A Ring-Imbedding Index and Its Use in Substructure Searching

Alan H. Lipkus

Chemical Abstracts Service, P.O. Box 3012, Columbus, Ohio 43210-0012

Received June 30, 1996[⊗]

An easily calculated index that characterizes the way in which a substructure is imbedded in the rings of a structure is described. The index can be assigned to each of the structures in a substructure-search answer set, and the structures can be classified according to their index value. This classification can be used to sample the answer set in such a way that the diversity of ring imbeddings of the substructure query is well represented. It may also be useful in finding novel structures in an answer set. An application of this index to a large substructure-search answer set is presented.

INTRODUCTION

A substructure search of a large database can often retrieve structures in which the substructure query is imbedded in unexpected or unusual ways. This is especially likely when many acyclic bonds in the query are allowed to match either chain or ring bonds in the file structures. In that case, the query may be imbedded in one or more rings or ring systems in a wide variety of ways, particularly when the database is very large and structurally diverse. These rings are of considerable interest because of the constraining effect they can have on the conformation of the imbedded query. This suggests that the extent and nature of ring imbedding could be useful as a basis for organizing and analyzing answer sets from certain substructure searches.

The present work describes a method for easily calculating a single number that characterizes the way in which a given substructure is imbedded in the rings of a structure. This ring-imbedding index depends on a substructure as well as a structure; for this reason it is unlike other topological indexes that have been developed to characterize rings and ring systems.[1−5] This index can be assigned to each of the structures in a substructure-search answer set. The index will group together those structures in which the query is imbedded in rings in topologically identical, or related, ways. By grouping the structures in this manner, the index provides a basis for analyzing even large answer sets.

A RING-IMBEDDING INDEX

A two-part approach has been taken to the design of a ring-imbedding index. The first part is to define a type of graph that represents in an abstract form the basic topology of the imbedding. This will be called a *ring-imbedding graph*. The second part is to characterize certain properties of the ring-imbedding graph using simple graph invariants and to combine these invariants to yield the desired index. This index will eventually be described in a form in which it can be calculated without explicitly constructing the graph.

The ring-imbedding graph can be derived from the substructure query, the file structure, and the atom-to-atom mapping of the former onto the latter. Figure 1 illustrates this stepwise derivation for a simple query and a hypothetical matching structure. The query bonds are allowed to match either chain or ring bonds. The atom numbers of the query are shown throughout for clarity. The four steps are as follows: (A) create a strictly topological representation of the structure by changing its atoms to nodes, which are qualitatively indistinguishable, and its bonds to edges, likewise indistinguishable; (B) remove any edges that are not in a cycle and then remove any isolated nodes; (C) remove any edges that do not match a query bond and then remove any isolated nodes; (D) attach to each node as many new nodes as may be needed to make its degree equal to what it was after step B (the degree of a node is the number of edges to it). The graph remaining after step D is the ring-imbedding graph. Any new nodes attached in step D are terminal nodes, i.e., nodes with a degree of one. These will always be the only terminal nodes in the ring-imbedding graph since every node in the graph after step B has a degree of at least two.

The ring-imbedding graph may be a disconnected graph, like the example in Figure 1. It is also possible that the ring-imbedding graph will be the so-called empty graph, consisting of zero edges and zero nodes. This can happen only when an acyclic query is imbedded in an acyclic structure or when none of the bonds in an acyclic query is imbedded in any of the rings of a cyclic structure. In either case, all edges and nodes will be removed in step B or C. For answer sets in which the ring-imbedding graph is the empty graph for all retrieved structures, the proposed index can still be of value by indicating the complete absence of ring imbedding.

The ring-imbedding graph represents different kinds of information about the topology of the imbedding. The number of components of the graph is the number of separate rings or ring systems in which the query is imbedded (true only if the query consists of one substructure—which the present work assumes). The graph in Figure 1 consists of two components because part of the query is imbedded in a fused ring system and another part is in a separate ring. The ring-imbedding graph also represents what might be called the "complexity" of the imbedding. This complexity is reflected in the overall size of the graph, which relates to the number of query bonds imbedded in rings, and the extent of branching in the graph. Branching nodes, i.e., nodes with degree greater than two, in the ring-imbedding graph represent query atoms that act as bridgeheads or ring fusion sites in the file structure. A larger number of branching nodes in the graph suggests a more complex type of imbedding. The purpose of the terminal nodes attached in

[⊗] Abstract published in *Advance ACS Abstracts,* December 15, 1996.

