Binary Formal Inference-Based Recursive Modeling Using Multiple Atom and Physicochemical Property Class Pair and Torsion Descriptors as Decision Criteria

Sung Jin Cho,*,† C. Frank Shen,‡ and Mark A. Hermsmeier§

Combinatorial Drug Discovery, Bristol-Myers Squibb Company, 5 Research Parkway, Wallingford, Connecticut 06492-7660, Nonclinical Biostatistics, Bristol-Myers Squibb Company, P.O. Box 5400, Hopewell, New Jersey 08543-5400, and Combinatorial Drug Discovery, Bristol-Myers Squibb Company, P.O. Box 4000, Princeton, New Jersey 08543-4000

Received August 19, 1999

Analysis of a large amount of information, typically generated by high-throughput screening, is a very difficult task. To address this problem, we have developed binary formal inference-based recursive modeling using atom and physicochemical property class pair and torsion descriptors. Recursive partitioning is an exploratory technique for identifying structure in data. The implemented algorithm utilizes a statistical hypothesis testing, similar to Hawkins' formal inference-based recursive modeling program, to separate a data set into two homogeneous subsets at each splitting node. This process is repeated recursively until no further separation can occur. Our implementation of recursive partitioning differs from previously reported approaches by employing a method to extract multiple features at each splitting node. The method was examined for its ability to distinguish random and real data sets. The effect of including a single descriptor and multiple descriptors in the splitting descriptor set was also studied. The method was tested using 27 401 National Cancer Institute (NCI) compounds and their pGI50 ($-\log(GI_{50})$) against the NCI-H23 cell line. The analyses show that partitioning using multiple descriptors is advantageous in analyzing the structure−activity relationship information.

INTRODUCTION

Rapid development of combinatorial chemistry,[1,2] the high-throughput screening technique, and automation in recent years has revolutionized the pharmaceutical industry, and provided a powerful alternative to traditional medicinal chemistry approaches to a lead generation and optimization. One visible outcome of these changes is the possible number of compounds that can be synthesized and tested. Nowadays, chemists no longer spend time contemplating design and synthesis of a compound to test a hypothesis one at a time. Through the power of combinatorial chemistry, a hypothesis can be tested more quickly, and the amount of time associated with a lead generation and optimization can be minimized considerably. As the amount of chemical and biological information increases, forming structure−activity relationships (SARs) becomes increasingly difficult. The simple fact of the matter is that the visual inspection of combinatorial products is impossible, and the application of many traditional quantitative structure−activity relationship (QSAR) methods to analyze a combinatorial library is not feasible due to their computational complexity.

Most QSAR methods basically consist of two steps: generation of descriptors and application of a correlation method. Many attempts have been made to capture the essence of a chemical entity;[3−5] numerous descriptors such as physicochemical properties,[6] molecular connectivity indices,[7,8] structural keys,[9,10] hashed fingerprints,[11] atom pairs,[12] topological torsions,[13] atom triples,[14] physicochemical atom pairs and topological torsions,[15] geometric pairs,[16] atom layers,[17] autocorrelation vectors,[18] BCUTs,[19,20] three- and four-point pharmacophore keys,[21−23] feature trees,[24] steric and electrostatic fields,[25] EVA,[26] and affinity fingerprints[27] have been developed and studied. Once a set of descriptors is computed (or measured), a correlation method can be applied to identify or correlate descriptors which are important in increasing or decreasing biological response. Among numerous traditional correlation methods available, multiple linear regression,[28] partial least squares (PLS) analysis,[29,30] back-propagation neural networks,[31−35] and counterpropagation neural networks[36,37] are most popular and can be combined with optimization methods such as evolutionary algorithms,[38,39] genetic algorithms,[40−42] or simulated annealing algorithms[43−45] to perform variable selection (i.e., to remove irrelevant descriptors). In theory, any combination of descriptor−correlation method is possible. Comparative molecular field analysis (CoMFA),[25] molecular shape analysis (MSA),[46] and genetic function approximation (GFA)[47] are well known and widely available examples. Because of the library size (e.g., tens and hundreds of thousands to millions), any computationally expensive descriptor−correlation method combination is not feasible. In addition, the selected descriptors should be informative and meaningful to chemists, providing a means to suggest new compounds for synthesis.

A number of methods have been reported to analyze large data sets. For example, King et al.[48,49] have described the application of inductive logic programming to QSAR. The algorithm attempts to explain the active and inactive compounds using structural or physicochemical information

* To whom correspondence should be addressed. E-mail: chos@bms.com; telephone: (203) 677-6168; fax (203) 677-6417.
† Combinatorial Drug Discovery, Wallingford, CT.
‡ Nonclinical Biostatistics.
§ Combinatorial Drug Discovery, Princeton, NJ.

10.1021/ci9908190 CCC: $19.00
Published on Web 03/24/2000

by forming a series of rules or inductive hypotheses. Klopman has described the multiple computer automated structure evaluation (MULTICASE) algorithm, which automatically generates fragments and evaluates their probability of being part of "biophores" (i.e., fragments which are necessary for activity) or "modulators" (i.e., fragments which inflect the activity).[50] Hurst et al. have described hologram QSAR (HQSAR),[51] which combines hashed fingerprints and PLS. HQSAR generates various linear and branched fragments for each compound in a data set and hashes them into a fixed length bin. The bin occupancies for each compound are then used in PLS analysis to identify fragments which strongly correlate with change in biological response. Gao et al. have described binary QSAR,[52] which is based on statistical discriminant analysis. Given a specific set of descriptors for a compound and the prior probability of a compound being active, the method estimates the posterior probability of the compound being active. Application of recursive partitioning methods, formal inference-based recursive modeling (FIRM)[53] and statistical classification of actives of molecules (SCAM),[54] using various 2D and 3D descriptors has been reported by Hawkins et al.,[55] Young et al.,[56,57] Rusinko et al.,[54] and Chen et al.[58] The algorithm works by recursively partitioning a data set into two homogeneous subsets until subsets can no longer be divided any more. The descriptor found in each splitting node of the resulting recursive partitioning (RP) tree can be used to analyze compounds in terminal nodes. The level of descriptive information one obtains depends strongly on the type of descriptors one uses. The recursive partitioning performed using 2D descriptors such as MACCS-II keys,[59] atom pairs,[12] and topological torsions[13] can identify biologically important fragments, whereas 3D atom pairs can provide important pharmacophore information.

In this paper, we report binary formal inference-based recursive modeling (BFIRM) using eight atom and physicochemical property class pair and torsion (APPCPT) descriptors: atom class pairs (ACPs), atom class torsions (ACTs), binding property pairs (BPPs), binding property torsions (BPTs), charge class pairs (CCPs), charge class torsions (CCTs), hydrophobic class pairs (HCPs), and hydrophobic class torsions (HCTs). Our implementation of recursive partitioning and APPCPT descriptors differs from previously reported approaches in three aspects. First, in our implementation of the recursive partitioning algorithm, we have employed a method to extract multiple features at each splitting node. Second, atom class is used to better decode identified descriptors, and unlike physicochemical property pairs and torsions described by Kearsely et al.,[15] our implementation utilizes the hydrophobicity contribution obtained from the atom-based log *P* calculation method rather than the fragment-based log *P* calculation method. Third, the descriptor count information has been encoded to each descriptor. The application of BFIRM using atom and physicochemical property class pair and torsion descriptors to analyze 27 401 compounds in the National Cancer Institute (NCI) database is described.

METHODS

NCI Database. The NCI database (December 98 release) containing 30 672 compounds was downloaded from http://dtp.nci.nih.gov/docs/cancer/cancer−data.html. A total of 29 422 compounds with pGIso (−log(concentration) that reduced cell growth to 50% of the starting level) tested against the NCI-H23 cell line (nonsmall cell lung cancer cell line) in molar concentration were extracted from the database. If there was more than one entry for the same compound, the entry with the highest number of tests was selected. Of the 29 422 compounds, 27 401 compounds were successfully converted to 3D MOL2 file format using CONCORD.[60] The reason for generating the 3D MOL2 file is explained below.

APPCPT. An atom pair descriptor is defined as all possible unique atom pairs with their shortest bond path and their counts. A topological torsion descriptor is defined as all possible unique four consecutively connected atoms and their counts. The definition of physicochemical property descriptors developed by Kearsley et al.[15] is very similar to the definition of original atom pairs and topological torsions. Rather than using atoms to generate pairs and topological torsions, Kearsely et al.[15] defined binding property, charge, and hydrophobic classes to group atoms that have similar binding properties, atomic charges, and atomic log *P* values, and used them to create pairs and topological torsions. The idea behind this approach is to perceive physicochemically equivalent atoms. Our implementation of atom pairs, topological torsions, and physicochemical property pairs and torsions differs in four ways. First, rather than using atom property values to describe atoms, atom classes are used. The definition of classes is shown in Table 1. The definition was originally developed by Wang et al.[61] for the calculation of XLOGP. Second, six different binding property classes were used: cations (1), anions (2), hydrogen bond donors (3), hydrogen bond acceptors (4), hydrophobic atoms (5), and polar atoms (6). The binding property classes were constructed by simply reclassifying atom classes shown in Table 1 to one of six classes. For example, atom classes 47 and 60 belong to the first binding property class (cations). Third, the Gasteiger and Huckel charge calculation method[62] was used to compute atomic charges. Fourth, atomic hydrophobic contribution values used to compute XLOGP were used as atomic log *P* values. A double-precision floating point is used to store each descriptor. For example, atom classes 2 and 56 separated 2 bonds away from each other can be encoded as 2.05602. The four consecutive atom classes 1, 5, 28, and 56 can be encoded as 1005.028056. Atom classes within each descriptor are sorted so that the lowest atom class always appears first. A compound can then be described by the total number of descriptors, the unique number of descriptors, and a set of double-precision floating points with their counts sorted in ascending order; descriptors are sorted by their names (i.e., double-precision floating points) and not by their counts. The complete description of a compound consists of eight sets of double-precision floating points. The internally developed C program MOL2CPT (MOL2 to Class Pairs and Torsions) has been written to generate ACP, ACT, BPP, BPT, CCP, CCT, HCP, and HCT descriptors from the CONCORD-generated MOL2 file. Two important pieces of information are extracted from the MOL2 file generated using CONCORD: atom types and atomic charges computed using the Gasteiger and Huckel method.[62] CONCORD provides an efficient means to generate consistent atom types and atomic charges.

