1 minus the mean pairwise similarity of the entire compound set. We employ Tripos fingerprints[6] as descriptors and use a fast procedure due to Turner et al.[32] to calculate this diversity value. At the bottom are shown pictures of fragments that are not present in any compound in the leaf. Again, these are taken from the splits along the path starting from the root to the leaf. Thus, at a glance the user can get an idea of the structure−activity relationships among the compounds in the leaf along with the structural rules. In the example shown, there are two compounds in

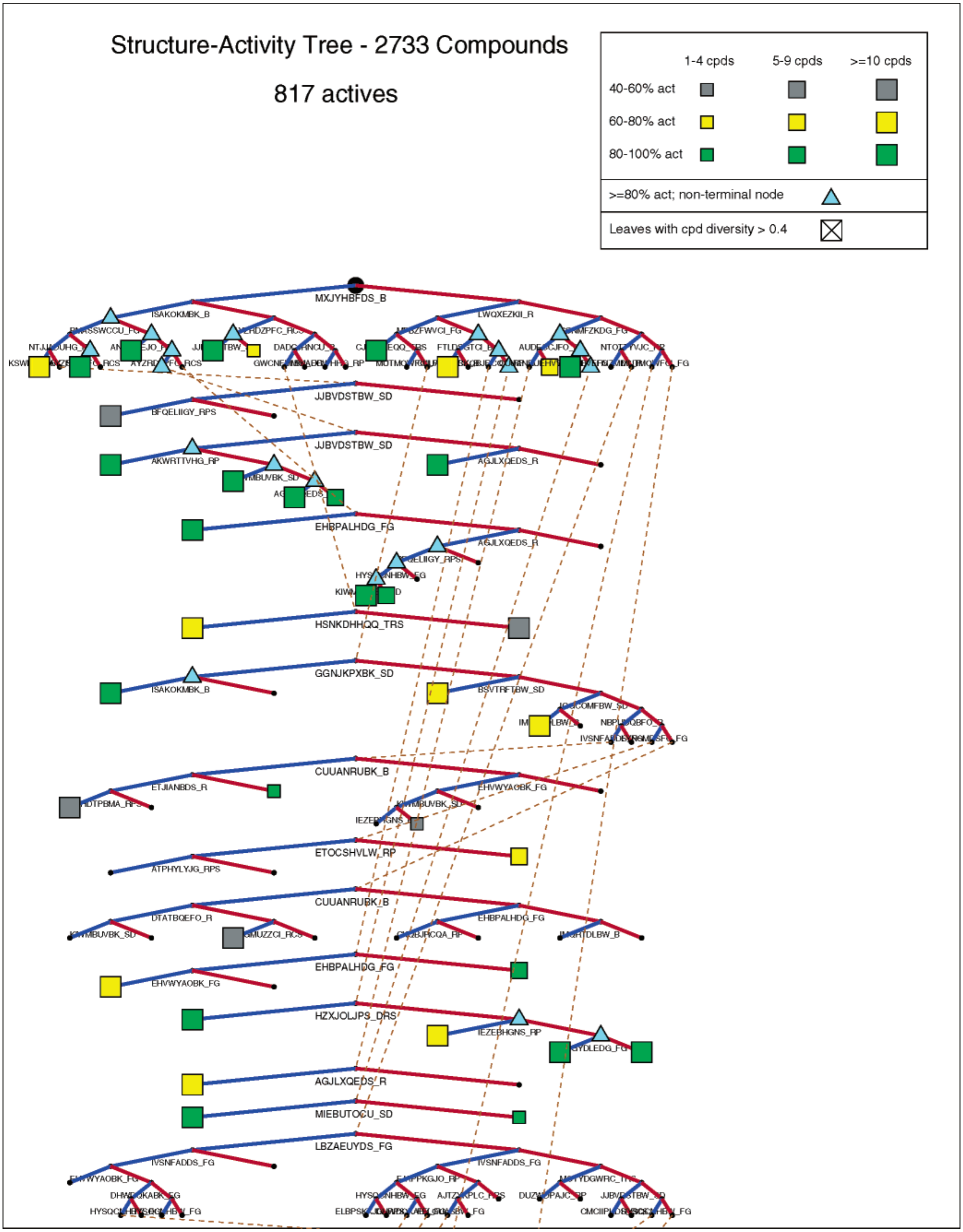

Figure 5. The recursive partition tree as drawn by our Web application. Note that the lower parts of the tree are not seen in the figure. Details of how to interpret the tree are given in the Appendix.

