# Generation and Display of Activity-Weighted Chemical Hyperstructures

Nathan Brown, Peter Willett,* and David J. Wilton

Krebs Institute for Biomolecular Research, Department of Information Studies, University of Sheffield, Western Bank, Sheffield, S10 2TN, U.K.

Richard A. Lewis

Eli Lilly & Company, Erl Wood Manor, Windlesham, Surrey, GU20 6PH, U.K.

Received December 20, 2001

A chemical hyperstructure is a single graph representation of a set of molecules that minimizes the degree of structural redundancy in the data set. This paper describes the use of a genetic algorithm to generate an activity-weighted chemical hyperstructure (AWCH) by sequentially mapping each molecule in the data set to the hyperstructure and then assigning activity and inactivity frequency weights to the nodes and edges of the hyperstructure. Experiments with several data sets demonstrate the level of activity clustering in an AWCH.

## 1. INTRODUCTION

Computer-aided methods for the prediction of biological activity have been studied for many years. Approaches that have been discussed in the literature include the use of physicochemical parameters, connectivity indices, and molecular fields; in this paper, we focus on approaches that draw directly on the structure diagrams of the molecules in a data set. The first such approach was that described by Free and Wilson, who correlated the presence of substituents on a fixed central ring system with quantitative biological activity data.[1] This provides a simple way of developing QSARs for sets of congeneric molecules, and there have been many developments of this approach (e.g., refs 2 and 3), including its application to noncongeneric data sets.[4] However, most work on the analysis of noncongeneric data sets has involved the use of qualitative (i.e,.active or inactive) bioactivity data (e.g., refs 5−10). Such approaches are becoming of increasing interest with the widespread availability of high-throughput screening (HTS) data, and we here discuss the use of chemical hyperstructures for the analysis of such data; related work has been described by Simon et al., Downs et al., and Palyulin et al.[11−13] The next section introduces chemical hyperstructures, and we then discuss the use of a genetic algorithm (hereafter a GA) for their generation. The fourth section describes the incorporation of activity information to yield an activity-weighted chemical hyperstructure (AWCH), and we then report a series of experiments that illustrate the use of AWCHs for the analysis and visualization of data sets for which both structural and activity data are available. The paper concludes with a summary of our major findings.

## 2. CHEMICAL HYPERSTRUCTURES

**2.1. Hyperstructure Generation.** The chemical hyperstructure (or hypergraph) is a single structure representation of a library, which is generated by the sequential overlapping of each molecular graph in the library to the current hyperstructure. The overlapping is carried out so as to minimize the size (in terms of numbers of nodes and edges) of the resulting hyperstructure; an example of a possible hyperstructure is shown in Figure 1. The atom and bond data for the input molecules are retained as features of the relevant nodes and edges of the hyperstructure, hence making it possible to analyze the resulting hyperstructure with respect to its constituent molecules. The chemical hyperstructure representation, as proposed by Vladutz and Gould,[14] was originally devised principally to improve the efficiency of substructure searching. The rationale for the approach is that

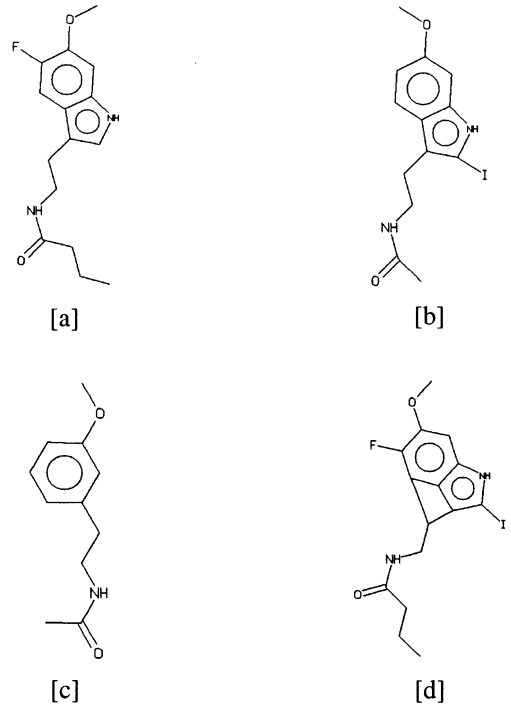


**Figure 1.** A chemical hyperstructure [d] generated from the molecules [a], [b], and [c].

* Corresponding author phone: +44-114-2222633; e-mail: p.willett@sheffield.ac.uk.

10.1021/ci0103875 CCC: $25.00 
Published on Web 01/27/2003

only a single run of the substructure search process need be applied to the hyperstructure to locate all the molecules that contain a given query fragment, as opposed to consecutively applying the same search process to each molecule in the source library. Although attractive in principle, experimental studies have shown that it is difficult to obtain substantial increases in search efficiency when compared with conventional substructure searching.[15,16]

A generated chemical hyperstructure can exhibit a number of characteristics not normally observed in molecular graphs. First, molecular graphs are bound by the laws of valence, so that the nodes of a molecular graph may only be connected to the maximum valence for the atom type of a particular node. However, the chemical hyperstructure is not bound by such laws, and it is therefore possible for a hyperstructure node to be connected to any, or all, of the other hyperstructure nodes. Similarly, two particular nodes of the hyperstructure may be connected by any of the possible bond types available, to preserve the structural integrity of the input molecules; in this sense the chemical hyperstructure can be defined as an instance of a multigraph. Finally, it is possible for a chemical hyperstructure to exhibit "ghost" substructures, which are substructures of the hyperstructure that are viable molecules or molecular features while not actually being a feature of any one molecule in the library from which the hyperstructure was generated. These features of the chemical hyperstructure introduce a number of complications, not least of which is that the chemical hyperstructure tends to be far more complex than conventional molecular graphs: the problem of discovering optimal mappings of input molecules to such a graph is hence computationally difficult.

**2.2. Graph Theory.** A graph is a pair $G = [V, E]$ of disjoint sets of nodes (or vertices, points) and edges (or arcs, lines), where each element of $E$ is a *2*-element subset of $V$. By representing two-dimensional (2D) chemical compounds as molecular graphs it is possible to discover structural commonalities by applying various forms of graph isomorphism algorithms; in particular, graph-based methods have been used to identify the largest substructure common to a pair of molecules as a way of calculating intermolecular structural similarities (see, e.g., refs 17−20).