S0095-2338(96)00093-5 CCC: $14.00

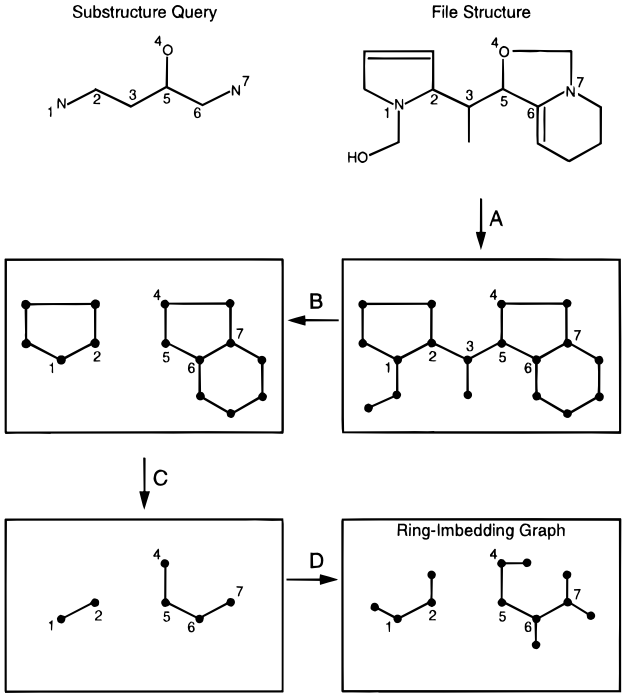

Figure 1. Derivation of the ring-imbedding graph for a substructure and a matching structure.

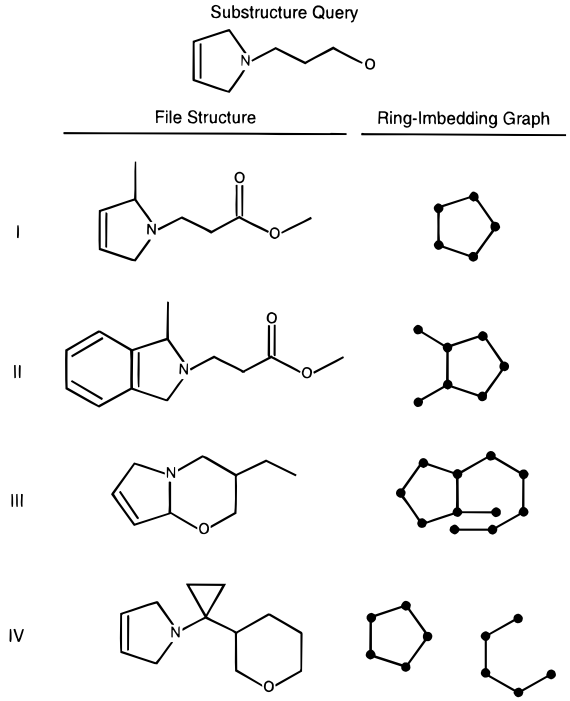

Figure 2. Ring-imbedding graphs for a cyclic substructure and several matching structures.

step D is to ensure that query atoms acting as bridgeheads or ring fusion sites display the appropriate amount of branching in the ring-imbedding graph. For example, query atoms 6 and 7 in Figure 1 are ring fusion atoms in the file structure shown. In the ring-imbedding graph, this ring fusion is represented by the fact that nodes 6 and 7 each have a degree of three, due to their attached terminal nodes.

It can be seen that the ring-imbedding graph contains a relatively small amount of structural information. The identity of specific elements and bond types is suppressed, and parts of the structure that are more distant from the query and not involved directly in the imbedding are omitted from the graph even though they may represent very important structural features. However, the purpose of the graph is to represent the basic topology of how a substructure query is imbedded in rings, and so the graph is deliberately focused on the query and its immediate environment. If the graph were to contain much more structural information, the specific topology of the imbedding might be lost, and, additionally, the index derived from the graph might be too discriminating to group structures together effectively, which is the intended purpose of the index.