the leaf that do not meet the potency criteria. In the case of one of them (the last compound in the leaf), one might postulate that the bulky substituent attached to the indole nitrogen might be the reason for diminished potency. It is not obvious why the other compound is less potent. It could be a complex SAR involving interacting effects of different functional groups. In any case, the point is that by clustering compounds with similar structures and activities together in

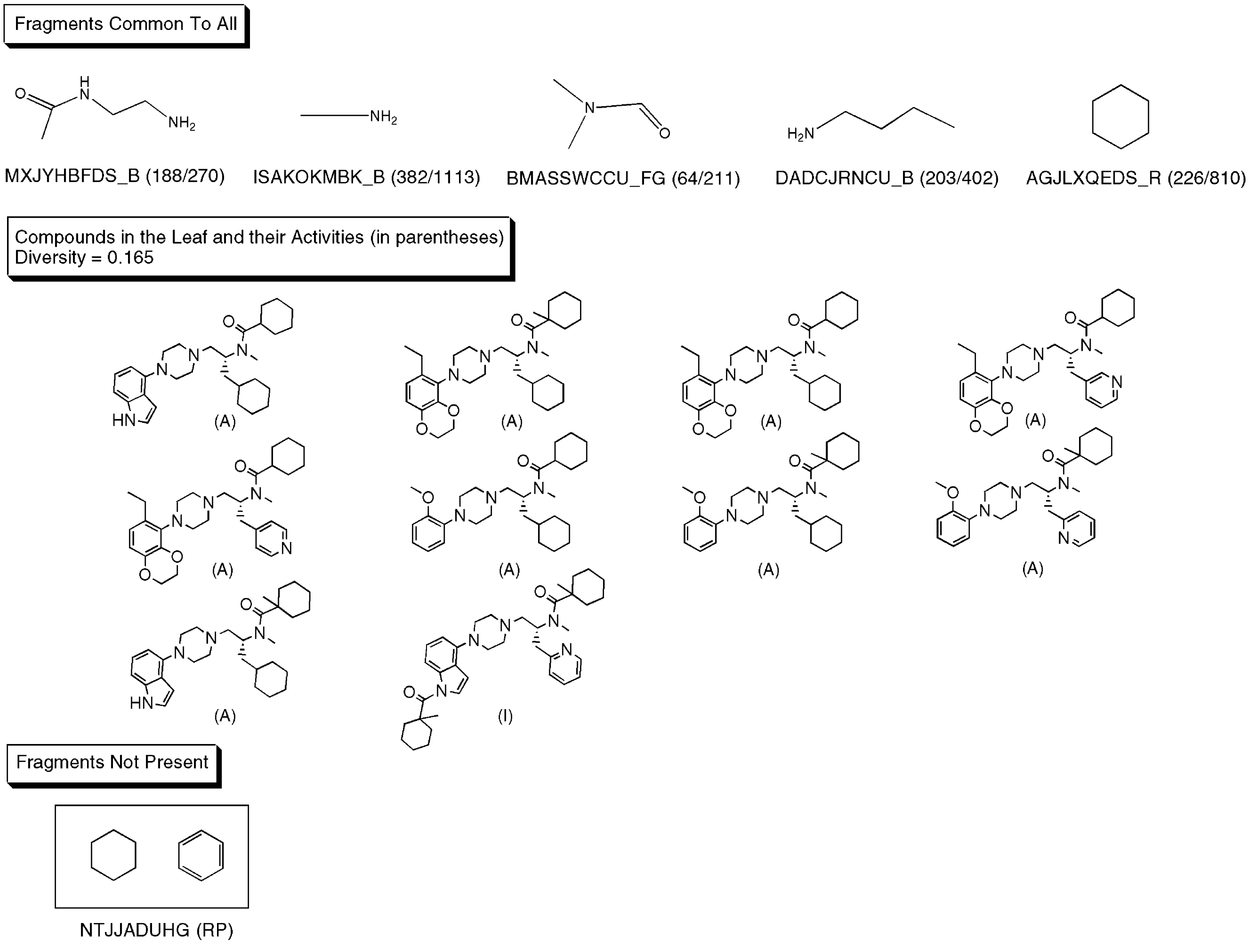

Figure 6. A schematic of the contents of one significant leaf are shown. The details are described in the text.

a leaf, it is possible for the chemist to make testable structure−activity hypotheses.