The differences between the alternative ways of inducing a subgraph are subtle but can be significant in the degree of overlap discovered when applying graph-matching algorithms: either a subset of the nodes or a subset of the edges may be used to induce a subgraph of a graph. When a subgraph is induced by a subset of nodes, then those nodes are present in the subgraph along with all edges which are connected to the nodes in that subset. An edge-induced subgraph is the converse of this, where a subset of edges is induced together with all their incident nodes. The extensions of these methods of inducing subgraphs from a graph are the similar, but distinct, maximum common subgraph (MCS) problem and maximum overlap set (MOS) (or maximum common edge subgraph (MCES)) problem. The MCS of two graphs is the largest node-induced subgraph that is common to both graphs and does not feature in any larger subgraph with the same property. The MOS, analogously, is the largest edge-induced subgraph that is common to both graphs and that is not contained in a larger subgraph with this characteristic. Although both techniques discover an optimal mapping, the MOS problem is superior to the discovery of the MCS in that the resulting common subgraph from the MOS retains more of the topological characteristics of the two graphs being considered.

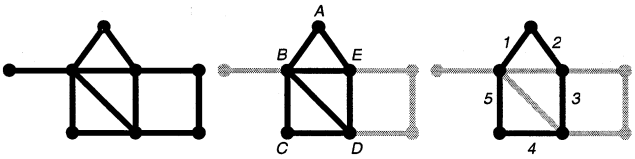


**Figure 2.** Source [$G$], node- [$G'$], and edge-induced [$G''$] subgraphs.

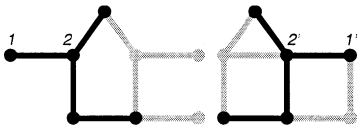


**Figure 3.** MCS of two graphs.

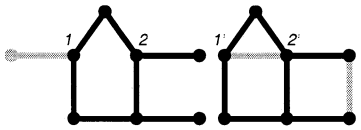


**Figure 4.** MOS of two graphs.

Figure 2 clearly illustrates the differences in inducing a subgraph by either a subset of nodes or a subset of edges of a graph $G$. The middle part of this figure highlights a node-induced subgraph, $G'$, with the node subset being $V = \{A, B, C, D, E\}$; it will be seen that all of the interconnecting edges of the nodes in the subset are also induced in the subgraph. Conversely, the right-hand graph, $G''$, emphasizes an edge-induced subgraph of $G$ in which $E = \{1, 2, 3, 4, 5\}$; here, it will be seen that all the incident nodes are also induced but not the interconnecting edges of those nodes.

**2.3. Generating Optimal Mappings.** The most intensive stage in the generation of a chemical hyperstructure with minimal structural redundancy is the discovery of an optimal mapping between a given molecule and the current hyperstructure. The definition of an optimal mapping in the context of hyperstructure generation is the equivalent of the graph-theoretic MOS-detection problem, i.e., the detection of the (not necessarily) disconnected set of maximum edge-induced subgraphs that is common to both the current molecule and hyperstructure, with each subgraph in common not being contained in a larger subgraph with the same property. The MOS provides the largest degree of overlap between two graphs being considered, including disconnected subgraphs that are common to the two graphs. Figures 3 and 4 illustrate the practical differences between the MCS and MOS of two graphs, respectively. Whereas the MCS of the two graphs in Figure 3 discovers *5* nodes and *4* edges in common between the two graphs, the MOS of the two graphs in Figure 4 discovers *7* nodes in common along with *7* edges.

Unfortunately, the discovery of such a mapping in hyperstructures is computationally intractable for all but the most trivial of problems, lying as it does in the domain of NP-Completeness. The more traditional approaches to graph matching, such as relaxation and backtracking, are not suitable for generating chemical hyperstructures, and previous workers have hence used an atom assignment approach;[15] while undoubtedly efficient in operation, this approach takes no account of the environments of the atoms and nodes of

```
initialise hyperstructure
foreach molecule in library [
    mapping: compare hyperstructure molecule
    append hyperstructure molecule mapping
]
```

**Figure 5.** Hyperstructure generation algorithm.

the molecule and hyperstructure, respectively, when the mapping is generated, this leading to hyperstructures with noticeably high edge densities. In this paper we employ a GA-based approach to discover a near-optimal overlap within a more reasonable time frame than a brute-force approach.

## 3. A GENETIC ALGORITHM FOR THE MOS PROBLEM

The GA is one of a family of heuristic optimization techniques that take their inspiration from processes observed in natural systems; the GA paradigm models an abstracted version of the processes in Darwinian evolution. The specific problem space is encoded in a chromosome, the form of which tends to be domain dependent. These chromosomes or candidate solutions are then evolved, by first sampling the more suitable individuals from the population and then applying computational analogues of biological recombination and mutation. Finally the candidate solutions are evaluated by a fitness function, which is able to decode the chromosome and determine its level of performance in the problem space. This process is then iterated until a suitable termination condition is met. Many of the problems encountered in computer-aided molecular design are inherently combinatorial in nature and GAs have hence been extensively applied in applications such as protein−ligand docking, the design of combinatorial libraries and QSAR variable selection; a review of such applications is provided by Clark.[22]

The only element of the basic hyperstructure generation algorithm shown in Figure 5 that is altered to incorporate the GA is the method used to discover the mapping, this being the computational bottleneck of the algorithm. Each mapping of an input molecule to the hyperstructure involves a single run of the GA to locate an optimal or near-optimal MOS between the two graphs. Our GA represents the candidate solutions as integer-encoded chromosomes, rather than the traditional bit-string representation. The length of each chromosome represents the number of atoms in the molecule, with the index of each gene within the chromosome representing a particular atom. The allele set, or the set of permitted values that the genes may take, is the set of indices of the nodes within the current hyperstructure, with the constraint that alleles may only map to valid genes determined by the hyperstructure and molecule atoms at these indices, respectively. Taking the molecule and the hyperstructure in Figure 6a as an example, an allele value of *3* at chromosome index *5* equates to a mapping between the hyperstructure node at index *3* and the molecular atom at index *5*, which is a valid mapping. Should a molecule contain an atom that is not represented in the hyperstructure, the relevant gene on the chromosome is set to *-1* to indicate that no mapping is possible. The fitness of a particular candidate solution is determined from the total number of edges that are preserved in the mapping, including any disconnected mappings if present.

The chromosome in Figure 6b provides the optimal mapping between the two graphs. One can see that, since the hyperstructure does not contain a node of the correct atom type for a mapping to be possible at index *10* (the iodine), the gene is flagged with the dummy allele value of *-1* to indicate that it should be ignored when attempting to locate a mapping. The fitness of this mapping can therefore be calculated as *18* since this is the number of edges preserved in the mapping.