To illustrate further the concept of a ring-imbedding graph, Figure 2 shows the graphs obtained for a cyclic query and several hypothetical matching structures. It is assumed that the acyclic bonds in the query are allowed to match either chain or ring bonds and that the query ring can be part of a ring system. In each graph appears a five-membered cycle corresponding to the heterocycle in the query. Rings in a query will always appear as cycles in the ring-imbedding graph. A cyclic query cannot yield a ring-imbedding graph that is the empty graph. With a cyclic query there is always some ring imbedding even though it may be "trivial," as in structure I where the query ring matches an isolated ring. In structure II, the query ring is imbedded in a ring system, and, as a result, the corresponding ring-imbedding graph has terminal nodes where the previous graph does not.

Structure III illustrates a case in which the file structure contains a bond between two query atoms (a ring carbon and the oxygen) that does not match a bond in the original query. In step C of the above procedure, the edge corresponding to this ring-closure bond is removed, so it does not appear in the ring-imbedding graph. This example illustrates the general rule that the ring-imbedding graph cannot contain more ring closures than the query.

In structure IV, two query bonds are imbedded in the tetrahydropyran ring. On the other hand, only a single atom of the query is imbedded in the cyclopropane. The query could be regarded as being imbedded in the cyclopropane ring, but in the present work a choice has been made not to recognize a ring imbedding unless at least one query bond is involved. The rationale for this choice is that ring imbedding at a single query atom will have much less impact on the conformation of the imbedded query. As can be seen from the ring-imbedding graph for structure IV, the edges and nodes corresponding to the cyclopropane ring have been removed from the graph in step C because none of the edges matches a query bond. As a result, the query is regarded as being imbedded in two, not three, separate rings. If one chooses instead to recognize ring imbedding at a single query atom, calculation of the ring-imbedding index can be modified accordingly (see below).

The ring-imbedding graph is only an intermediate step toward the derivation of an index. The index is based on a combination of two graph invariants. These invariants describe two aspects of the ring imbedding already mentioned: the number of separate rings or ring systems in which the query is imbedded and the complexity of the imbedding. It will be shown later that these invariants can be calculated from the values $N[1]$, $N[2]$, $N[3]$, ..., where $N[k]$ is the number of nodes in the ring-imbedding graph that have degree k. This is very convenient because the array N can be obtained by a simple algorithm that does not require the explicit construction of the graph.

The following algorithm to find N uses the substructure query and file structure connection tables and assumes that the connection table format distinguishes ring and chain bonds. The algorithm also uses the atom-to-atom mapping between the query and file structure; in the notation used below, f_i is the file-structure atom to which query atom q_i is mapped. The size of array N must be made large enough to handle the highest node degree likely to be encountered in a file structure (this must be greater than four even for organic chemistry because of heteroatoms like pentacoordinate phosphorus).

(1) initialize array N to 0
(2) for each query atom q_i
  (i) set $d1 = 0$
  (ii) for each query atom q_j connected to q_i
    (a) if f_j is connected to f_i by a ring bond, increment $d1$ by one
  (iii) if $d1 > 0$
    (a) set $d2 = 0$
    (b) for each file-structure atom connected to f_i by a ring bond, increment $d2$ by one
    (c) increment $N[d2]$ by one
    (d) if $(d2 - d1) > 0$, increment $N[1]$ by $(d2 - d1)$

As calculated in the "for" loop of step (2), $d1$ is the number of query bonds to q_i that map into ring bonds to f_i, and $d2$ is the total number of ring bonds to f_i. The degree of the node corresponding to q_i in the ring-imbedding graph is $d2$, so $N[d2]$ is increased by one; the number of terminal nodes to this node is $(d2 - d1)$, so $N[1]$ is increased by that amount. The purpose of statement (iii) is to skip, in effect, to the next query atom if $d1 = 0$. As a result, ring imbedding that does not involve at least one query bond will not be taken into account. As noted earlier, this can be changed. If the algorithm is modified so that it skips to the next query atom only if $d2 = 0$, ring imbedding at a single query atom will be taken into account.