Table 1. Atom Class Definition

class	description[a]	class	description[a]	class	description[a]
1	C.3–$CH_3R(\pi=0)$	38	C.ar–R..C(X)..X	75	N.2–X=NR
2	C.3–$CH_3R(\pi\neq0)$	39	C.ar–X..C(A)..X	76	N.2–X=NX
3	C.3–CH_3X	40	C.ar–A..C(A)..A	77	N.2–other
4	C.3–$CH_2R_2(\pi=0)$	41	C.ar–other	78	N.ar
5	C.3–$CH_2R_2(\pi\neq0)$	42	C.1–R≡CH	79	N.pl3–R-NH-R
6	C.3–CH_2RX	43	C.1–R≡CR	80	N.pl3–R-NH-X
7	C.3–CH_2X_2	44	C.1–R≡CX	81	N.pl3–X-NH-X
8	C.3–$CHR_3(\pi=0)$	45	C.1–R=C=R	82	N.pl3–A-NH-A(inRing)
9	C.3–$CHR_3(\pi\neq0)$	46	C.1–other	83	N.pl3–NA_3
10	C.3–CHR_2X	47	C.cat	84	N.pl3–NA_3(inRing)
11	C.3–$CHRX_2$	48	O.3–R-OH($\pi=0$)	85	N.pl3–other
12	C.3–CHX_3	49	O.3–R-OH($\pi\neq0$)	86	N.am–-NH_2
13	C.3–$CR_4(\pi=0)$	50	O.3–R-O-R	87	N.am–-NHR
14	C.3–$CR_4(\pi\neq0)$	51	O.3–R-O-R	88	N.am–-NHX
15	C.3–CR_3X	52	O.3–R-O-X	89	N.am–-NR_2
16	C.3–$C_{23}X_2$	53	O.3–X-O-X	90	N.am–-NRX
17	C.3–CRX_3	54	O.3–π-O-π(inRing)	91	N.am–other
18	C.3–CX_4	55	O.3–other	92	N.1
19	C.3–other	56	O.2–O=R	93	S.3–A-SH
20	C.2–R=CH_2	57	O.2–O=X	94	S.3–R-S-R
21	C.2–R=CHR	58	O.2–other	95	S.3–R-S-X
22	C.2–R=CHX	59	O.co2	96	S.3–π-S-π(inRing)
23	C.2–X=CHR	60	N.4	97	S.3–other
24	C.2–X=CHR	61	N.3–R-$NH_2(\pi=0)$	98	S.2
25	C.2–R=CR_2	62	N.3–R-$NH_2(\pi\neq0)$	99	S.o
26	C.2–R=CRX	63	N.3–X-NH_2	100	S.o2
27	C.2–R=CX_2	64	N.3–R-NH-R	101	P.3
28	C.2–X=CR_2	65	N.3–R-NH-X	102	F
29	C.2–X=CXR	66	N.3_X-NH-X	103	Cl
30	C.2–X=CX_2	67	N.3–NR_3	104	Br
31	C.2–other	68	N.3–NR_2X	105	I
32	C.ar–R..C(H)..R	69	N.3–NRX_2	106	T_-CN
33	C.ar–R..C(H)..X	70	N.3–NX_3	1078	T–-NCS
34	C.ar–X..C(H)..X	71	N.3–other	108	T_-NO
35	C.ar–R..C(R)..R	72	N.2–R=NH	109	T_-NO_2
36	C.ar–R..C(R)..X	73	N.2–R=NR	110	rest
37	C.ar–R..C(R)..X	74	N.2–R=NX		

[a] C.3 = sp^3 carbon; C.2 = sp^2 carbon; C.ar = aromatic carbon; C.1 = sp carbon; C.cat = carbocation; 0.3 = sp^3 oxygen; O.2 = sp^2 oxygen; O.co2 = oxygen in carboxylate and phosphate groups; N.4 = positively charged sp^3 nitrogen; N.3 = sp^3 nitrogen; N.2 = sp^2 nitrogen; N.ar = aromatic nitrogen; N.pl3 = trigonal planar nitrogen; N.am = amide nitrogen; N.1 = sp nitrogen; S.3 = sp^3 sulfur; S.2 = sp^2 sulfur; S.o = sulfoxide sulfur; S.o2 = sulfone sulfur; P.3 = sp^3 phosphorous; T = terminal group; R = any group linked through carbon; X = any heteroatom (0, N, S. P, and halogens); A = R or X; (π=O) = a connected atom does not have a π electron; ($\pi\neq0$) = a connected atom has a π electron; (inRing) = part of a ring system; ( ) = branching; - indicates a single bond; = indicates a double bond; ≡ indicates a triple bond; .. indicates an aromatic bond; other = other than specified; rest = does not belong to atom classes 1−109.

BFIRM. BFIRM is based on the FIRM algorithm published by Hawkins et al.[53] Implementation of the FIRM algorithm has been simplified by allowing only binary splits. BFIRM requires a QSAR-like table, where rows represent compounds, and columns represent the biological activity of each compound and unique APPCPT descriptors, as an input. The value of each cell in the descriptor columns is either one or zero for the presence or absence of a descriptor, respectively. To utilize the descriptor count information and still maintain the binary split, we have decided to modify APPCPT descriptors to reflect count information. For example, when the maximum number of occurrences of 2.05602 (atom classes 2 and 56 separated 2 bonds away from each other) is less than or equal to 3, additional descriptors (or column headings) will be created by simply appending 2 or 3 to 2.05602 (i.e., 2.0560202 and 2.0560203). If the number of occurrences is greater than 3, three additional descriptors will be created to cover three different count ranges. For example, if a compound contains 30 2.05602 descriptors and the maximum number of occurrences for this ACP descriptor is 78, columns 2.0560226, 2.0560252, and 2.0560278 will be added to the QSAR table in addition to the existing 2.05602 column. Columns 2.05602 and 2.0560226 will have the cell value of one because the compound contains 30 2.05602 descriptors, but columns 2.0560252 and 2.0560278 will have the cell value of zero. This makes it possible to distinguish descriptors according to their occurrences and, at the same time, limit the number of additional descriptors, at most, to three per descriptor. Once the QSAR table is created, each descriptor is used to separate compounds according to the presence (group 1) or absence (group 0) of the descriptor. The F value, which is the ratio of variances between groups and within groups, is then calculated using the following equation:

$$F = \frac{N_0(\bar{Y}_0 - \bar{Y})^2 + N_1(\bar{Y}_1 - \bar{Y})^2}{\left(\frac{1}{N_0 + N_1 - 2}\right)\left(\sum_{i=1}^{N_0}(Y_{0i} - \bar{Y}_0)^2 + \sum_{i=1}^{N_1}(Y_{1i} - \bar{Y}_1)^2\right)}$$

where

Table 2. Randomization Tests

		number of nodes[c] at different p values[d]												
RS[a]	MD[b]	1×10^{-40}	1×10^{-30}	1×10^{-20}	1×10^{-10}	1×10^{-9}	1×10^{-8}	1×10^{-7}	1×10^{-6}	1×10^{-5}	1×10^{-4}	1×10^{-3}	1×10^{-2}	1×10^{-1}
0	1	1	1	1	3	5	5	7	7	7	7	11	11	19
0	20	1	1	1	3	5	5	7	7	7	7	11	11	19
1	1	1	1	1	1	1	1	3	3	3	**9**	11	17	29
1	20	1	1	1	1	1	1	3	3	3	**11**	11	17	29
2	1	1	1	1	1	1	1	1	1	1	7	11	13	19
2	20	1	1	1	1	1	1	1	1	1	7	11	13	19
3	1	1	1	1	1	1	1	3	3	3	7	11	15	27
3	20	1	1	1	1	1	1	3	3	3	7	11	15	27
4	1	1	1	1	1	1	1	5	5	5	9	11	15	25
4	20	1	1	1	1	1	1	5	5	5	9	11	15	25
5	1	1	1	1	1	1	1	5	7	9	11	13	17	29
5	20	1	1	1	1	1	1	5	7	9	11	13	17	29
6	1	1	1	1	1	1	1	3	7	7	15	23	35	45
6	20	1	1	1	1	1	1	3	7	7	15	23	35	45
7	1	1	1	1	1	1	3	3	3	5	11	13	23	37
7	20	1	1	1	1	1	3	3	3	5	11	13	23	37
8	1	1	1	1	1	1	1	9	9	11	11	15	25	37
8	20	1	1	1	1	1	1	9	9	11	11	15	25	37
9	1	1	1	1	1	1	1	5	5	5	13	15	23	27
9	20	1	1	1	1	1	1	5	5	5	13	15	23	27

[a] Random sets generated by randomizing activity values using different random seeds. [b] Maximum number of allowed descriptors. [c] The number of generated nodes, including the node containing the entire data set. [d] The p value used for terminating BFIRM; the Bonferroni adjusted p value has to be less than this number to be considered for a split.

$$\bar{Y}_0 = \frac{\sum_{i=1}^{N_0} Y_{0i}}{N_0}, \quad \bar{Y}_1 = \frac{\sum_{i=1}^{N_1} Y_{1i}}{N_1}, \quad \text{and} \quad \bar{Y} = \frac{\sum_{i=1}^{N} Y_i}{N}$$

N_1 and N_0 represent the number of compounds in groups 1 and 0, respectively, and N represents the number of compounds in both groups ($N_1 + N_0$). Y_1 and Y_0 represent the activities of compounds in groups 1 and 0, respectively. The numerator can be simplified by expressing it without the average activity of compounds in groups 1 and 0.

$$F = \frac{(N_0 N_1)/(N_0 + N_1)(\bar{Y}_0 - \bar{Y}_1)^2}{\left(\frac{1}{N_0 + N_1 - 2}\right)\left(\sum_{i=1}^{N_0}(Y_{0i} - \bar{Y}_0)^2 + \sum_{i=1}^{N_1}(\bar{Y}_{1i} - \bar{Y}_1)^2\right)}$$

Once the F value is computed for each descriptor, they are sorted in descending order of their F values. The descriptor with the highest F value (which is the most significant because a high F value leads to a lower p value when the number of degrees of freedom is constant) is identified, and its p value, the probability of two averages generated by this descriptor being equal, is computed; the smaller the p value, the more evidence against two averages being equal. If the p value is less than a terminal p value, the descriptor with the next highest F value is identified, and compounds are separated again, this time, using top two descriptors (the ones with the highest and second highest F values). The significance of the resulting split is computed. This process is repeated each time using an additional descriptor to split until the p value becomes greater than a specified terminal value or until the number of selected descriptors reaches the maximum number of descriptors allowed. This group of selected descriptors represents the first split point. The idea is to identify as many related descriptors (or as much information) as possible. This process of calculating F values, sorting according to F values, selecting a set of descriptors to split, and computing the p value of the split is applied recursively to each separated subgroup until the p value is greater than a specified terminal value. Because of the large number of descriptors generated for each compound, the false positive probability of a descriptor is quite high. To limit such spurious splits, the Bonferroni adjustment[55] was made to each p value. To adjust, each p value was multiplied by the number of descriptors. The adjusted p value must be less than the terminal p value. Otherwise the split is not accepted. The terminal p value (i.e., the maximum p value allowed) of 1×10^{-40} was used for analyzing the NCI data set. The maximum number of descriptors for each split was set to 20.