Four genetic operators were implemented for this GA: mutation along with single-point, uniform, and node-based crossover. All these genetic operators function with the constraint that hyperstructure nodes must map to molecular atoms of the same atom type; in the case of the crossover operators a PMX-style crossover is employed to ensure this constraint is satisfied. The algorithm is described in detail by Brown et al.,[21] who also discuss the chromosome repair

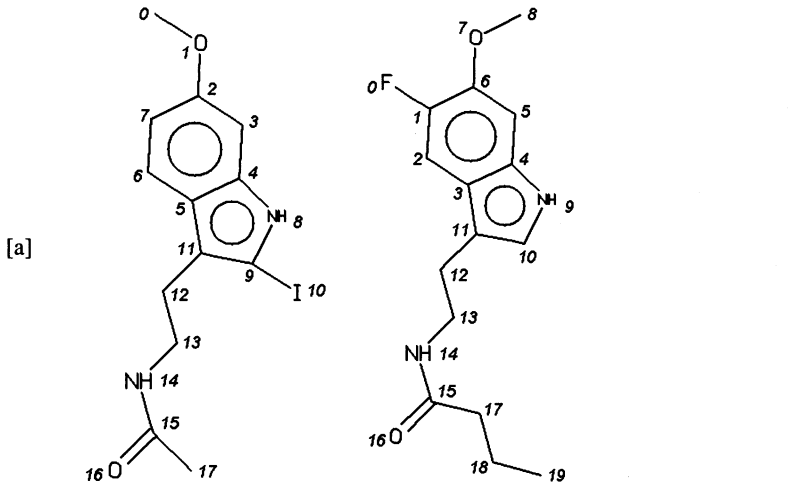


**Figure 6.** [a] One molecular and hyperstructure graph representations and [b] the chromosome of an optimal mapping where the chromosome length is determined from the molecule on the left and the allele set is given by the hyperstructure on the right.

mechanism necessary to remove any duplicates in the child chromosomes resulting from the crossover operators.

In this work, we have made two modifications to the algorithm of Brown et al. First is the introduction of optional mapping by SYBYL[23] atom type as a constraint, rather than mapping by the elemental type. This tends to increase the number of nodes while decreasing the number of resulting edges, in the hyperstructure; this improvement arguably renders a hyperstructure with less structural redundancy in a reduced time frame since the complexity of the mapping operation is reduced.

Second, the fine-tuning of the parameters of a GA is itself a combinatorial problem, in which there are a large number of permutations, from which it is difficult to determine (near) optimal values for each of the parameters. Therefore a program was written to evolve the parameter sets for the hyperstructure generation GA in an attempt to locate (near) optimal mappings while also endeavoring to reduce the CPU cycles of the process. This program evolves the parameter sets by means of another GA, in the style of Goldberg's simple GA.[24] The four genetic operator probabilities were encoded as a *24*-bit chromosome, with four components, each of six bits, representing the operator probability of each of the operators in the range [*0.63*]. To judge the fitness of each of the individuals in the population, it was necessary to test the probabilities expressed by the chromosomes by calling the hyperstructure generation algorithm with these parameters. The stochastic nature of the GA led us to run the program with two separate data sets, each containing five structures, and each being called three times. This equates to a total of *24* calls to the GA function to calculate the fitness of each individual in the population, ensuring that the fitness is both more accurately determined and that the evolved parameter set is robust under alternative conditions. The evolved parameter set for the genetic operators generates marginally smaller hyperstructures in runtimes that are, on average, *75%* of the average runtime using the original parameter set proposed by Brown et al.[21] The evolved parameters were used in all of the subsequent experiments and involved assigning operator probabilities of *35*, *8*, *5*, and *11* to mutation, node-based, uniform, and single-point crossover, respectively.

The hyperstructure generation program was implemented in ANSI C under SGI IRIX, with the output of the program being the hyperstructure in the form of a REBOL (Relative Expression-Based Object Language) script.[25] The hyperstructure is output in this way to permit offline analysis of its structure including its decomposition into the constituent molecules, as discussed further below.

## 4. THE ACTIVITY-WEIGHTED CHEMICAL HYPERSTRUCTURE

Given a data set containing molecules that are known to be either active or inactive in a particular assay of interest, it is possible to generate a chemical hyperstructure that is weighted according to the activities of each of its constituent molecules. In what follows, we will refer to such an *activity-weighted chemical hyperstructure* as an AWCH. Weighting the generated hyperstructure by the activity of its constituent molecules requires that all the hyperstructure nodes and edges are assigned two weights, one for activity and one for inactivity. Since the hyperstructure is initialized with a single molecule all the hyperstructure nodes and edges are initialized to one or zero according to the activity of the molecule. Subsequent mappings to these nodes and edges cause the respective activity or inactivity frequency counters to be incremented, dictated by the activity of the input molecule being mapped. Any new nodes and edges that are appended to the hyperstructure will also have their activity and inactivity weights initialized to one or zero accordingly to preserve the integrity of the activity of the input molecule.

Once the generation of the chemical hyperstructure has been completed, the nodes and edges of the hyperstructure contain weights for activity and inactivity in the range [*0..Act*] and [*0..Inact*], respectively; where *Act* and *Inact* are the numbers of active and inactive molecules in the input library, respectively. With these activity and inactivity weights it is possible to calculate summary values, which reflect the activity of each of the nodes and edges within the hyperstructure and hence summary statistics for the entire hyperstructure.

Substructural analysis studies have used a range of weighting schemes to measure the extent of the relationship between structure and activity.[5,7] Here, we have used two particularly simple ways of reporting the activity of nodes and edges within an AWCH: the *differential activity* and the *proportional activity*. The differential activity of a particular node or edge is calculated by simply subtracting the inactivity weight ($inact_i$) of the node or edge from the activity weight ($act_i$) of the same feature, i.e.

$$diff_i = act_i - inact_i$$

Whether the result is positive, negative, or zero indicates that the feature is weighted to be predominantly active, inactive, or neutral, respectively, given the particular sequence of molecular mappings applied during the hyperstructure generation process. The proportional activity is calculated by dividing the activity of a feature by the sum of its activity and inactivity weights, i.e.

$$prop_i = \frac{act_i}{act_i + inact_i}$$

The proportional activity hence lies in the range [*0.0...1.0*], *0.5* ostensibly indicating neutral behavior. It should be noted, however, that the differential and proportional activities might be skewed by the differences in size between the active and inactive molecules in the input library; this will also be the case should an uneven ratio of actives to inactives be used. We have found that both weights behave similarly, and we hence report just the results that were obtained with the proportional activity weights.