For a given file structure, let RS be the number of separate rings or ring systems in which the query is imbedded in that structure. RS is one of the quantities to be incorporated into the ring-imbedding index. RS can be calculated from N, as found by the above algorithm, through a simple formula derived as follows. The number of rings, or cycles, in a connected graph is related to the numbers of edges and nodes by the well-known expression

$$rings = edges - nodes + 1 \tag{1}$$

For a disconnected graph, eq 1 can be applied to each of its connected components. The result is an expression for the number of rings in a graph with any number of components:[6]

$$rings = edges - nodes + components \tag{2}$$

This expression will now be applied to the ring-imbedding graph. The number of components in that graph is RS. It is clear that

$$nodes = \sum_{k=1}^{D} N[k] \tag{3}$$

where D is the size of array N, and that

$$edges = \frac{1}{2}\sum_{k=1}^{D} k \cdot N[k] \tag{4}$$

since the sum of the degrees of all nodes in a graph equals twice the number of edges. Because the ring-imbedding graph does not contain more ring closures than the query, it has the same number of rings as the query. Let QR be the number of rings in the query, as calculated by eq 1. Then eq 2 can be solved to give

$$RS = QR + \frac{1}{2}\sum_{k=1}^{D} (2 - k) \cdot N[k] \tag{5}$$

The other quantity needed to construct the ring-imbedding index, besides RS, is a measure of the complexity of the imbedding. As already discussed, the imbedding complexity is reflected in the size and branching of the ring-imbedding graph. This suggests that a quantitative measure of the topological complexity of that graph can be taken as a measure of the imbedding complexity. Many approaches to quantifying topological complexity have been proposed.[7] A measure that serves well for the present purpose is the simple one used by Gutman et al.: $\Sigma_i d_i^2$, where d_i is the degree of the ith node and the summation is over all nodes.[8] This measure can be applied to disconnected graphs as well, in which case it is essentially a sum over the components of the graph. This measure, applied to the ring-imbedding graph, gives the imbedding complexity, IC, which can be calculated as

$$IC = \sum_{k=1}^{D} k^2 \cdot N[k] \tag{6}$$

Since this quantity depends only on the degrees of the nodes, not their specific connectivity, a certain amount of degeneracy, i.e., the assignment of the same value to nonisomorphic graphs, is unavoidable. However, for the way in which IC is going to be used, some degeneracy is acceptable.

RS and IC are combined to give the desired index, called RIQI (for Ring-Imbedding-of-Query Index), by the formula

$$\mathrm{RIQI} = RS \times 10^5 + IC \tag{7}$$

For the example in Figure 1, RIQI = 200 040. The RIQIs for structures I−IV in Figure 2 are 100 020, 100 032, 100 048, and 200 034, respectively. For an acyclic query that is not ring imbedded, all entries of N equal zero (as expected for the empty graph), and so the RIQI will equal zero.

RIQI values calculated for a multiring query are shown in Figure 3. The number of rings in this query is five. In structure I, the query is imbedded in four separate rings or ring systems, while in structure II the query is imbedded in one ring system; this difference is reflected in the first digit of their respective RIQIs. The query is also imbedded in one ring system in structure III, but in this case it is due to metal coordination of the query. Coordination to a metal can create cycles in a chemical graph, causing some bonds in the connection table to be identified as ring bonds only because of this coordination.[9] The possible effects of metal