Implementation. MOL2CPT and BFIRM programs have been implemented as C programs on a R10000 SGI server. Computationally expensive steps within the BFIRM program have been threaded to take advantage of the multiprocessor SGI server.

RESULTS

A binary file containing 244 264 unique APPCPT descriptors describing 27 401 compounds in the NCI data set was generated and used for all of the BFIRM runs described in our study. The BFIRM settings described in the Methods are used except otherwise noted.

Randomization Tests. The BFIRM method was initially examined for its ability to distinguish real and random data sets. Table 2 shows the randomization test results. Ten

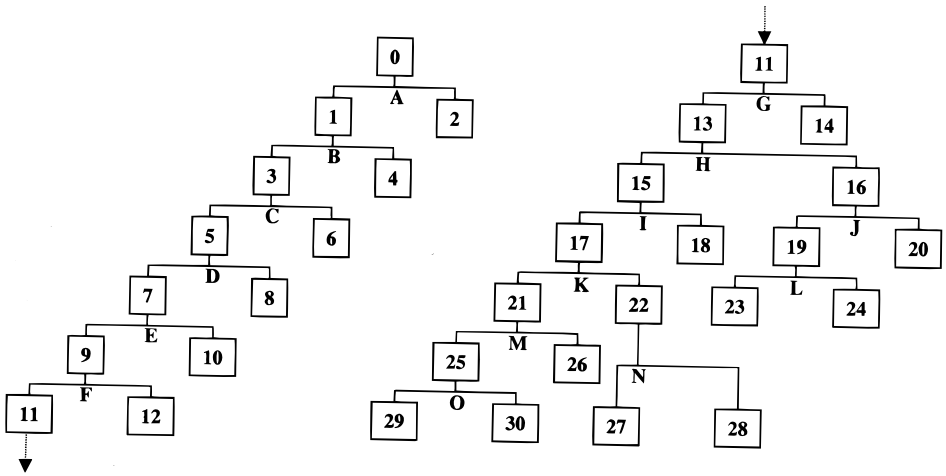

Figure 1. BFIRM tree generated with splitting descriptor sets containing a single descriptor.

different test data sets were created by randomizing pGI_{50}. For a comparison, each test was performed with the maximum number of allowed descriptors of 1, and it was repeated with 20. The terminal *p* value was varied from 1×10^{-40} to 1×10^{-1}. For each terminal *p* value, BFIRM was run, and the number of generated nodes was recorded. Except for one case, changing the maximum number of allowed descriptors from 1 to 20 did not alter the number of nodes generated. As the terminal *p* value was raised, an increase in the number of nodes was observed.

BFIRM Tree Generated with Splitting Descriptor Sets Containing a Single Descriptor. Figure 1 shows the BFIRM tree generated using the NCI data set, and Table 3 shows the corresponding RP tree node statistics and splitting descriptors. The number of descriptors that can be included in each split was set to one. The RP tree consists of 31 nodes (15 terminal nodes) and 15 splitting descriptors. Each descriptor set is denoted by the letters A−O. Compounds in a node located to the right of each letter in Figure 1 contain a descriptor denoted by the letter, and compounds in a node located to the left do not. The presence or absence of any particular descriptor set can also be represented by a plus (right) or minus (left). Compounds in nodes 18, 20, and 4 were found to contain the three highest averaged activities (9.2198, 7.1550, and 7.0021). The descriptor sets A−, B−, C−, D−, E−, F−, G−, H−, and I+ produced node 18 (I = ACP 10.03700902 ([C.3_CHR2X]-(9)-[C.ar_R..C(R)..X])). The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split were found to be 3.6900×10^2, 1.6988×10^{-79}%, and 4.1496×10^{-74}%, respectively. The descriptor sets A−, B−, C−, D−, E−, F−, G−, H+, and J+ produced node 20 (H = ACP 35.03600200 ([C.ar_R..C(R)..R]-(2)-[C.ar_R..C(X)..R]); J = ACT 21035.03203603 ([C.2_R=CHR]-[C.ar_R..C(R)..R]-[C.ar_R..C(H)..R]-[C.ar_R..C(X)..R])). The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split were found to be 2.5305×10^2, 1.3245 x 10^{-53}%, and 3.2353×10^{-48}%, respectively. The descriptor sets A− and B+ produced node 4 (B = ACT 10025.02502100 ([C.3_-CHR2X]-[C.2_R=CR2]-[C.2_R=CR2]-[C.2_R=CHR])). The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split were found to be 1.1045×10^3, 2.2135×10^{-235}%, and 5.4068×10^{-230}%, respectively. Figure 2 shows the structures of example compounds (378734, 615258, and 757) found in nodes 18, 20, and 4. Only ACP and ACT descriptors in descriptor sets I, H and J, and B are shown; a number next to a dashed line represents a bond distance.

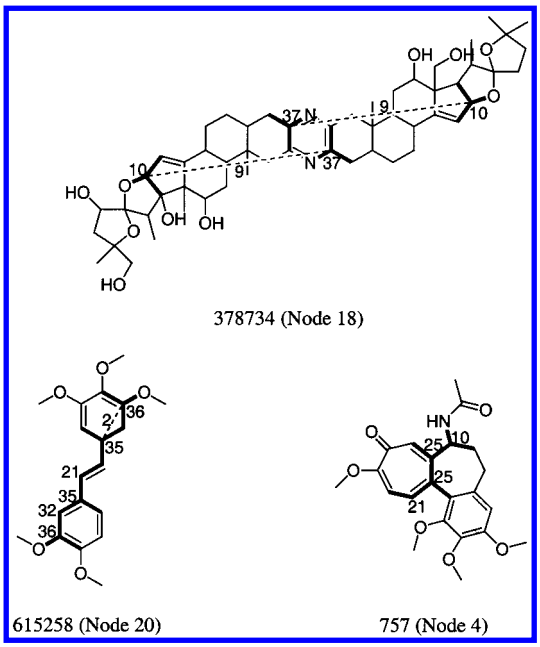

Figure 2. Example compounds found in the terminal nodes 18, 20, and 4 of the BFIRM tree generated with splitting descriptor sets containing a single descriptor (numbers represent atom classes, numbers next to a dashed line represent bond distances, and highlighted areas represent a fragment which was described by ACP and ACT descriptors).

BFIRM Tree Generated with Splitting Descriptor Sets Containing Multiple Descriptors. Figure 3 shows the tree generated by extracting multiple descriptors at each splitting point. Each descriptor set denoted by the letters A−U is described in Table 4; the program was stopped because the *p* value of the last subset was greater than the terminal *p* value. The maximum number of descriptors in any set was set to 20. The significance of each split was expressed in *F* and *p* values. The RP tree consists of 42 nodes (22 terminal nodes) and 21 splitting sets of descriptors. The average, standard deviation, and highest and lowest activities of the compounds in each node are shown in Table 5. Again, the presence or absence of any particular descriptor set is represented by a plus (right) or minus (left). Since each descriptor set contains multiple descriptor members, the presence of any descriptor set means that compounds must

Table 3. Descriptors Used To Split Each Node and Recursive Partitioning Tree Node Statistics

node[a]	descriptor[b]	*F* value	*p* value (%)[c]		N^d	av act.[e]	SD[f]
0					27401	4.4623	0.7917
1	A = CCT:2040404.18	1.5827×10^3	~0[g]	A−	26925	4.4377	0.7440
2				A+	476	5.8538	1.6772
3	B = ACT:10025.02502100	1.1045×10^3	5.4068×10^{-230}	B−	26836	4.4292	0.7254
4				**B+**	**89**	**7.0021**	**1.4773**
5	C = ACP:6.04000400	7.2287×10^2	9.3817×10^{-150}	C−	26075	4.4091	0.6956
6				C+	761	5.1169	1.2218
7	D = ACT:6008.00900900	7.0779×10^2	1.6790×10^{-146}	D−	25983	4.4024	0.6859
8				D+	92	6.3096	0.8226
9	E = ACP:33.09800700	7.1959×10^2	5.4271×10^{-149}	E−	25892	4.3957	0.6723
10				E+	91	6.3016	1.4561
11	F = ACP:28.02800300	7.2804×10^2	9.0221×10^{-151}	F−	24940	4.3740	0.6495
12				F+	952	4.9648	0.9526
13	G = ACP:8.02100700	6.0400×10^2	2.0451×10^{-124}	G−	24796	4.3664	0.6309
14				G+	144	5.6845	1.6767
15	H =ACP:35.03600200	5.8130×10^2	1.3820×10^{-119}	H−	19855	4.3187	0.5758
16				H+	4941	4.5578	0.7869
17	I = ACP:10.03700902	3.6900×10^2	4.1496×10^{-74}	I−	19850	4.3175	0.5703
18				**I+**	**5**	**9.2198**	**1.3269**
19	J = ACT:21035.03203603	2.5305×10^2	3.2353×10^{-48}	J−	4919	4.5462	0.7566
20				**J+**	**22**	**7.1550**	**2.1231**
21	K = CCP:30312.00	3.0660×10^2	9.5566×10^{-61}	K−	12747	4.2650	0.5338
22				K+	7103	4.4117	0.6195
23	L = ACP:35.08200202	2.7227×10^2	3.4500×10^{-52}	L−	4586	4.4995	0.7049
24				L+	333	5.1892	1.0821
25	M = ACP:32.03200400	2.6217×10^2	5.3816×10^{-5}[f]	M−	11596	4.2411	0.5138
26				M+	1151	4.5056	0.6578
27	N = ACT:3051.01001000	2.6080×10^2	2.9641×10^{-50}	N−	7086	4.4060	0.6032
28				N+	17	6.7922	1.7938
29	O = ACP:20.05600300	2.6184×10^2	7.2158×10^{-51}	O−	11424	4.2318	0.5014
30				O+	172	4.8634	0.8440

[a] Node number (see Figure 1). [b] The descriptor set used to split each node. See Table 1 and the text for decoding descriptors. [c] The Bonferroni adjusted *p* value. The actual *p* value can be obtained by dividing the Bonferroni adjusted *p* value by 244 264 (the number of descriptors). [d] The number of compounds found in each node. [e,f] The average and standard deviation of activities found in each node. [g] The actual *p* value ($<4.9407 \times 10^{-322}$) was outside the double-precision floating point range.