The resulting proportional activities can be summarized in several ways, as will be illustrated by an AWCH generated using *50* active compounds and *50* inactive compounds randomly selected from the NCI AIDS database.[26] This hyperstructure contained a total of *274* nodes and *704* edges (as against totals of *2643* nodes and *2856* edges in the set of source molecules). The graphs in Figures 7 and 8 illustrate the separation between the active and inactive nodes and edges of the hyperstructure in terms of their proportional activities. In these figures the nodes (or edges) of the

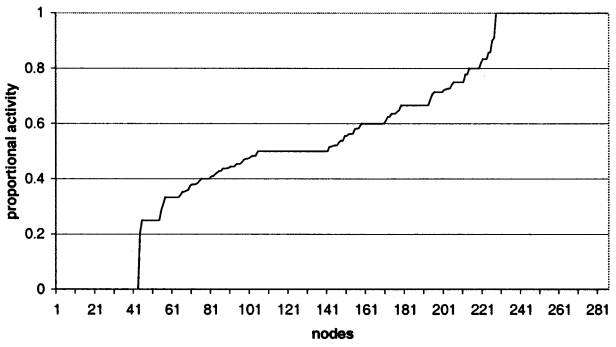


**Figure 7.** Proportional activities of hyperstructure nodes.

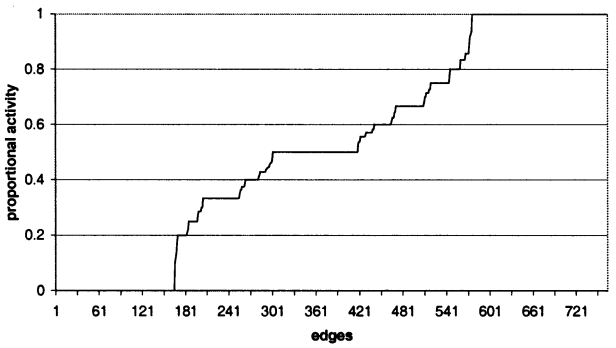


**Figure 8.** Proportional activities of hyperstructure edges.

hyperstructure have been sorted into increasing order of the calculated proportional activities. It will be seen that there tends to be a number of hyperstructure features that exhibit a high association with either activity (a value of *1*) or inactivity (a value of *0*).

It is possible to calculate both an activity and an inactivity figure for an entire AWCH, these summarizing the overall degrees of activity clustering in the hyperstructure. The degree of summary activity of a given hyperstructure is obtained by summing the products of the proportional activities and number of mappings of each of the features of interest when their proportional activities are greater than *0.5*, subtracting *0.5* from each proportional activity figure to ensure that the activities are given in the range [*0.0..0.5*]. This sum is calculated over all of the *n* active features, and the resulting mean value normalized to give a value in the range [*0.0..1.0*], i.e.

$$SummaryActivity = \frac{\sum_{i=1}^{n}\left(\frac{act_i}{act_i + inact_i} - 0.5\right)(act_i + inact_i)}{0.5n(Act + Inact)}$$

In just the same way, the inactivity figure summarizing a hyperstructure is calculated as

$$SummaryInactivity = \frac{\sum_{i=1}^{n}\left(0.5 - \frac{act_i}{act_i + inact_i}\right)(act_i + inact_i)}{0.5n(Act + Inact)}$$

with *n* here being the number of inactive features over which the summary is being calculated.

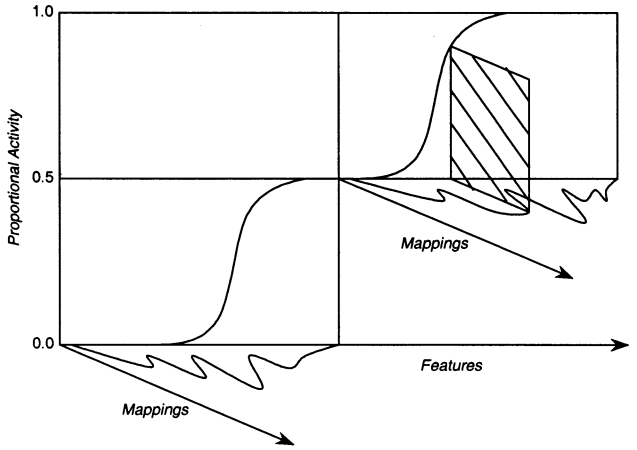


**Figure 9.** Graphical description of hyperstructure activity summaries.

**Table 1.** Summary Statistics for an AWCH Generated from *50* Actives and *50* Inactive Molecules from the NCI AIDS Database

| nodes | | edges | |
|---|---|---|---|
| *SummaryActivity* | *SummaryInactivity* | *SummaryActivity* | *SummaryInactivity* |
| 0.0294 | 0.0193 | 0.0242 | 0.0166 |

These calculations can be illustrated diagrammatically by the proportional plot shown in Figure 9. For each of the features in question (i.e., nodes or edges), the activity of the representational activity is given as the product of the activity and frequency of molecular mappings, as indicated. The sum of these individual feature activities is then divided by the product of active features, *0.5*, and the number of molecules in the input library; the same is true for the calculation of the inactivity figure from the lower-left quadrant. Nonzero values indicate a separation between those features that are perceived to be active or inactive. Table 1 provides the summary statistics for the AIDS AWCH. It will be seen that, for both the nodes and the edges, there would appear to be some clustering of the activity data in the hyperstructure. The significance of such clusterings is considered further in the section on validation below. It will be noted that the values in Table 1 are very low, this reflecting the fact that many of the features occur only once or twice in the hyperstructure, with a consequent low value for the $(act_i + inacti_i)/(Act + Inact)$ term in the two summary calculations. This suggests an alternative way of summarizing an AWCH, which gave similar results. Instead of multiplying the proportional activities by the frequency of molecular mappings, cutoff points were applied where only those hyperstructure features that had a number of molecular mappings greater than the cutoff point were taken into account when summarizing the activity of the hyperstructure. The application of this technique tends to result in a statistical separation from randomized activities when a cutoff point of *4* molecules is adopted; however the arbitrary nature of a specific cutoff point for all instances suggests that this technique may not be particularly robust. Therefore, for this paper, only the former method of summarizing an AWCH was applied.