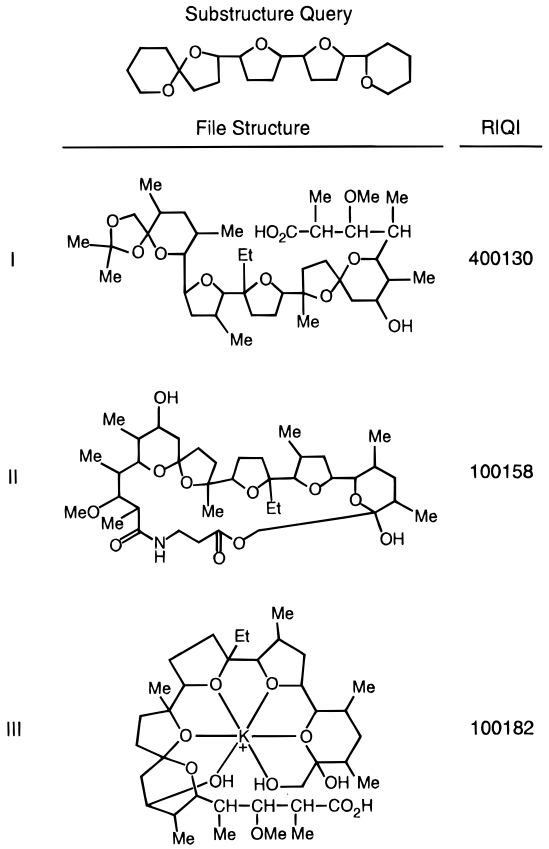

Figure 3. RIQI values calculated for a multiring substructure and some matching structures.

coordination need to be recognized when analyzing the ring imbedding of a substructure.

A common device in substructure searching is the use of "variable" atoms or bonds in the query. These are allowed to match more than one element (e.g., any halogen) or bond type. Since the identity of specific elements and bond types is not used in the above calculation, the RIQI is applicable to such queries as long as there is an atom-to-atom mapping between the query and the file structure.

RESULTS AND DISCUSSION

To analyze a substructure-search answer set with the RIQI, each structure must be assigned a RIQI value. For each structure, *N* is calculated by the above algorithm and then used, along with the precomputed value of *QR*, to calculate the RIQI by eqs 5−7. After each structure has been assigned a RIQI value, all the structures can be sorted according to their RIQI. Each set of structures having the same RIQI will be called a *RIQI class*.

The substructure search program used in this work is set to find only one occurrence of the query in the file structure (since one occurrence is sufficient to make the structure a valid answer). For this reason, only one atom-to-atom mapping of query to file structure was used to calculate the RIQI even though the query might occur in the structure more than once. Each structure is thus constrained to belong to only one RIQI class. This could result in a loss of information about ring imbeddings. A structure might contain two or more imbeddings with different RIQIs, in which case it should belong to more than one RIQI class. This limitation could be avoided by setting the search program to find all possible mappings between query and structure.

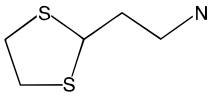

Figure 4. Substructure query used to obtain the test answer set.

Table 1. RIQI Classes in the Test Answer Set

class no.	RIQI	answers	class no.	RIQI	answers
1	100020	79	11	100058	70
2	100038	4	12	100060	9
3	100040	1	13	100064	21
4	100042	21	14	100066	8
5	100044	1	15	100072	5
6	100046	157	16	200030	40
7	100050	10	17	200034	15
8	100052	68	18	200036	4
9	100054	1	19	200040	29
10	100056	2	20	200046	8

To illustrate answer-set analysis with the RIQI, a test answer set was obtained by a search of the CAS Registry File. The search query was based on the substructure shown in Figure 4. The acyclic bonds were allowed to match either chain or ring bonds in the file structures, and the query ring was allowed to be part of a ring system. The search produced an answer set of 553 different Registry Numbers. This answer set contains 20 RIQI classes. These classes, sorted by RIQI value, and their sizes are listed in Table 1.

The distribution of RIQI class sizes in Table 1 is highly uneven. The ten smallest classes contain less than 8% of the answers. Such an uneven distribution has certain consequences for answer-set sampling. A random sample of this answer set would be dominated by structures from the largest RIQI classes. Structures from the smallest classes, which represent types of ring imbeddings that are relatively rare, might not even appear in the sample. Thus, a random sample may not adequately represent the diversity of ring imbeddings in a large answer set.

The analysis of an answer set into RIQI classes provides a more direct way to sample ring-imbedding diversity. This can be done by taking a small sample from each of the RIQI classes and merging these to form a larger answer-set sample. The RIQI classes could be sampled equally or in some proportion to their size. Either way, every class in the original answer set will be represented in the final sample. This sort of procedure was applied to the test answer set by selecting just one structure from each of the 20 RIQI classes (three classes contain only one structure). The sample created is shown in Figure 5. The selection of these structures was done manually by inspecting all the structures in each class, or only the smallest structures in the largest classes, and choosing one that seemed representative of the class and could also be displayed in a compact diagram. The selection of representative structures from a RIQI class could certainly be automated by either choosing structures at random or based on features characteristic of that particular class (e.g., commonly occurring ring topologies).