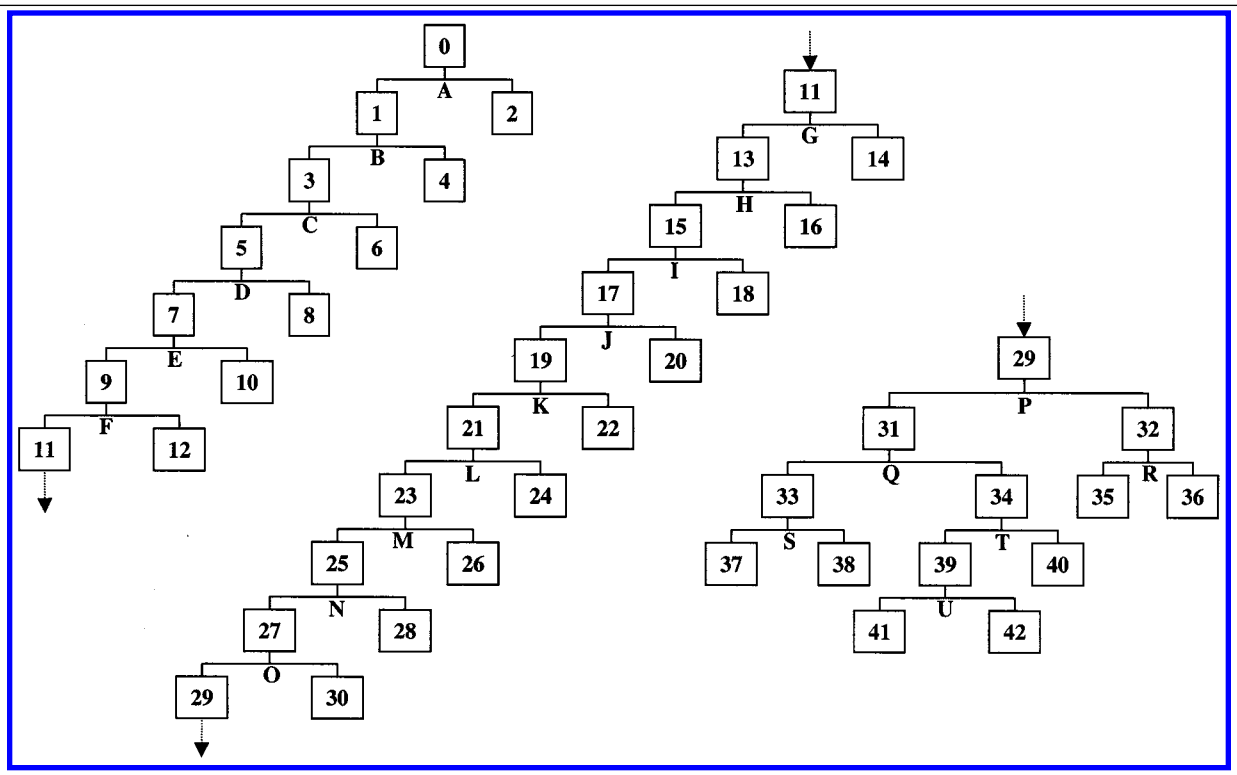

Figure 3. BFIRM tree generated with splitting descriptor sets containing multiple descriptors.

Table 4. Descriptors Used To Split Each Node

DS[a]	ND[b]	descriptors[c] in the set	*F* value	*p* value (%)[d]
A	10	CCT: 2040404.18 CCT: 3050505.18 ACT:4010.02502500 ([C.3_CH2R2(π=0)]-[C.3_CHR2X]-[C.2_R=CR2]-[C.2_R=CR2]) ACT:10004.01504800 ([C.3_CHR2X]-[C.3_CH2R2(π=0)]-[C.3_CR3X]-[O.3_R-OH(π=0)]) CCP: 50502.20 ACP: 10.01000400([C.3_CHR2X]-(4)-[C.3_CHR2X]) ACP:10.01000500 ([C.3_CHR2X]-(5)-[C.3_CHR2X]).ACP: 10.01500600 ([C.3_CHR2X]-(6)-[C.3_CR3X]) ACT: 15004.01005100([C.3_CR3X]-[C.3_CH2R2(π=0)]-[C.3_CHR2X]-[O.3_R-O-R]) ACP: 10.04800700 ([C.3_CHR2X]-(7)-[0.3_R-OH(π=0)])	6.5067×10^{2}	1.7695×10^{-134}
B	6	ACP: 10.02800902 ([C.3_CHR2X]-(9)-C.2_X=CR2]) ACP: 28.05100800 ([C.2_X=CR2]-(8)-[O.3_R-O-R]) CCT: 2040404.18 CCT:3050505.18 ACP: 10.02801000 ([C.3_CHR2X]-(10)-[C.2_X=CR2]) ACP: 28.02800300([C.2_X=CR2]-(3)- [C.2_X=CR2])	1.1787×10^{3}	1.4766×10^{-245}
C	20	ACT: 10025.02502100 ([C.3_CHR2X]-[C.2 _R=CR2]-[C. 2_R=CHR])ACT: 21025.02503500 ([C.2_R=CHR]-[C.2_R=CR2]-[C.2_R=CR2]-[C.ar_R..C(R)..R])ACT: 36036.03605100 ([C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[O.3_R-O-R]) ACT: 25021.02802600 ([C.2_R=CR2]-[C.2_R=CHR]-[C.2_X=CR2]-[C.2_R=CRX])ACT: 21021.02503500 ([C.2_R=CHR]-[C.2_R=CHR]-[C.2_R=CR2]-[C.ar_R..C(R)..R])ACT: 10025.02503500 ([C.3_CHR2X]-[C.2_R=CR2]-[C.2_R=CR2]-[C.ar_R..C(R)..R])ACT: 25035.03605100 ([C.2_R=CR2]-[C.ar_R..C(R)..R]-[C.ar_R..C(X)..R]-[0.3_R-O-R]) ACT: 5004.01002500 ([C.3_CH2R2($\pi\neq$0)]-[C.3_CH2R2(π=0)]-[C.3_CHR2X]-[C.2_R=CR2]) ACT: 25025.02102800([C.2_R=CR2]-[C.2_R=CR2]-[C.2_R=CHR]-[C.2_X=CR2]) ACP: 3.02500503 ([C.3_CH3X]-(5)-[C.2_R=CR2]) ACT: 25025.03503500([C.2_R=CR2]-[C.2_R=CR2]-[C.ar_R..C(R)..R]-C.aR..C(R)..R]) ACP: 21.05100602([C.2_R=CHR]-(6)-[O.3_R-O-R]) ACT:21025.03503500 ([C.2_R=CHR]-[C.2_R=CR2]-[C.arR..C(R)..R]-[C.ar_R..C(R)..R]) ACT:10025.02102800 ([C.3_CHR2X]-[C.2_R=CR2]-[C.2_R=CHR]-[C.2_X=CR2]) ACT:25021.02102600 ([C.2_R=CR2]-[C.2_R=CHR]-[C.2_R=CHR]-[C.2_R=CRX]) ACT:10004.00503500 ([C.3_CHR2X]-[C.3_CH2R2(π=0)]-[C.3_CH2R2($\pi\neq$0)]-[C.ar_R..C(R)..R]) ACT: 4010.02502500([C.3_CH2R2(π=0)]-[C.3_CHR2X]-[C.2_R=CR2]-[C.2_R=CR2]) ACT:32036.03603600 ([C.ar_R..C(H)..R]-[C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]) ACT: 3094.02602800([C.3_CH3X]-[S.3_R-S-R]-[C.2_R=CRX]-[C.2_X=CR2]) ACT: 21028.02609400([C.2_R=CHR]-[C.2_X=CR2]-[C.2_R=CRX]-[S.3_R-S-R])	8.6448×10^{2}	1.0558×10^{-179}
D	3	ACP: 6.04000300 ([C.3_CH2RX]-(3)-[C.ar_A..C(..A)..A]) ACP: 6.04000400([C.3_CH2RX]-(4)-[C.arA..C(..A)..A]) ACP:1.02100400 ([C.3_CH3R(π=0)]-(4)-[C.2_R=CHR])	5.7940×10^{2}	2.6141×10^{-119}
E	3	ACT: 36036.03605100 ([C.ar_R..C(X)..R]-C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[O.3_R-O-R]) ACP: 35.03600200 ([C.ar_R..C(R)..R]-(2)-[C.ar_R..C(X)..R]) BCP: 50602.19	5.1303×10^{2}	3.7392×10^{-105}
F	1	ACP: 28.02800300 ([C.2_X=CR2]-(3)-[C.2_X=CR2])	6.4582×10^{2}	1.9932×10^{-133}
G	1	ACP: 8.02100700 ([C.3_CHR3(π=O)]-(7)-[C.2_R=CHR])	6.8671×10^{2}	4.9241×10^{-142}
H	3	ACP: 33. 09800700 ([C.ar_R..C(H)..X]-(7)-[S.2])ACP: 33.08000500 ([C.ar_R..C(H)..X]-(5)-[N.pl3_R-NH-X]) ACP: 37.09800500-([C.ar_R..C(R)..X]-(5)-[S.2])	7.5051×10^{2}	1.6126×10^{-155}
I	9	ACP: 32.03200500 ([C.ar_R..C(H)..R]-(5)-[C.ar_R..C(H)..R]) HCP: 60607.00 ACT:6079.03604002 ([C.3_CH2RX]-[N.pl3_R-NH-R]-[C.ar_R..C(X)..R]-[C.ar_A..C(..A)..A]) ACP:6.04000302 ([C.3_CH2RX]-(3)-[C.ar_A..C(..A)..A]) HCP: 60606.00 HCP:60608.00 ACP: 6.04000400 ([C.3_CH2RX]-(4)-[C.ar_A..C(..A)..A]) ACP: 32.04000400-([C.ar_R..C(H)..R]-(4)-[C.ar_A..C(..A)..A]) HCP:50507.00	8.3550×10^{2}	1.9351×10^{-173}
J	12	HCP: 60607.00 ACP: 32.03500400([C.ar_R..C(H)..R]-(4)-[C.ar_R..C(R)..R]) ACP:35.03600200 ([C.arR_C(R)..R]-(2)-[C.ar_R..C(X)..R]) HCP: 60606.00 HCP: 60608.00ACP: 32.03200500 ([C.ar_R..C(H)..R]-(5)-[C.ar_R..C(H)..R] HCP: 50507.00 HCP: 50508.00ACP: 35.08200202 ([C.ar_R..C(R)..R]-(2)-[N.pl3_A-NH-A(inRing)]) HCP: 60609.00 HCP:60605.00 HCP: 50509.00	2.4811×10^{2}	2.9800×10^{-48}
K	17	HCP: 60607.00 HCP: 60606.00 HCP: 60608.00HCP: 50507.00 ACP: 32.03500400([C.ar_R..C(H)..R]-(4)-[C.ar_R..C(R)..R]) HCP:50508.00 ACP: 35.03600200 ([C.ar_R..C(R)..R]-(2)-[C.ar_R..C(X)..R]) ACP: 32.03200500([C.ar_R..C(H)..R]-(5)-[C.ar_R..C(H)..R]) HCP:60605.00 HCP: 60609.00 ACT: 36036.03605100([C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[C.ar_R..C(X)..R]-[O.3_R-O-R]) HCP: 50509.00CCP: 40410.00 CCP: 40411.00 BCP: 50609.00HCP: 60604.00 HCP: 50506.00	4.9659×10^{2}	1.3974×10^{-101}
L	10	HCP: 60607.00 HCP: 60606.00 HCP. 60608.00HCP: 50507.00 HCP: 50508.00 ACP:10.03701302 ([C.3_CHR2X]-(13)-[C.ar_R..C(R)..X]) ACP: 15.03701200([C.3_CR3X]-(12)-[C.ar_R..C(R)..X]) ACP:15.03701202 ([C.3_CR3X]-(12)-[C.ar_R..C(R)..X]) ACP: 10.03701300([C.3_CHR2X]-(13)-[C.ar_R..C(R)..X]) ACP:10.03701002 ([C.3_CHR2X]-(10)-[C.ar_R..C(R)..X])	4.3778×10^{2}	5.2622×10^{-89}