## 5. VALIDATION OF THE ACTIVITY-WEIGHTED HYPERSTRUCTURE

**5.1. Randomization of Activity Data**. Randomization methods are widely used to validate SAR techniques, and

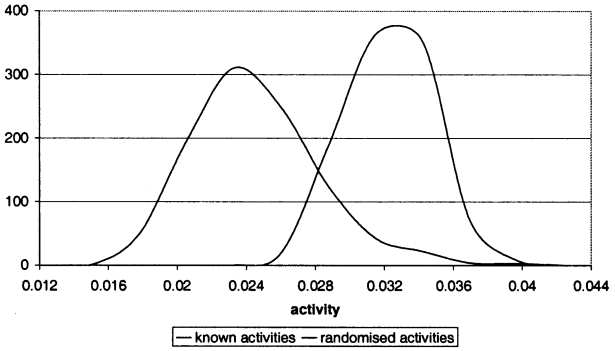


**Figure 10.** Node proportional activity distributions.

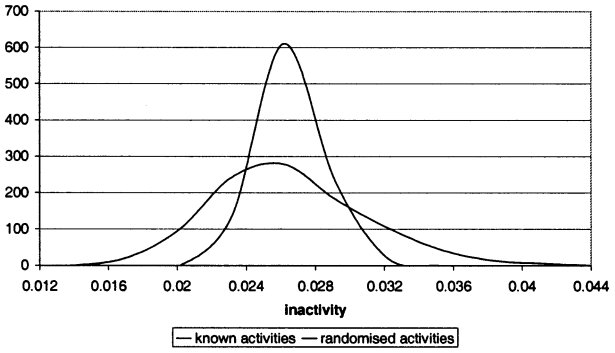


**Figure 11.** Node proportional inactivity distributions.

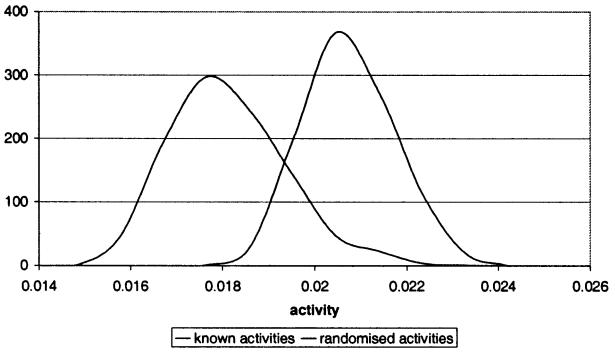


**Figure 12.** Edge proportional activity distributions.

we have used such a method to validate the clustering of the activity and inactivity classes within AWCHs. Specifically, the results from the actual activity classes were compared with AWCHs generated from the same set of input molecules but with randomized activity classes.

A data set of *50* active and *50* inactive molecules was randomly selected from the NCI AIDS database to generate the hyperstructures. This data set was used to generate *1000* hyperstructures with the original activity classes and *1000* hyperstructures with the randomized activity classes; the activity classes of this data set were randomized for each run of the randomized tests while retaining the balance of active and inactive structures. Each run of the hyperstructure generation program varied in its random ordering but with all other parameters being constant. Once the *2000* hyperstructures had been generated, they were processed to obtain the summary node and edge activity and inactivity values. The distributions of these values are shown in Figures 10−13. Visual inspection of these pairs of distributions suggests that they are different in character, and this is confirmed by a test of statistical significance using the *SummaryActivity* and *SummaryInactivity* values. These values were calculated,

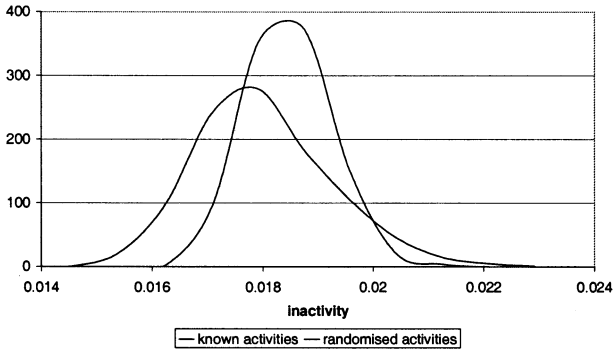


**Figure 13.** Edge proportional inactivity distributions.

**Table 2.** *t*-Test Scores Comparing the AWCHs Generated Using the Known Activity Classes and the Randomized Activity Classes[a]

| | nodes | | edges | |
|---|---|---|---|---|
| data set | active | inactive | active | inactive |
| AIDS | 54.57 | 13.80 | 53.03 | 9.78 |
| 5HT3 antagonists | 27.02 | 21.96 | 34.07 | 18.54 |
| ACE inhibitors | 31.59 | 4.83 | 35.41 | 2.47 |
| H1 antagonists | 35.64 | 6.21 | 33.70 | 0.86 |
| PDE IV inhibitors | 17.16 | 10.14 | 20.40 | 10.45 |

[a] Each data set consists of 50 active and 50 inactive molecules.

and a *t*-test was carried out to test the hypothesis that there is a significant difference between the means of the values for the actual and the random AWCHs. The *t*-test statistics are listed in the first row of Table 2, which demonstrates that statistically differences exist for all four of the types of plot; although the level of significance is (perhaps unsurprisingly) much greater in the case of the proportional activities.

To ensure that the results obtained thus far are not specific to the AIDS database, analogous experiments were carried out using sets of compounds from the IDAlert database.[27] Four activity classes were selected: 5HT3 antagonists, ACE inhibitors, H1 antagonists, and PDE IV inhibitors. In each case, *50* compounds were selected that had been assigned that pharmacological activity classification, along with another *50* compounds that had not been assigned that classification and that were thus assumed to be inactive. Each set of *100* compounds was used to generate *100* hyperstructures with the known activity classes and a further *100* hyperstructures with randomized activities. The *SummaryActivity* and *SummaryInactivity* values were calculated as described previously, and the resulting *t*-test statistics are listed in Table 2. Highly significant separations of the two distributions are again evident for all of the activity distributions; the *t*-values are much lower for the inactivity distributions and not significant at the 0.01 level in the case of the ACE inhibitor and H1 antagonist edge distributions. We hence conclude that there is a significant difference between AWCHs generated using real and randomized activity data.

**5.2. Effect of Ordering**. Since hyperstructure generation is a sequential process, it is likely that the order of processing will affect the nature of the resulting AWCH. A number of different ordering schemes were hence applied to gauge their effect, if any, on the degree of activity clustering in the final AWCH. The same 100-member AIDS data set was used here that had been employed previously in the initial tests. The

**Table 3.** *t*-Test Scores for AWCHs Generated Using Different Orderings of the Same Set of *50* Actives and *50* Inactives from the NCI AIDS Database

| ordering scheme | activity | |
|---|---|---|
| | node | edges |
| actives then inactives, ascending | 10.89 | 35.80 |
| actives then inactives, descending | 24.05 | 56.07 |
| actives then inactives, random | 12.16 | 39.76 |
| ascending | −8.14 | −8.06 |
| descending | 5.44 | 13.47 |
| inactives then actives, ascending | 3.02 | 6.78 |
| inactives then actives, descending | 14.02 | 9.04 |
| inactives then actives, random | 8.69 | 17.52 |

following orderings were considered (with each being repeated 100 times to alleviate the inherent variability of the GA): Ascending order (of molecular size); Descending order; Actives then inactives in random order; Inactives then actives in random order; Actives then inactives in ascending order; Inactives then actives in ascending order; Actives then inactives in descending order; and inactives then actives in descending order. In each case, the distribution of the *SummaryActivity* and *SummaryInactivity* values were compared with the corresponding distributions obtained from random ordering, where we used the first *100* AWCHs generated in the validation tests in Section 5.1 above. By calculating the *t*-scores between each of the ordering tests and the random ordering using the proportional summary, it is possible to compare the effectiveness of different ordering schemes, in terms of the degree of separation of the resulting activity classes. Table 3 indicates that sorting the molecular library by actives followed by inactives, or by inactives followed by actives, and then in descending order in the preprocessing phase tends to produce greater degrees of separation in the resulting AWCHs.