The sample in Figure 5 gives a concise overview of the wide variety of ring imbeddings in the test answer set. Because the structures are ordered by RIQI value, there can

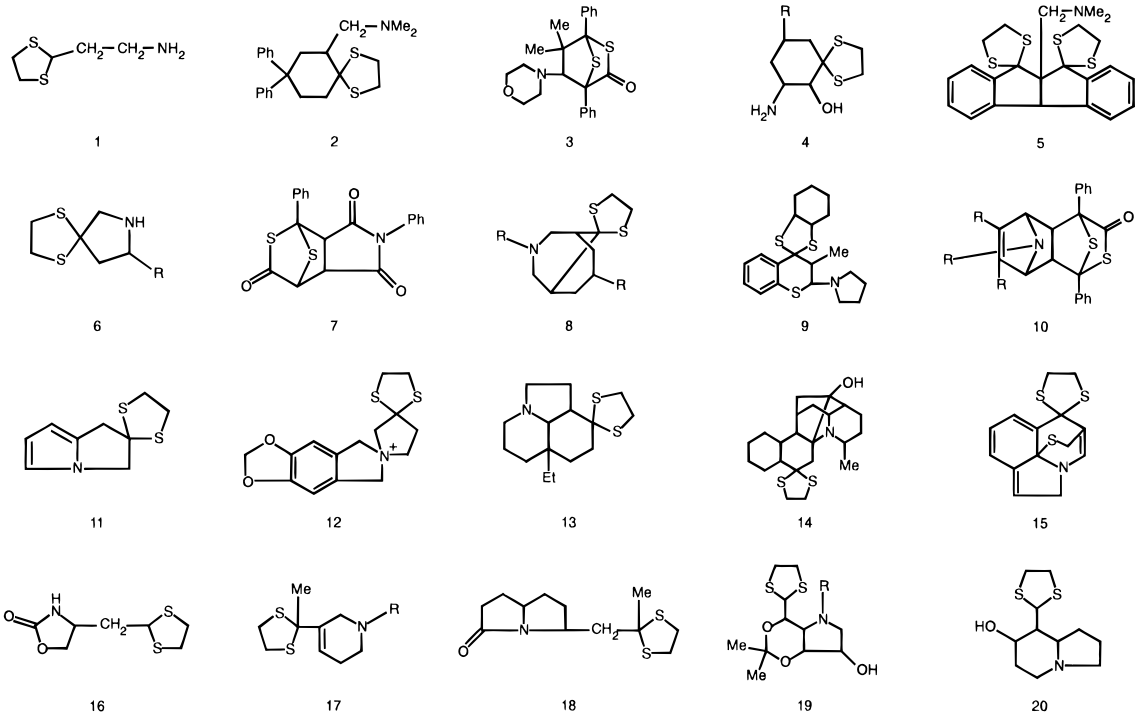

Figure 5. Answer-set sample created by selecting one structure from each of the twenty RIQI classes (see Table 1) in the test answer set. The corresponding class number is shown below the structure. Some substituents have been replaced by R to compress the diagram.

be seen a general progression in the complexity of the imbedding in going from structures 1 to 15. In these structures, the query is imbedded in one ring system. A progression in imbedding complexity is also apparent in going from structures 16 to 20. In these, the query is imbedded in two separate rings or ring systems.

Structures that appear in exceptionally small RIQI classes represent types of ring imbeddings that are very rare in the answer set but are nevertheless made noticeable by the RIQI classification. Even a type of imbedding that occurs only once in a large answer set will be noticeable if it receives a distinctive RIQI value and thereby falls into its own class (like classes 3, 5, and 9 in Table 1). The structures that display the rarest types of imbeddings are a possible source of structural novelty. Direct examination of the structures in the smallest RIQI classes may therefore be useful as a way to find novel structures in an answer set.