Table 4 (Continued)

DS[a]	ND[b]	descriptors[c] in the set	*F* value	*p* value (%)[d]
M	7	HCP: 60607.00 HCP: 60606.00 HCP: 60608.00HCP: 50507.00 ACP: 35.04000100([C.ar_R..C(R)..R]-(1)-[C.ar_A..C(..A)..A]) HCP:50508.00 HCP: 60605.00	4.0916×10^{2}	7.0324×10^{-83}
N	17	HCP: 60607.00 ACT: 21025.02503500([C.2_R=CHR]-[C.2_R=CR2]-[C.2_R=CR2]-[C.ar_R..C(R)..R]) ACT: 25021.02802600([C.2_R=CR2]-[C.2_R=CHR]-[C.2_X=CR2]-[C.2_R=CRX]) HCP: 60606.00 ACT:10025.02502100 ([C.3_CHR2X]-[C.2_R=CR2]-[C.2_R=CR2]-[C.2_R=CHR]) HCP: 50507.00HCP: 60608.00 ACT: 21021.02503500([C.2_R=CHR]-[C.2_R=CHR]-[C.2_R=CR2]-[C.ar_R..C(R)..R]) ACT: 21025.03503500([C.2_R=CHR]-[C.2_R=CR2]-[C.ar_R..C(R)..R]-[C.arR..C(R)..R]) HCP: 50508.00 ACT:25025.02102800 ([C.2_R=CR2]-[C.2_R=CR2]-[C.2_R=CHR]-[C.2_X=CR2]) HCP: 60605.00ACT: 25025.03503500 ([C.2_R=CR2]-[C.2_R=CR2]-[C.ar_R..C(R)..R]-[C.arR..C(R)..R]) HCP: 50509.00 HCP: 60609.00ACT: 21028.02605100 ([C.2_R=CHR]-[C.2_X=CR2]-[C.2_R=CRX]-[0.3_R-O-R]) HCP:50506.00	3.8955×10^{2}	1.1738×10^{-78}
O	14	HCP: 60607.00 HCP: 60606.00 HCP: 50507.00HCP: 60608.00 HCP: 50508.00 ACP: 10.01000400([C.3_CHR2X]-(4)-[C.3_CHR2X]) HCP: 60605.00HCP: 50509.00 HCP: 50506.00 HCP: 60609.00HCP: 60604.00 ACP: 9.01001200([C.3_CHR3($\pi \neq 0$)]-(12)-[C.3_CHR2X]) ACP:11.05601400 ([C.3_CHRX2]-(14)-[O.2_O=R])CCP: 40409.00	3.0166×10^{2}	8.8189×10^{-60}
P	20	HCP: 60607.00 HCP: 60606.00 HCP: 50507.00HCP: 60608.00 HCP: 50508.00 HCP: 60605.00HCP: 50506.00 HCP: 50509.00 HCP: 60609.00HCP: 60604.00 CCP: 40409.00 CCP: 30309.00CCP: 40410.00 BCP: 50609.00 HCP: 60611.00HCP: 50511.00 HCP: 60610.00 CCP: 40411.00HCP: 50512.00 CCP: 40412.00	3.8419×10^{2}	1.6502×10^{-77}
Q	1	ACP: 29.05600100 ([C.2_X=CXR]-(1)-[O.2_O=R])	3.1564×10^{2}	1.0675×10^{-62}
R	6	ACT: 4010.05101000 ([C.3_CH2R2(π=0)]-[C.3_CHR2X]-[O.3_R-O-R]-[C.3_CHR2X]) ACP:10.01000400 ([C.3_CHR2X]-(4)-[C.3_CHR2X])ACP: 10.04800400 ([C.3_CHR2X]-(4)-[O.3_R-OH(π=0)]) ACP: 10.04800500 ([C.3_CHR2X]-(5)-[O.3_R-OH(π=0)]) ACP: 48.05100300([O.3_R-OH(π=0)]-(3)-[O.3_R-O-R]) ACP:10.01000700 ([C.3_CHR2X]-(7)-[C.3_CHR2X])	2.8294×10^{2}	3.7595×10^{-54}
S	2	ACT: 35009.02503 500 ([C.arR..C(R)..R]-[C.3_CHR3($\pi \neq 0$)]-[C.2_R=CR2]-[C.ar_R..C(R)..R]) ACT: 21025.00903500([C.2_R=CHR]-[C.2_R=CR2]-[C.3_CHR3($\pi \neq 0$)]-[C.ar_R..C(R)..R])	2.3211×10^{2}	1.9627×10^{-44}
T	4	ACP: 20.05600300 ([C.2_R=CH2]-(3)-[0.2_0=R])ACP: 20.02900200 ([C.2_R=CH2]-(2)-[C.2_X=CXR]) ACT: 20025.02905600([C.2_R=CH2]-[C.2_R=CR2]-[C.2_X=CXR]-[O.2_0=R]) ACT: 20025.02905100([C.2_R=CH2]-[C.2_R=CR2]-[C.2_X=CXR]-[O.3_R-0-R])	2.5617×10^{2}	1.3537×10^{-49}
U	1	ACT: 10010.01501300 ([C.3_CHR2X]-[C.3_CHR2X]-[C.3_CR3X]-[C.3_CR4(π=0)])	2.4413×10^{2}	5.0629×10^{-47}

[a] Descriptor set used to split each node (see Figure 3). [b] The number of descriptors in the descriptor set. [c] See Table 1 and text for decoding descriptors. [d] The Bonferroni adjusted *p* value. The actual *p* value can be obtained by dividing the Bonferroni adjusted *p* value by 244 264 (the number of descriptors).

contain all of the descriptors in the set. If any one of the descriptors in the set is missing, compounds are considered as not having the full set and partitioned to the left side of the RP tree. Compounds in nodes 38, 24, and 6 were found to contain the three highest averaged activities (9.7523, 9.4190, and 7.8526). The descriptor sets A−, B−, C−, D−, E−, F−, G−, H−, I−, J−, K−, L−, M−, N−, O−, P−, Q−, and S+ produced node 38. The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split containing two descriptors were found to be 2.3211×10^{2}, 8.0351×10^{-50}%, and 1.9627×10^{-44}%, respectively. The descriptor sets A−, B−, C−, D−, E−, F−, G−, H−, I−, J−, K−, and L+ produced node 24. The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split containing 10 descriptors were found to be 4.3778×10^{2}, 2.1543×10^{-94}%, and 5.2622×10^{-89}%, respectively. The descriptor sets A−, B−, and C+ produced node 6. The *F* value, *p* value, and Bonferroni adjusted *p* value for the final split containing 20 descriptors were found to be 8.6448×10^{2}, 4.3223×10^{-185}%, and 1.0558×10^{-179}%, respectively. Figure 4 shows the structures of example compounds (650393, 363980, and 186301) found in nodes 38, 24, and 6. Again only ACP and ACT descriptors in descriptor sets S, L, and C are shown; a number with a double-headed arrow represents a bond distance. Figure 5 shows a common fragment described by three ACT and four ACP descriptors and structures of six example compounds found in the first terminal node, node 2. The NSC (NCI's internal ID) number and pGI_{50} of each compound is also shown in Figure 5.

Run Time. The CPU times for three different BFIRM runs, which differ only in the maximum number of descriptors allowed, using one, three, five, or seven processors were measured. Table 6 shows CPU seconds required to perform BFIRM runs and the speedup achieved by increasing the number of processors. For each BFIRM run, adding an additional processor increased the speedup linearly. Using a R10000 processor, a data set containing 27 401 compounds and 244 264 unique APPCPT descriptors (with the maximum number of descriptors allowed equal to 20) can be analyzed in 14 645 CPU seconds (~4 CPU hours). About a 7-fold decrease in CPU seconds (2135 CPU seconds) was observed when the number of processors was increased to seven.

Visualization of the RP Tree. Figure 6 shows the screen shot of the RP tree viewer. The viewer, written in Java, is designed specifically to visualize and analyze the output of BFIRM. A user can translate, scale, and rotate the RP tree to display any node. Tree node statistics and the structure of compounds can also be displayed by clicking any node of interest.