**5.3. Ratio of Actives to Inactives**. Training sets should ideally have equal numbers of actives and inactives,[28] as has been the case in all of the experiments reported thus far. However, it is often the case that this is not so when dealing with HTS analyses, where there may be only a few actives in an assay. We have hence carried out some limited experiments in which we have varied the ratio of active to inactive molecules using NCI AIDS data sets. Nine alternative ratios were used, from *10* actives and *90* inactives to *90* actives and *10* inactives, in *10*-molecule increments and decrements, respectively. Each such ratio of actives to inactives was used to generate five AWCHs and the proportional activities averaged over the five AWCHs in each case. The resulting averaged values are plotted in Figures 14 and 15, for the node and edge values, respectively. The same basic shape is obtained in all cases, albeit skewed in emphasising the predominant activity class. This suggests that there is some degree of clustering even when unequal numbers of active and inactive molecules are used to generate an AWCH.

## 6. VISUALIZATION OF ACTIVITY-WEIGHTED HYPERSTRUCTURES

As the AWCH is weighted according to activity and inactivity, it is possible to visualize any activity clustering that is present within it, thus providing a more user-oriented analytic approach than a simple listing of the calculated

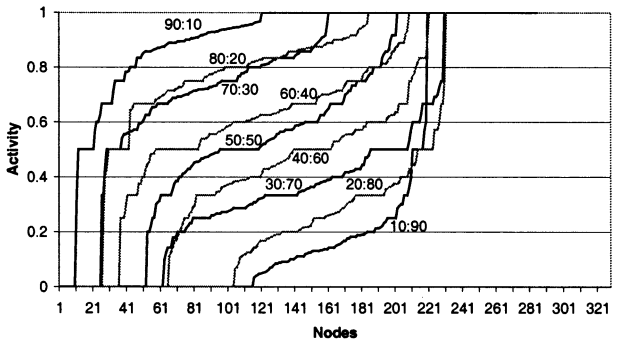


**Figure 14.** Effect of activity/inactivity ratios on hyperstructure nodes.

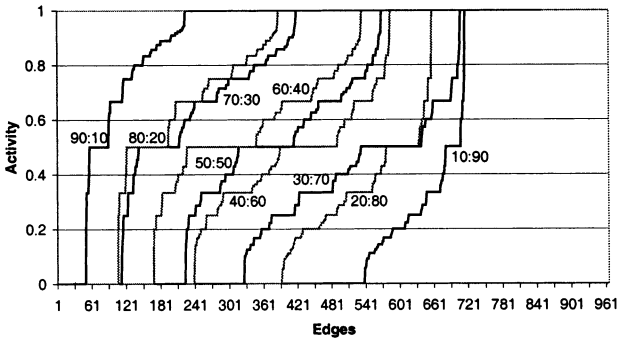


**Figure 15.** Effect of activity/inactivity ratios on hyperstructure edges.

proportional activities of the AWCH's nodes and edges. Two methods of achieving this visualization of activity clustering were considered: inducing activity subgraphs from the hyperstructure and visualizing the activity weights of individual molecules decomposed from the activity-weighted hyperstructure.

By calculating the proportional activities of the individual features within the hyperstructure it is possible to induce a subgraph containing only those features that have activities above or below some user-defined threshold value, indicating areas of high activity or inactivity, respectively. We have considered two ways of inducing such subgraphs of the AWCH: by utilizing both the node and the edge activity weights to identify the features in the subgraph and by utilizing either the node or the edge activity weights to identify the features in the subgraph. Once an activity subgraph has been induced it is possible to visualize it by applying color mapping to indicate the degree of activity of the features within that subgraph. This visualization technique provides a more accurate representation of the internal makeup of the subgraph, enabling the viewer to observe any areas that may be of interest. It does, however, have the problem that most hyperstructures are very complex (unless only a small number of molecules have been used to generate it), and similar comments can apply to subgraphs induced from it unless a further restriction is applied: here, features are induced from the AWCH only if some minimal number of input molecules have been mapped to them (since features with a greater number of mappings will provide a more reliable indication of their perceived activity). One might thus, for example, consider only those AWCH nodes or edges associated with at least 5% of the input file, but this often results in highly disconnected hyperstructures that are, again, difficult to comprehend.