Advantages and Limitations of the Descriptor Set. Our set of descriptors, chosen to be chemically recognizable, are useful in interpreting the results. On the other hand, these descriptors, which are coarse grained, might miss subtle changes. With experience, these descriptors could be fine-tuned or expanded to suit our needs.

Advantages and Limitations of Recursive Partitioning. The results of a recursive partitioning analysis reflect only the input data set, and any generalizations outside of the data set are speculative. However, this is not unlike any other SAR method. RP like many other methods is unstable. Small changes to the input data can (though not necessarily) drastically alter the tree. Recursive partitioning done with our fragments have the advantage that chemists easily understand structure−activity relationships expressed in terms of these fragments.

CONCLUSION

We have presented a family of molecular fragments as a descriptor set to be used for data mining and structure−activity analysis. These fragments are chemically recognizable and easy to interpret. We have shown an example of data-mining using these fragments. We have also developed a recursive partitioning procedure using these fragments as descriptors. Our results show that these fragments are useful in the development, analysis, and interpretation of structure−activity ideas.

APPENDIX

The tree generation and display program is written in Perl and is delivered as a Web application. The user can browse in a data file containing compound identifiers and active/inactive (1/0) information and request a tree to be built. The program builds the tree as described above and outputs the results as a gif image of a tree. In the tree diagram, the significant leaves are shown as squares of three different colors and three different sizes. The encoding scheme is as follows.

small → 1−4 compounds

medium → 5−9 compounds

large → > = 10 compounds

green → > = 80% of the compounds are active

yellow → > = 60% and <80% of the compounds are active

gray → > = 40% and <60% of the compounds are active

As the tree gets larger and deeper, more nodes need to be drawn at the same tree depth. For example, if every node split into two nodes, at the 10th level we would need to draw 1024 nodes, a very crowded situation. To overcome this problem, every time a branch gets more than five levels deep, we relocate that node to the bottom center of the growing tree and indicate its new position by a dashed line. Another alternative, not implemented at this time, would be to enable zooming in on selected parts of the tree.

The squares representing significant leaves are hyperlinked to information on the content of the leaves. Specifically, by clicking on a significant leaf, we get the structures of all the compounds in the leaf and the structures of all the fragments involved in the splits leading up to that leaf. The activities of the compounds (0 or 1) are also indicated explicitly. Furthermore, the actives are shown with a black background, while the inactives are shown with a white background. The fragments are divided into two groups, those that *every compound* in the leaf contains, and those that *no compound* in the set contains, representing the left and right sides of the splits. Along with each fragment the number of actives containing that fragment and the total number of compounds containing that fragment are shown. This gives us an idea of the strength of association between the fragment and biological activity. Also shown is a diversity number, which is calculated as 1 minus the sum of all pairwise similarities of the compounds in the leaf. This calculation is done using a method due to Turner et al.,[30] which has a linear rather than quadratic runtime dependency on the number of compounds. Leaves with a diversity value of >0.4 are marked with a black X in the tree diagram, indicating to the user that these leaves are perhaps too heterogeneous to be of interest. Thus, a glance at the contents of a leaf clearly shows the user the following:

1. The compounds in the leaf.
2. The set of fragments (rules) that placed the compounds in the leaf.
3. The set of excluded fragments.
4. The structural homogeneity of that compound set.
5. The association of the fragments in the leaf with activity in the entire compound set.

REFERENCES AND NOTES

(1) Hodes, L.; Hazard, G. F.; Geran, R. I.; Richman, S. A Statistical Heuristic Method for Automated Selection of Drugs for Screning. *J. Med. Chem.* **1977**, *20*, 469−475.

(2) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom-Pairs as Molecular Features in Structure−Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64−73.

(3) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsion: A New Molecular Descriptor for SAR Applications: Comparison with Other Descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82−85.

(4) MACCS keys are a product of Elsevier MDL, San Leandro, CA.

(5) James, C. A.; Weininger, D.; Delany, J. In *Daylight Theory: Fingerprints*; Daylight Chemical Information Systems Inc.

(6) UNITY fingerprints are a product of Tripos Inc.: St. Louis, MO.

(7) BCI fingerprints are a product of Barnard Chemical Information Systems Ltd. Sheffield, U.K.

(8) Willett, R. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.