Table 5. Recursive Partitioning Tree Node Statistics

node[a]	descriptor set[b]	*N*[c]	av act.[d]	SD[e]	highest act.[f]	lowest act.[g]
0		27401	4.4623	0.7917	12.9700	−4.0000
1	A−	27354	4.4573	0.7817	12.9700	−4.0000
2	A+	47	7.3713	1.1661	8.9810	4.6020
3	B−	27269	4.4484	0.7627	12.9700	−4.0000
4	B+	85	7.3031	1.3844	10.6100	4.1370
5	C−	27227	4.4432	0.7510	12.9700	−4.0000
6	**C+**	**42**	**7.8526**	**0.7061**	**9.0950**	**4.5940**
7	D−	27177	4.4385	0.7422	12.9700	−4.0000
8	D+	50	6.9706	1.1475	8.9440	4.0000
9	E−	27034	4.4312	0.7329	12.9700	−4.0000
10	E+	143	5.8276	1.1059	8.1460	4.0000
11	F−	26030	4.4092	0.7122	12.9700	−4.0000
12	F+	1004	5.0012	0.9884	10.000	−4.0000
13	G−	25860	4.3999	0.6924	10.4900	−4.0000
14	G+	170	5.8174	1.6604	12.9700	4.0000
15	H−	25779	4.3934	0.6791	10.4900	−4.0000
16	H+	81	6.4743	1.3982	8.0550	4.0000
17	I−	25723	4.3878	0.6660	10.4900	−4.0000
18	I+	56	6.9724	1.4057	10.0000	4.0000
19	J−	25566	4.3827	0.6597	10.4900	−4.0000
20	J+	157	5.2184	1.0514	8.8100	3.6050
21	K−	25361	4.3745	0.6434	10.4900	−4.0000
22	K+	205	5.3956	1.4261	10.0000	4.0000
23	L−	25354	4.3731	0.6379	10.0000	−4.0000
24	**L+**	**7**	**9.4190**	**0.7388**	**10.4900**	**8.5120**
25	M−	24649	4.3595	0.6238	10.0000	−4.0000
26	M+	705	4.8485	0.8940	8.7480	4.0000
27	N−	24632	4.3575	0.6182	10.0000	−4.0000
28	N+	17	7.3214	1.3261	9.1460	4.0000
29	O−	24614	4.3556	0.6135	10.0000	−4.0000
30	O+	18	6.8721	1.4335	9.4050	4.6480
31	P−	20315	4.3206	0.5843	10.0000	−4.0000
32	P+	4299	4.5210	0.7134	9.5070	3.4260
33	Q−	9835	4.3953	0.6162	10.0000	−4.0000
34	Q+	10480	4.2506	0.5434	10.0000	−4.0000
35	R−	4279	4.5089	0.6874	9.5070	3.4260
36	R+	20	7.1144	1.2810	8.9830	4.7240
37	S−	9832	4.3936	0.6092	10.0000	−4.0000
38	**S+**	**3**	**9.7523**	**0.4290**	**10.0000**	**9.2570**
39	T−	10328	4.2404	0.5302	10.0000	−4.0000
40	T+	152	4.9426	0.8820	7.0690	−3.1600
41	U−	10322	4.2385	0.5233	10.0000	−4.0000
42	U+	6	7.5825	1.4229	8.5110	4.7410

[a] Node number (see Figure 3). [b] The descriptor set used to split each node (see Table 4). [c] The number of compounds found in each node [d] The average and standard deviation of activities found in each node. [f,g] The highest and lowest activities found in each node.

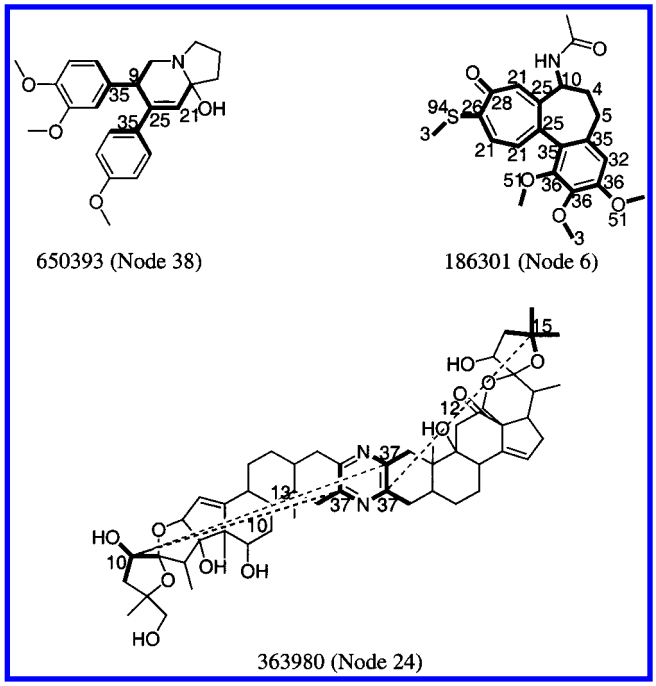

Figure 4. Example compounds found in the terminal nodes 38, 24, and 6 of the BFIRM tree generated with splitting descriptor sets containing multiple descriptors (numbers represent atom classes, numbers next to a dashed line represent bond distances, and highlighted areas represent a fragment which was described by ACP and ACT descriptors).

DISCUSSION

The current trend in pharmaceutical industry is to make and test a large number of diverse, druglike compounds. The goal is to increase the "hit rate" while minimizing the cost and time involved in chemical synthesis and biological testing. Undoubtedly, the success of any pharmaceutical company will strongly depend on its ability to capture and analyze SAR information and convert it to useful knowledge. In this paper, we describe an exploratory technique, BFIRM, to address the problem related to analyzing a large data set. The method utilizes a statistical hypothesis testing to identify a statistically significant set of descriptors and use them to partition the data set recursively until the data set can no longer be partitioned. Partitioned compounds and identified sets of descriptors can be useful in analyzing the data set and predicting the activity of novel compounds.

To test this methodology, we have selected 27 401 compounds in the NCI database tested against the NCI-H23 cell line. The selection of this particular cell line was based on the number of compounds tested. We wanted a large enough data set to simulate typical HTS data.

The method was first tested for its ability to distinguish real and random data sets. Since the BFIRM method tries to identify structure in data, randomly generated data sets having no structure should not be partitioned at all. At most the randomly generated data sets should give rise to a RP tree with a small number of nodes compared to the RP tree generated using a real data set. Table 2 shows that the method, indeed, is effective in distinguishing real from random data sets. None of the random data sets at the terminal p value of 1×10^{-40} were partitioned at all. As the terminal p value increases, the method began to partition the random data set, but the number of nodes found in each RP tree was very small compared to the number of nodes found in the RP tree generated using the real data set (at the terminal p value of 1×10^{-1} using the random set 0 with the maximum number of allowed descriptors of 20, the numbers of nodes found in random and real data sets were 19 and 579, respectively). Changing the maximum number of allowed descriptors did not affect the randomization tests, and the similar numbers of descriptors are found to be important at each node. This was expected since the method is designed to identify additional descriptors with an acceptable significance. In the absence of such additional descriptors, the results should be identical. In the case of random set 1 in Table 2, the discrepancy at the p value of 1×10^{-4} is due to identification of additional descriptors, which leads to a spurious split.

One of the goals of this paper is to demonstrate the need to better capture statistically significant structural features during the recursive partitioning process. Previously published studies were aimed at identifying a single descriptor

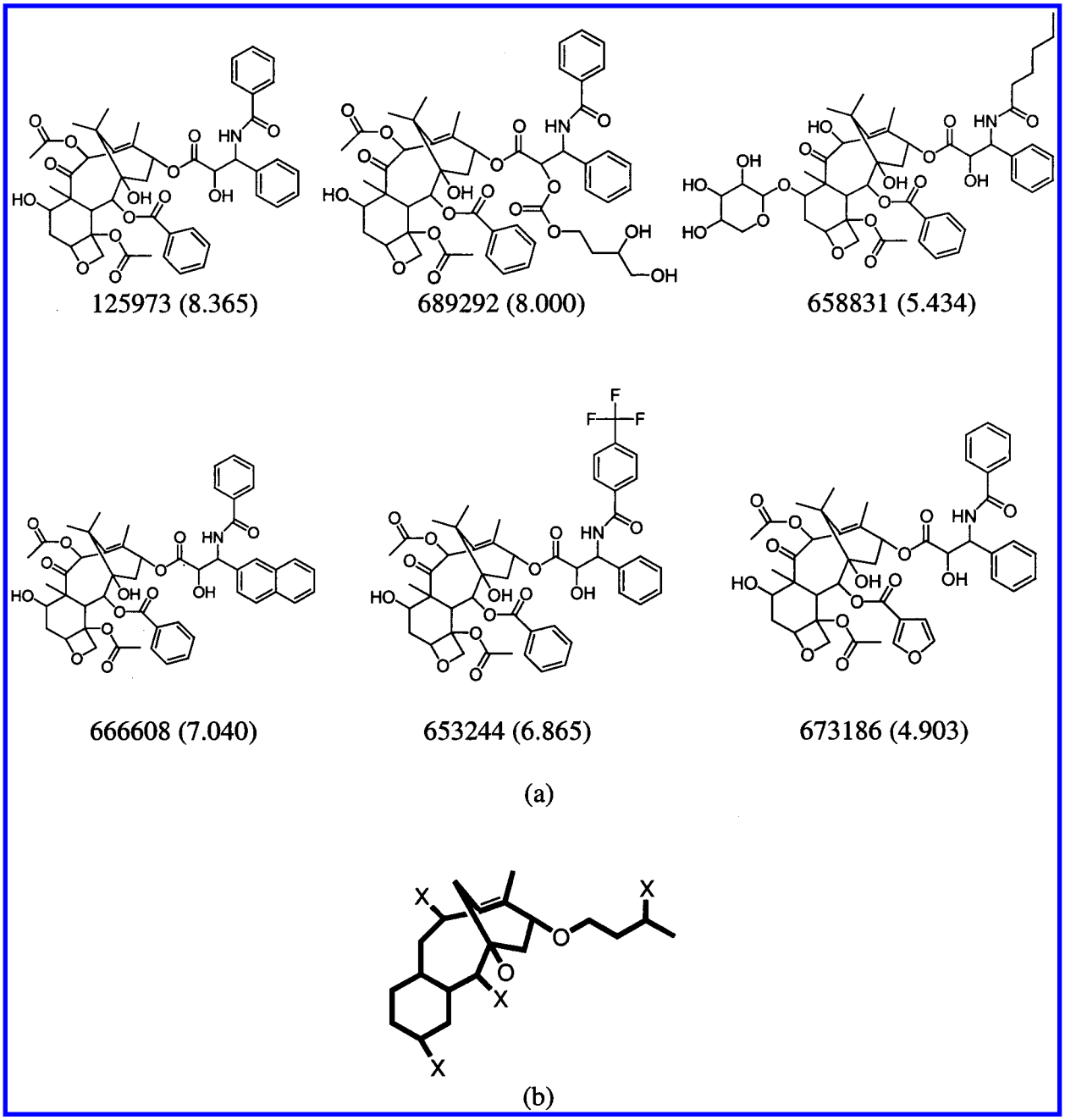

Figure 5. Example compounds found in the terminal node 2 of the BFIRM tree generated with splitting descriptor sets containing multiple descriptors: (a) paclitaxel and its derivatives; (b) a fragment described by the descriptor set A (only ACP and ACT descriptors were used; X = heteroatoms).