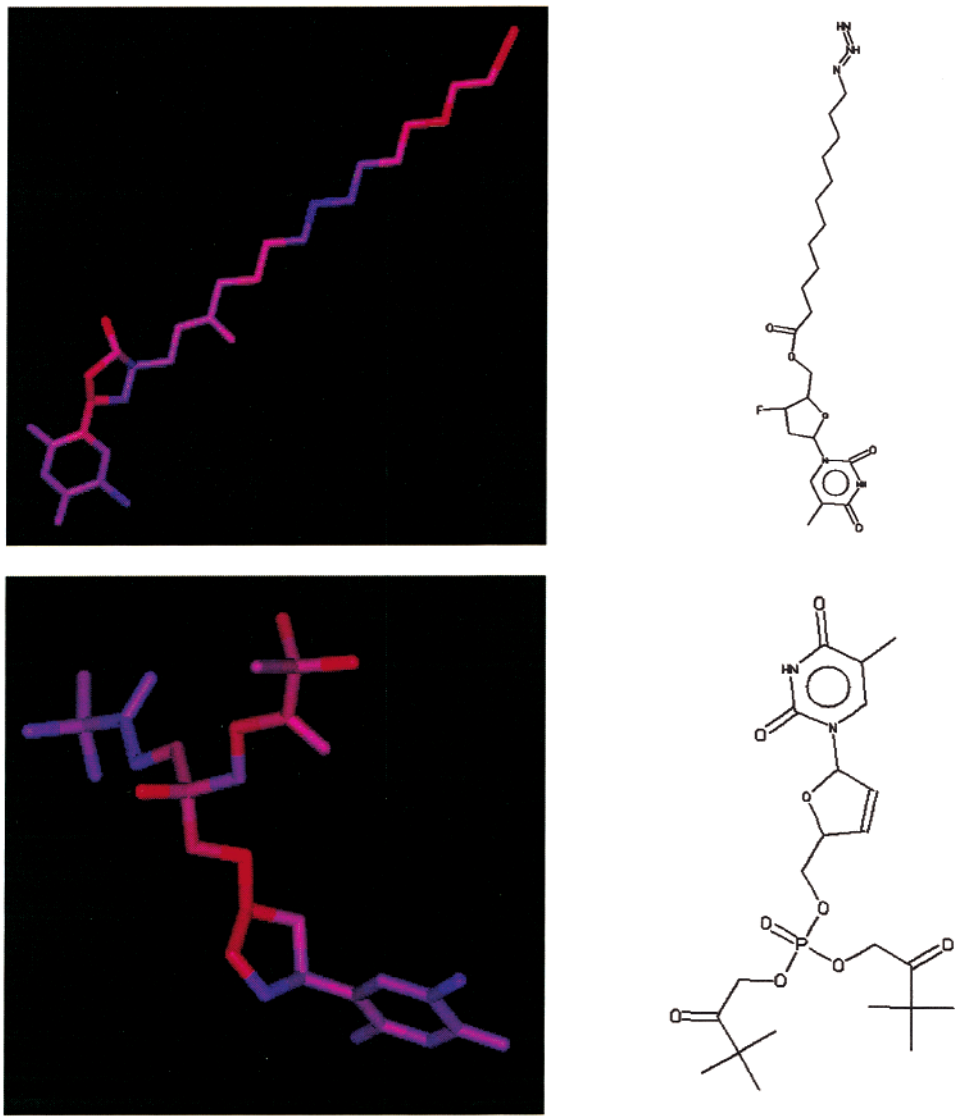

**Figure 16.** Examples of active molecules induced from an activity-weighted chemical hyperstructure and colored according to activity and inactivity.

Alternatively, we can consider the visualization of individual molecules. In mapping a particular molecule during the generation of a hyperstructure (see Figure 5), the relevant nodes and edges of the hyperstructure are either amended or appended to reflect the fact that those hyperstructure features also represent the features of the given molecule. The molecular source of the atom and bond information is hence preserved in the node and edge origin lists of the hyperstructure, respectively. Once the AWCH has been generated, each hyperstructure feature contains the two activity weights, and it is hence possible to induce the molecules from the hyperstructure while retaining these activity weights over the induced features. The preservation of these weights from the AWCH permits the interpretation of each of the molecules in the decomposed library individually of their perceived activity according to mappings applied in the hyperstructure generation stage; in fact, to obtain a more accurate representation of the perceived activity of a particular molecule in the AWCH, each feature of a given molecule has its relevant activity weight decreased by one so as to ensure that its own activity does not affect the perceived activity of the molecule. Since we can induce the original input molecules from the AWCH, together with their features' proportional activities, this can provide a basis for visualizing the molecules and hence facilitating the identification of particularly active or inactive substructures within the molecules, i.e., the visual identification of potential topological pharmacophores. The atoms and bonds of the induced molecules are colored according to the activity and inactivity weights associated with each feature of the graph. Two active molecules and two inactive molecules are presented in Figures 16 and 17, respectively; these are all induced from the same AWCH and colored according to their activity and inactivity weights, with red indicating activity and blue indicating inactivity.

## 7. CONCLUSIONS

In this paper, we have proposed the use of an activity-weighted chemical hyperstructure (AWCH) as a way of analyzing structural and bioactivity data by incrementing activity and inactivity features during hyperstructure generation. A number of methods have also been described for the analysis of the AWCH, permitting the hyperstructure to be summarized in terms of activity and inactivity. By inducing the constituent molecules of the AWCH it is possible to calculate a single summary figure of the molecule's perceived

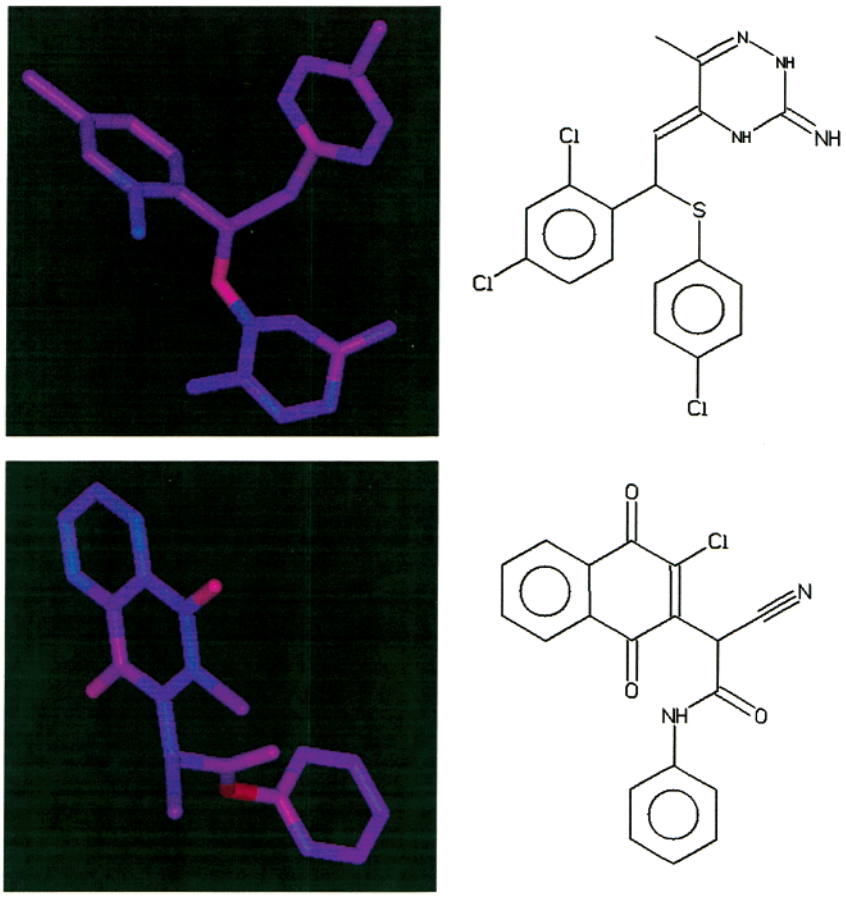


**Figure 17.** Examples of inactive molecules induced from an activity-weighted chemical hyperstructure and colored according to activity and inactivity.

activity as relating to the number of active and inactive mappings applied to its features during hyperstructure generation. Randomization experiments demonstrate the statistical significance of the activity clustering in the AWCHs. The visualizations of activity subgraphs of the AWCH and the induced molecules have additionally been proposed as a method of permitting a user to observe the activity mappings incremented during hyperstructure generation. These visualizations may assist the user in identifying interesting SARs that may not be apparent with the alternative summary techniques.