(9) Nilakantan, R.; Bauman, N.; Haraki, K. S.; Venkataraghavan, R. A Ring-Based Structural Query System: Use of a Novel Ring-Complexity Heuristic. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 65−68.

(10) Nilakantan, R.; Bauman, N.; Haraki, K. S. Database Diversity assessment: New Ideas, Concepts and Tools. *J. Comput-Aided Mol. Des.* **1997**, *11*, 447−452.

(11) Nilakantan, R.; Immermann, N.; Haraki, K. S. A Novel Approach to Combinatorial Library Design. *Comb. Chem. High-Throughput Screening* **2002**, *5*, 105−110.

(12) Nilakantan, R.; Nunn, D. S. A Fresh Look at Pharmaceutical Screening Library Design. *Drug Discovery Today* **2003**, *8*, 668−672.

(13) Bemis, G. W.; Murcko, M. A. Properties of Known Drugs: Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.

(14) Bemis, G. W.; Murcko, M. A. Properties of Known Drugs: Sidechains. *J. Med. Chem.* **1999**, *42*, 5095−5099.

(15) Xu, J. A New Approach to Finding Natural Chemical Structure Classes. *J. Med. Chem.* **2002**, *45*, 5311−5320.

(16) Roberts, R.; Myatt, G. J.; Johnson, W. P.; Cross, K. P.; Blower Jr., P. E. LeadScope: Software for Exploring Large Sets of Screening Data. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1302−1314.

(17) Xu, Y.-J.; Johnson, M. Algorithm for Naming Molecular Equivalence Classes Represented by Labeled Pseudographs. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 181−185.

(18) Xu, Y.-J.; Johnson, M. Using Molecular Equivalence Numbers to Visually Explore Structural Features that Distinguish Chemical Libraries. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 912−926.

(19) Weininger, D. SMILES, A Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31.

(20) Hawkins, D. M.; Young, S. S.; Rusinko III, A. Analysis of a Large Structure−Activity Data Set using Recursive Partitioning. *Quant. Struct.-Act. Relat.* **1997**, *16*, 296−302.

(21) Rusinko III, A.; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017−1026.

(22) van Rhee, A. M.; Stocker, J.; Printzenhoff, D.; Creech, C.; Wagoner, P. K.; Spear, K. L. Retrospective Analysis of an Experimental High-Throughput Screening Data Set by Recursive Partitioning. *J. Comb. Chem.* **2001**, *3*, 267−277.

(23) van Rhee, A. M. Use of Recursion Forests in the Sequential Screening Process: Consensus Selection by Multiple Recursion Trees. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 941−948.

(24) Godden, J. W.; Furr, J. R.; Bajorath, J. Recursive Median Partitioning for Virtual Screening of Large Databases. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 182−188.

(25) Miller, D. W. Results of a New Classification Algorithm Combining K Nearest Neighbors and Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 168−175.

(26) Blower, P.; Fligner, M.; Verducci, J.; Bjoraker, J. On Combining Recursive Partitioning and Simulated Annealing to Detect Groups of Biologically Active Compounds. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 393−404.

(27) Cho, S. J.; Shen, C. F.; Hermsmeier, M. A.. Binary Formal Inference-Based Recursive Modeling using Multiple Atom and Physicochemical Property Class Pair and Torsion Descriptors as Decision Criteria. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 668−680.

(28) DeLisle, R. K.; Dixon, S. L. Induction of Decision Trees via Evolutionary Programming. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 862−870.

(29) Chen, X.; Rusinko III, A.; Tropsha, A.; Young, S. S. Automated Pharmacophore Identification for Large Chemical Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 887−896.

(30) Wagener, M.; van Geerestein, V. J. Potential Drugs and Nondrugs: Prediction and Identification of Important Structural Features. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 280−292.

(31) Childers, W. E., Jr.; Abou-Gharbia, M. A.; Kelly, M. G.; Andree, T. H.; Harrison, B. L.; Ho, D. M.; Hornby, G.; Huryn, D. M.; Rosenzweig-Lipson, S. J.; Schmid, J.; Smith, D. L.; Sukoff, S. J.; Zhang, G.; Schechter, L. *J. Med. Chem.* **2005**, *48*, 3467−3470.

(32) Turner, D. B.; Tyrrell, S. M.; Willett, P. Rapid Quantification of Molecular Diversity for Selective Database Acquisition. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 18−22.

CI050521B