Table 6. CPU Time

	max. no. of des[a] = 1		max. no. of des[a] = 10		max. no. of des[a] = 20	
NP[b]	CPU (s)	speedup	CPU (s)	speedup	CPU (s)	speedup
1	9260	1	14421	1	14645	1
3	3074	3.01	4808	3.00	4873	3.01
5	1867	4.96	2906	4.96	2947	4.97
7	1391	6.66	2129	6.77	2135	6.86

[a] The maximum number of descriptors allowed. [b] The number of processors used.

to partition a data set. Often, descriptors identified are simple in nature and difficult to interpret because they encode a part of a statistically significant fragment. Because of this, any prediction of activities of compounds not in the training data set is very difficult. The BFIRM method addresses these problems by identifying multiple descriptors to partition the data set. This is achieved by computing the statistical significance of each unique descriptor in the data set, identifying a set of statistically significant descriptors, and partitioning the data set using the descriptors. The NCI data set was analyzed by BFIRM runs with descriptor sets containing a single descriptor (Figure 1 and Table 3) and multiple descriptors (Figure 3 and Tables 4 and 5) for a comparison. The main advantage of identifying multiple descriptors is that descriptors are no longer out of context. As a group, descriptors are more chemically and physicochemically meaningful and can better predict the activities of compounds found outside the training set. For example, a compound having a CCT 2040404.18 descriptor was found to have an average pGI_{50} value of 5.8538 (see node 2 in

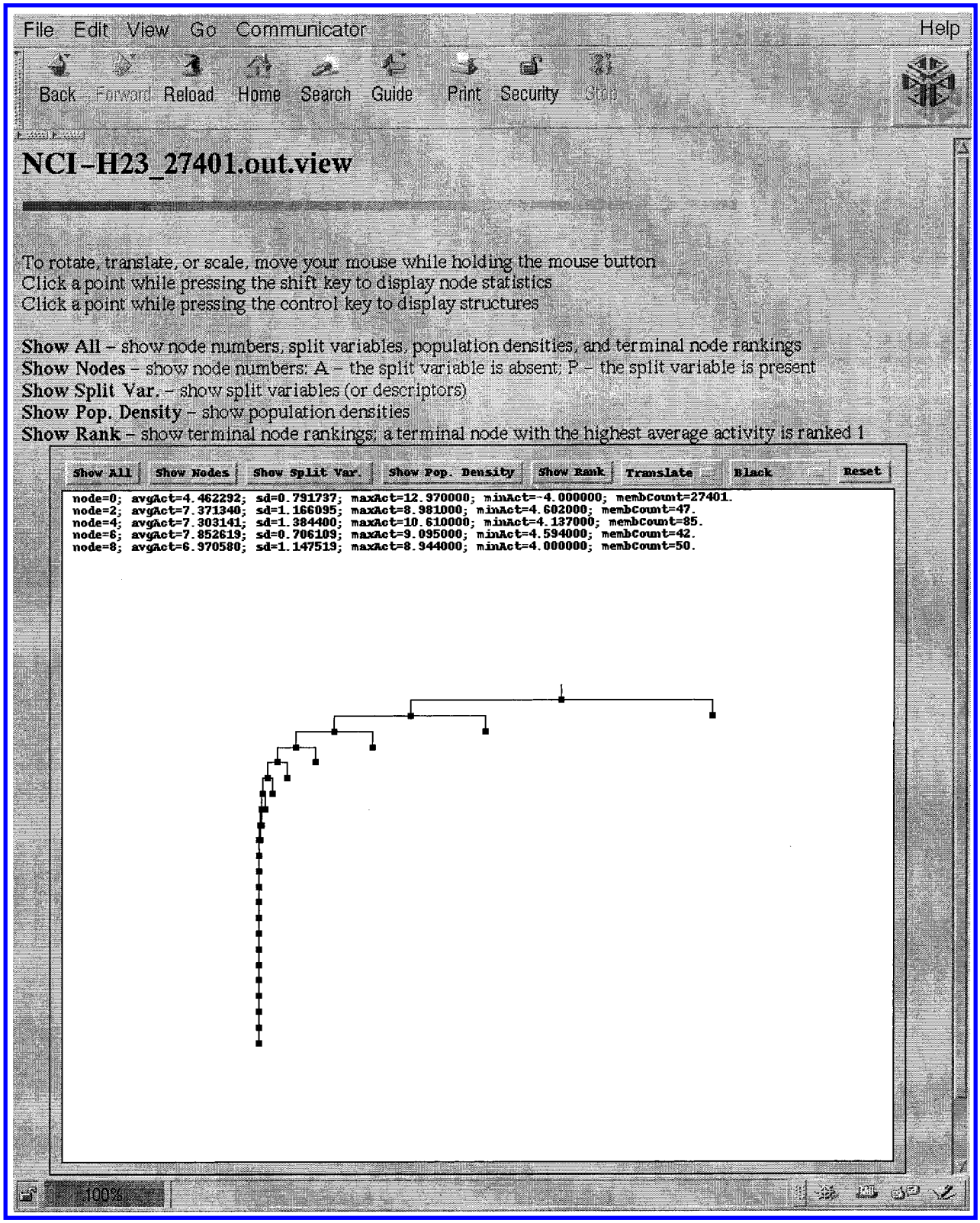

Figure 6. RP tree viewer.

Table 3), but the chance of a compound with a CCT 2040404.18 descriptor that is not in the NCI data set having a pGI_{50} value of around 5.8538 is very small. However, when multiple descriptors were used to split the same data set, 10 descriptors were identified to be significant (descriptor set A in Table 4). Figure 5 shows six example compounds found in this node and a common fragment among them; the fragment is described only by ACP and ACT descriptors. A major ring fragment common to both paclitaxel and its derivatives is identified. Clearly much more useful information can be extracted from a descriptor set containing 10 descriptors than a descriptor set containing 1 descriptor. In addition, the descriptor set A was able to partition the data set much tighter as shown by the average activities and their standard deviations of node 2 in Figures 1 and 3 (5.8538 ± 1.6772 for node 2 in Figure 1 and 7.3713 ± 1.1661 for node

2 in Figure 3). Unlike the compound with the only CCT 2040404.18 descriptor, the chance of a compound with these 10 descriptors having a pGI_{50} value of around 7.3713 is now very high.

In essence, BFIRM is a supervised maximum common substructure (SMCS) detection algorithm. The method detects MCSs by identifying a set of descriptors encoding them. It is supervised because the selection of compounds and descriptors, which encode MCSs, is based on the biological activities of compounds. One could then view the right terminal node of a RP tree as a cluster which contains compounds having a MCS, encoded by the splitting descriptor set.

One interesting point to note is that both RP trees (Figures 1 and 3) of the NCI data set were found to be one-sided (i.e., most of the split led to a terminal node), unlike previously published studies using peptides[56] and monoamine oxidase inhibitors.[55,58] This is because compounds found in the NCI data set are structurally much more diverse and have many different mechanisms of action (alkylating agents, alkyl transferase-dependent cross-linkers, DNA intercalators, polymerase inhibitors, ribonuclease reductase inhibitors, topoisomerase I and II inhibitors, tubulin-active antimitotic agents, etc.)[63] than data sets used in previously published studies. Because compounds are structurally diverse and compounds in the same structural class are more likely to exhibit similar pGI_{50} values, fragments which are unique to each structural class are likely to be identified as statistically most significant. Further partitioning within each structural class, however, is discouraged due to a very low terminal *p* value used in the BFIRM analyses. Much more branching was indeed observed when the terminal *p* value was raised (unpublished result). Figures 2, 4, and 5 show example compounds found in some of the terminal nodes.

Visual inspection, which is an essential part of the SAR study process, is almost an impossible task when dealing with HTS data. The RP tree constructed using BFIRM can alleviate the problem by "filtering" the data set. Figure 5 shows six example compounds found in node 2 in Figure 3 and their pGI_{50} values. Looking at these compounds, one could quickly form SARs important for increasing or decreasing pGI_{50} values. These compounds can then be further studied by the more traditional QSAR technique, which is now possible because of the substantially reduced number of compounds.

CONCLUSIONS

A novel recursive partitioning technique using APPCPT descriptors has been described. The method was examined for its ability to distinguish random and real data sets. The effect of including a single descriptor and multiple descriptors in the splitting descriptor set was also studied. Analyzing 27 401 NCI compounds using BFIRM clearly shows that partitioning using multiple descriptors is advantageous in analyzing the SAR information. The constructed RP tree can be potentially very useful in database searching as well as focused combinatorial library design.