To illustrate the variety of structures that can get grouped together in a RIQI class, six structures from class 20 are shown in Figure 6. These structures, together with structure 20 in Figure 5, constitute all the topologically different structures in this class (the original eight answers include a pair of stereoisomers). As expected, these structures differ outside the immediate environment of the query since the RIQI is not sensitive to that part of a structure. These structures would not necessarily be considered similar on the basis of their overall topology, but they can be considered similar in a "local" sense, i.e., the imbedded queries in these structures are alike in terms of their relationship to the local topology of the structure. The structures in this RIQI class would probably not be grouped together by cluster analysis using a conventional similarity measure because such measures take into account all parts of a structure. By being able to group structures on the basis of "local" similarity, RIQI analysis may serve as a complement to conventional similarity analysis.

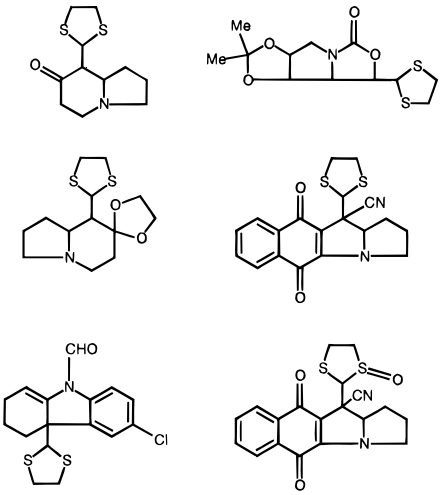

Figure 6. Structures from RIQI class 20.

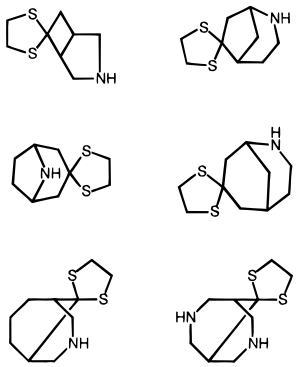

Figure 7. Similar structures from RIQI class 8.

In some cases, structures in the same RIQI class will display a similarity that is more global in nature. For instance, a manual inspection of class 8 revealed the structures shown in Figure 7. The similarity of these ring systems is apparent. This suggests that the RIQI classifica-

tion of a substructure-search answer set may sometimes be able to highlight similar structures by grouping them together in the same class.

CONCLUSION

The RIQI is a potentially useful tool for organizing and analyzing a substructure-search answer set in terms of the ring imbedding of the query. The classification of answer-set structures according to their RIQI can help to sample large answer sets in such a way that the diversity of ring imbeddings is well represented. The RIQI may also help in searching for novel structures in an answer set since it can make rare types of ring imbeddings noticeable. Direct examination of the structures in the smallest RIQI classes may be useful for this purpose.

ACKNOWLEDGMENT

The author would like to thank Steven Layten for his implementation of the RIQI algorithm.

REFERENCES AND NOTES

(1) Bonchev, D.; Mekenyan, O.; Trinajstić, N. Topological Characterization of Cyclic Structures. *Int. J. Quantum Chem.* **1980**, *17*, 845−893.

(2) Mekenyan, O.; Bonchev, D.; Trinajstić, N. Algebraic Characterization of Bridged Polycyclic Compounds. *Int. J. Quantum Chem.* **1981**, *19*, 929−955.

(3) Randić, M. Ring ID Numbers. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 142−147.

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

(5) Bonchev, D.; Balaban, A. T.; Liu, X.; Klein, D. J. Molecular Cyclicity and Centricity of Polycyclic Graphs. I. Cyclicity Based on Resistance Distances or Reciprocal Distances. *Int. J. Quantum Chem.* **1994**, *50*, 1−20.

(6) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1969; p 39.

(7) Bonchev, D. The Problems of Computing Molecular Complexity. In *Computational Chemical Graph Theory*; Rouvray, D. H., Ed.; Nova Science Publishers: New York, 1990; pp 33−63.

(8) Gutman, I.; Ruscić, B.; Trinajstić, N.; Wilcox, C. F., Jr. Graph Theory and Molecular Orbitals. XII. Acyclic Polyenes. *J. Chem. Phys.* **1975**, *62*, 3399−3405.

(9) For example, all the bonds in the CAS connection table for ethylenediamine are chain bonds, whereas all the bonds for the ethylenediamine-manganese chelate are ring bonds.

CI960093X