Thus far, we have considered the use of AWCHs for the qualitative visualization of structure−activity relationships. Two further applications in lead-discovery immediately suggest themselves: substructural analysis[5] and similarity searching.[29] Substructural analysis involves taking a training set of molecules for which qualitative activity data are available. Fragment weights are then calculated representing the probability of activity (or inactivity) for a molecule containing a specific fragment, and each test set molecule can then be scored by the sum-of-weights for its constituent fragments. This approach is simple but often surprisingly effective; however, its focus on discrete substructural features means that it takes no account of the overall topologies of the molecules in the training set and thus ignores potential interfragment relationships. An alternative approach would involve building a hyperstructure representation of the training set molecules and then mapping each of the test set molecules onto the hyperstructure, thus obtaining a sum-of-weights based on the parts of the hyperstructure to which it is mapped; these weights might be expected to be more discriminating than weights based on individual discrete fragments. Similarity searching involves taking a (typically bioactive) target structure and then ranking the molecules in a database in order of decreasing similarity with the target structure, using some quantitative measure of structural similarity. This measure is most commonly based on a comparison of the target-structure and database-structure bit-strings to identify the numbers of fragments in common and hence the similarity using a coefficient such as the Tanimoto coefficient. Shemetulskis et al. have considered the use of bit-strings that encode multiple target structures and have shown that they can result in improvements over the use of single target structures.[30] A hyperstructure generated from multiple target structures would encode in some detail the relationships between these molecules and could then be used for similarity searching, using either a bit-string[29] or a maximum common subgraph[20] measure of structural similarity. We hope to investigate these applications in the future.

## ACKNOWLEDGMENT

The authors thank the Engineering and Physical Sciences Research Council and Eli Lilly and Company for funding, Current Drugs Limited for provision of the ID Alert database, and the Royal Society, the Wolfson Foundation and Tripos Inc. for software and hardware support. The Krebs Institute for Biomolecular Research is a designated center of the Biotechnology and Biological Sciences Research Council.

## REFERENCES AND NOTES


(1) Free, S. M.; Wilson, J. W. A Mathematical Contribution to Structure Activity Studies. *J. Med. Chem.* **1964**, *7*, 395−399.
(2) Fujita, T.; Ban, T. Structure−Activity Study of Phenethylamines as Substrates of Biosynthetic Enzymes of Sympathetic Transmitters. *J. Med. Chem.* **1971**, *14*, 148−152.
(3) Mercier, C.; Mekenyan, O.; Dubois, J. E.; Bonchev, D. DARC/PELCO and OASIS Methods. I. Methodological Comparison. Modelling Purine $pK_a$ and Antitumour Activity. *Eur. J. Med. Chem.* **1991**, *26*, 575−592.
(4) Adamson, G. W.; Bush, J. A. A Method for Relating the Structure and Properties of Chemical Compounds. *Nature* **1974**, *248*, 406−407.
(5) Cramer, R. D.; Redl, G.; Berkoff, C. E. Substructural Analysis. A Novel Approach to the Problem of Drug Design. *J. Med. Chem.* **1973**, *17*, 533−535.
(6) Klopman, G. Artificial Intelligence Approach to Structure−Activity Studies. Computer Automated Structure Evaluation of Biological Activity of Organic Molecules. *J. Am. Chem. Soc.* **1984**, *106*, 7315−7321.
(7) Ormerod, A.; Willett, P.; Bawden, D. Comparison of Fragment Weighting Schemes for Substructural Analysis. *Quant. Struct.-Activ. Relat.* **1989**, *8*, 115−129.
(8) Sheridan, R. P.; Nachbar, R. B.; Bush, B. L. Extending the Trend Vector − the Trend Matrix and Sample-Based Partial Least-Squares. *J. Comput.-Aid. Mol. Design* **1994**, *8*, 323−340.
(9) Roberts, G.; Myatt, G. J.; Johnson, W. P.; Cross, K. P.; Blower, P. E. LeadScope: Software for Exploring Large Sets of Screening Data. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1302−1314.
(10) Poroikov, V.; Akimov, D.; Shabelnikova, E.; Filimonov, D. Top 200 Medicines: can new Actions be Discovered through Computer-Aided Prediction? *SAR QSAR Environ. Res.* **2001**, *12*, 327−344.
(11) Simon, Z.; Chirica, A.; Holban, S.; Ciubotaru, D.; Mihala, G. I. *Minimum Steric Difference. The MTD Method for QSAR Studies*; Research Studies Press: Letchworth, UK, 1994.
(12) Downs, G. M.; Gill, G. S.; Willett, P.; Walsh, P. T. Automated Descriptor Selection and Hyperstructure Generation to Assist SAR studies. *SAR QSAR Environment. Res.* **1995**, *3*, 253−264.
(13) Palyulin, V. A.; Radchenko, E. V.; Zefirov, N. A. Molecular Field Topology Analysis Method in QSAR Studies of Organic Compounds. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 659−667.
(14) Vladutz, G.; Gould, S. R. Joint Compound/Reaction Storage and Retrieval and Possibilities of a Hyperstructure-Based Solution. In *Chemical Structures. The International Language of Chemistry*; Warr, W. A., Ed.; Springer-Verlag: Berlin, 1988; pp 371−384.
(15) Brown, R. D.; Downs, G. M.; Willett, P. A Hyperstructure Model for Chemical Structure Handling: Generation and Atom-by-Atom Searching of Hyperstructures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 522−531.
(16) Brown, R. D.; Downs, G. M.; Jones, G.; Willett, P. A Hyperstructure Model for Chemical Structure Handling: Techniques for Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 47−53.
(17) Hagadone, T. R. Molecular Substructure Similarity Searching: Efficient Retrieval in Two-Dimensional Structure Databases. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 515−521.
(18) McGregor, J. J. Backtrack Search Algorithms and the Maximal Common Subgraph. *Software-Pract. Exper.* **1982**, *12*, 23−34.
(19) Bayada, D.; Simpson, R. W.; Johnson, A. P. An Algorithm for the Multiple-Common Subgraph Problem. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 680−685.
(20) Raymond, J. W.; Gardiner, E. J.; Willett, P. Heuristics for Rapid Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 305−316.
(21) Brown, R. D.; Jones, G.; Willett, P. Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 63−70.
(22) *Evolutionary Algorithms in Computer-Aided Molecular Design*; Clark, D. E., Ed.; Wiley-VCH: Weinheim, 2000.
(23) SYBYL is produced by Tripos Inc. at http://www.tripos.com/.
(24) Goldberg, D. E. *Genetic Algorithms in Search, Optimization and machine Learning*; Addison-Wesley: Reading, MA, 1989.
(25) REBOL Technologies Inc. is at http://www.rebol.com/.
(26) The NCI AIDS database is available at http://dtp.nci.nih.gov/.
(27) The ID Alert database is available from Current Drugs Limited at http://www.current-drugs.com/.
(28) Rosenkranz, H. S.; Cunningham, A. R. SAR Modeling of Unbalanced Data Sets. *SAR QSAR Environ. Res.* **2001**, *12*, 267−274.
(29) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(30) Shemetulskis, N. E.; Weininger, D.; Blankley, C. J.; Yang, J. J.; Humblet, C. STIGMATA: An Algorithm to Determine Structural Commonalities in Diverse Data Sets. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 862−871.


CI0103875