REFERENCES AND NOTES

(1) Gallop, M. A.; Barret, R. W.; Dower, W. J.; Fodor, S. P. A.; Gordon, E. M. Applications of Combinatorial Technologies to Drug Discovery. 1. Background and Peptide Combinatorial Libraries. *J. Med. Chem.* **1994**, *37*, 1233−1251.
(2) Gordon, E. M.; Barret, R. W.; Dower, W. J.; Fodor, S. P. A.; Gallop, M. A. Applications of Combinatorial Technologies to Drug Discovery. 2. Combinatorial Organic Synthesis, Library Screening Strategies, and Future Directions. *J. Med. Chem.* **1994**, *37*, 1385−1401.
(3) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(4) Agrafiotis, D. K. Molecular Diversity. In *Encyclopedia of Computational Chemistry*; Schleyer, P. V. R., Ed.; John Wiley & Sons: New York, 1998.
(5) Brown, R. D.; Martin, Y. C. Use of Structure-Activity Data to Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *J. Chem. Inf Comput. Sci.* **1996**, *36*, 572−584.
(6) Kubinyi, H. QSAR: Hansch Analysis and Related Approaches. In *Methods and Principles in Medicinal Chemistry*; Mannhold, R., Krogsgaard-Larsen, P., Timmerman, H., Eds.; VCH: Weinheim, 1993; Vol. 1, pp 21−36.
(7) Kier, L. B.; Hall, L. H. *Molecular Connectivity in Chemistry and Drug Research*; Academic Press: New York, 1976.
(8) Kier, L. B.; Hall, L. H. *Molecular Connectivity in Structure-Activity Analysis*; John Wiley: London, 1986.
(9) Feldman, A.; Hodes, L. An Efficient Design for Chemical Structure Searching. I. The Screens. *J. Chem. Inf Comput. Sci.* **1975**, *15*, 147−152.
(10) Hodes, L. Selection of Descriptors According to Discrimination and Redundancy. Application to Chemical Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 88−93.
(11) James, C. A.; Weininger, D.; Delany, J. In *Daylight Theory Manual*, version 4.61; Daylight Chemical Information Systems Inc.: Irvine, CA, 1997.
(12) 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.
(13) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsions: A New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf Comput. Sci.* **1987**, *27*, 82−85.
(14) Nilikantan, R.; Bauman, N.; Venkataraghavan, R. New Method for Rapid Characterization of Molecular Shapes: Application in Drug Design. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 79−85.
(15) Kearsley, S. K.; Sallamack, S.; Fluder, E. M.; Andose, J. D.; Mosley, R. T.; Sheridan, R. P. Chemical Similarity Using Physiochemical Property Descriptors. *J. Chem. Inf Comput. Sci.* **1996**, *36*, 118−127.
(16) Sheridan, R. P.; Miller, M. D.; Underwood, D. J.; Kearsley, S. K. Chemical Similarity Using Geometric Atom Pair Descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 128−136.
(17) Martin, E. J.; Blaney, J. M.; Siani, M. A.; Spellmeyer, D. C.; Wong, A. K.; Moos, W. H. Measuring Diversity: Experimental Design of Combinatorial Libraries for Drug Discovery. *J. Med. Chem.* **1995**, *38*, 1431−1436.
(18) Moreau, G.; Broto, P. Autocorrelation of Molecular Structures: Application to SAR Studies. *Nouv. J. Chim.* **1980**, *4*, 757−764.
(19) Burden, F. R. Molecular Identification Number for Substructure Searches. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 225−227.
(20) Pearlman, R. S.; Smith, K. M. In *3D-QSAR and Drug Design: Recent Advances*; Kubinyi, H., Martin, Y., Folkers, G., Eds.; Kluwer Academic: Dordrecht, The Netherlands, 1997; pp 339−353.
(21) *Chem-X user guide*; Chemical Design Ltd: Oxon, U.K., 1997.
(22) Pickett, D.; Mason, J. S.; McLay, I. M. Diversity Profiling and Design Using 3D Pharmacophores: Pharmacophore-Derived Queries (PDQ). *J. Chem. Inf Comput. Sci.* **1996**, *36*, 1214−1223.
(23) Mason, J. S.; Cheney, D. L.; Morize, I.; Menard, P. R.; Bauerschmidt, S. Similarity and Diversity Methods for Drug Design: Use of Ligand-Based Properties and of Properties such as Potential Pharmacophores Determined from Both Ligands and Protein Targets. 215th ACS National Meeting, Dallas, TX, 1998; COMP 070.
(24) Rarey, M.; Dixon, J. S. Feature Trees: A New Molecular Similarity Measure based on Tree Matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471−490.
(25) Cramer, R. D., III; Patterson, D. E.; Bunce, J. D. Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959−5967.
(26) Ferguson, A. M.; Heritage, T. W.; Jonathon, P.; Pack, S. E.; Philips, L.; Rogan, J.; Snaith, P. J. EVA: A New Theoretically based Molecular Descriptor for Use in QSAR/QSPR Analysis. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 143−152.
(27) Kauvar, L. M.; Higgins, D. L.; Villar, H. O.; Sportsman, J. R.; Engqvist-Goldstein, A.; Bukar, R.; Bauer, K. E.; Dilley, H.; Rocke, D. M. Predicting Ligand Binding to Proteins by Affinity Fingerprinting. *Chem. Biol.* **1995**, *2*, 107−118.

(28) Draper, N. R.; Smith, H. *Applied Regression Analysis*; Wiley: New York, 1966.
(29) Wold, S.; Dunn, W. J., III. Multivariate QSARs: Conditions for their Applicability. J. *Chem. Inf: Comput. Sci.* **1983**, *23*, 6−13.
(30) Collantes, E. R.; Dunn, W. J., III. Amino Acid Side Chain Descriptors for Quantitative Structure-Activity Relationship Studies of Peptide Analogues. *J. Med. Chem.* **1995**, *38*, 2705−2713.
(31) Jansson, P. A. Neural Networks: An Overview. *Anal. Chem.* **1991**, *63*, 357A−362A.
(32) Tetko, I. V.; Luik, A. I.; Poda, G. I. Application of Neural Networks in Structure-Activity Relationships of a Small Number of Molecules. *J. Med. Chem.* **1993**, *36*, 811−814.
(33) Ajay. A Unified Framework for Using Neural Networks To Build QSARs. *J. Med. Chem.* **1993**, *36*, 3565−3571.
(34) So, S. S.; Richards, W. G. Application of Neural Networks: Quantitative Structure-Activity Relationships of the Derivatives of 2,4-Diamino-5-(substituted-benzyl)pyrimidines as DHFR Inhibitors. *J. Med.. Chem.* **1992**, *35*, 3201−3207.
(35) Gateiger, J.; Zuapn, J. Neural Networks in Chemistry. *Angew. Chem., Int. Ed. Engl.* **1993**, *32*, 503−527.
(36) Hecht-Nielsen, R. Counterpropagation Networks. *Appl. Opt.* **1987**, *26*, 4979−4984.
(37) Zupan, J.; Novic, M. Investigation of Infrared Spectra-Structure Correlation Using Kohonen and Counterpropagation Neural Network. *J. Chem. Inf Comput. Sci.* **1995**, *35*, 454−466.
(38) Fogel, D. B. Applying Evolutionary Programming to Selected Traveling Salesman Problems. *Cybern. Syst. (U.S.A.)* **1993**, *24*, 27−36.
(39) Fogel, D. B.; Fogel, L. J.; Porto, V. W. Evolutionary Methods for Training Neural Networks. *IEEE Conference on Neural Networks for Ocean Engineering (Cat. No. 91CH3064-3)*; 1991, pp 317−327.
(40) Goldberg, D. E. *Genetic Algorithm in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, 1989.
(41) Holland, J. H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66−72.
(42) Forrest, S. Genetic Algorithms: Principles of Natural Selection Applied to Computation. *Science* **1993**, *261*, 872−878.
(43) Bohachevsky, I. O.; Johnson, M. E.; Stein, M. L. Generalized Simulated Annealing for Function Optimization. *Technometrics* **1986**, *28*, 209−217.
(44) Kalivas, J. H.; Sutter, J. M.; Roberts, N. Global Optimization by Simulated Annealing with Wavelength Selection for Ultraviolet-Visible Spectrophotometry. *Anal. Chem.* **1989**, *61*, 2024−2030.
(45) Kalivas, J. H.; Generalized Simulated Annealing for Calibration Sample Selection from an Existing Set and Orthoginization of Undesigned Experiments. *J. Chemom.* **1991**, *5*, 37−48.
(46) Hopfinger, A. J. A QSAR Investigation of Dihydrofolate Reductase Inhibition by Baker Triazines Based upon Molecular Shape Analysis. *J. Am. Chem. Soc.* **1980**, *102*, 7196−7206.
(47) Rogers, D. R.; Hopfinger, A. J. Application of Genetic Function Approximation to Quantitative Structure-Activity Relationships and Quantitative Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854−866.
(48) King, R. D.; Muggleton, S.; Lewis, R. A.; Steinberg, M. J. E. Drug Design by Machine Learning: The Use of Inductive Logic Programming to Model the Structure-Activity Relationships of Trimethoprim Analogues Binding to Dihydrofolate Reductase. *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89*, 11322−11326.
(49) King, R. D.; Muggleton, S. H.; Srinivasan, A.; Sternberg, M. J. E. Structure-Activity Relationships Derived by Machine Learning: The Use of Atoms and Their Bond Connectivities to Predict Mutagenicity by Inductive Logic Programming. *Proc. Natl. Acad. Sci. U.S.A.* **1996**, *93*, 438−442.
(50) Klopman, G. MULTICASE 1. A Hierarchical Computer Automated Structure Evaluation Program. *Quant. Struct.-Act. Relat.* **1992**, *11*, 176−184.
(51) Hurst, T.; Heritage, T. HQSAR—A Highly Predictive QSAR Technique Based on Molecular Holograms. 213th ACS National Meeting, San Francisco, CA, 1997; CINF 019.
(52) Gao, H.; Williams, C.; Labute, P.; Bajorath, J. Binary Quantitative Structure-Activity Relationship (QSAR) Analysis of Estrogen Receptor Ligands. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 164−168.
(53) Hawkins, D. M.; Kass, G. V. Automatic Interaction Detection. In *Topics in Applied Multivariate Analysis;* Hawkins, D. H., Ed.; Cambridge University Press: Cambridge, U.K., 1982; pp 269−302.
(54) Rusinko, A.; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. SCAM: Statistical Classification of Activities of Molecules Using Recursive Partitioning. 213th ACS National Meeting, San Francisco, CA, 1997; CINF 068.
(55) Hawkins, D. M.; Young, S. S; Rusinko, A. Analysis of a Large Structure-Activity Data Set Using Recursive Partitioning. *Quant. Struct.-Act. Relat.* **1997**, *16*, 296−302.
(56) Young, S. S.; Hawkins, D. M. Analysis of a 29 Full Factorial Chemical Library. *J. Med. Chem.* **1995**, *38*, 2784−2788.
(57) Young, S. S.; Hawkins, D. M. Using Recursive Partitioning to Analyze a Large SAR Data Set. *SAR QSAR Env. Res.* **1998**, *8*, 183−193.
(58) Chen, X.; Rusinko, A.; Young, S. S. Recursive Partitioning Analysis of a Large Structure-Activity Data Set Using Three-Dimensional Descriptors. *J. Chem. Inf Comput. Sci.* **1998**, *38*, 1054−1062.
(59) *MACCS II*; Molecular Design Limited: San Leandro, CA, 1997.
(60) *CONCORD. A Program for the Rapid Generation of High Quality Approximate 3-Dimensional Molecular Structures*; The University of Texas: Austin, TX; Tripos Associates: St. Louis, MO.
(61) Wang, R.; Fu, Y.; Lai, L. A New Atom-Additive Method for Calculating Partition Coefficients. *J. Chem. Inf Comput. Sci.* **1997**, *37*, 615−621.
(62) *Force Field Manual*; Tripos Associates: St. Louis, MO, 1997.
(63) van Osdol, W. W.; Myers, T. G.; Paull, K. D.; Kohn, K. W.; Weinstein, J. N. Use of the Kohonen Self-organizing Map to Study the Mechanisms of Action of Chemotherapeutic Agents. *J. Natl. Cancer Inst.* **1994**, *86*, 1853−1859.

CI9908190