Heuristics for Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm

John W. Raymond

Pfizer Global Research and Development, Ann Arbor Laboratories, 2800 Plymouth Road, Ann Arbor, Michigan 48105

Eleanor J. Gardiner and Peter Willett*

Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom

Received September 25, 2001

Recently a method (RASCAL) for determining graph similarity using a maximum common edge subgraph algorithm has been proposed which has proven to be very efficient when used to calculate the relative similarity of chemical structures represented as graphs. This paper describes heuristics which simplify a RASCAL similarity calculation by taking advantage of certain properties specific to chemical graph representations of molecular structure. These heuristics are shown experimentally to increase the efficiency of the algorithm, especially at more distant values of chemical graph similarity.

INTRODUCTION

The concept of chemical structure similarity is widely used but poorly defined, with many divergent approaches having been proposed.[1−3] Few have been as effective as those measures operating on a graphical representation of chemical structure, with most of these employing a bit string description of chemical structures (i.e., fingerprints) represented as molecular graphs. Similarity comparisons involving chemical fingerprints are computationally simple as well as effective in practice.[2]

There exists, however, an alternative approach to chemical graph similarity that has attracted much less attention to date, viz. cost-based approaches to similarity,[4] of which those premised on the maximum common subgraph (MCS) are the most prevalent.[5−7] When applied to chemical graphs (see e.g., refs 8−10), these techniques have the desirable property that the calculated similarity measure is intuitive and easily visualized (i.e., highlighting the maximum common subgraph between two molecular graphs), but they have suffered historically due to the computational intractability inherent in the MCS problem.

A new, efficient, MCS-based similarity algorithm, RASCAL, has recently been proposed[11] that is based on the reduction of the MCS problem to the maximum clique problem. Since the maximum clique in a modular product graph corresponds to the MCS between the two factor graphs used to construct the modular product, RASCAL establishes an edge induced formulation of the MCS by first transforming the two chemical graphs under consideration into their corresponding line graphs, constructing the modular product graph, and then determining the maximum clique in the modular product graph. The algorithm is sufficiently general to be applied to any undirected graph and consequently does not take optimum advantage of some of the properties specific to molecular graphs in order to increase the efficiency of the algorithm for chemical applications. In this paper, we propose several simplification heuristics that can significantly improve the performance of the original RASCAL procedure.

TERMINOLOGY

All graphs referred to in the following text are assumed to be simple, undirected graphs. A graph G consists of a set of vertices V(G) and a set of edges E(G). The vertices in G are connected by an edge if there exists an edge $(v_i,v_j) \in$ E(G) connecting the vertices v_i and v_j in G such that $v_i \in$ V(G) and $v_j \in$ V(G). In 2D chemical graphs, the vertices of the graph correspond to the atoms of the molecule, and the edges represent the chemical bonds.

The number of vertices will be denoted by |V(G)|. A vertex v_i is said to be incident with an edge in G if one of the two endpoints in the edge is v_i. Two vertices v_i and v_j are adjacent if vertex v_i and vertex v_j are connected by an edge in E(G). Two edges are said to be incident if they have a vertex in common. The set of vertices adjacent to a vertex v_i are referred to as the neighbors of v_i, N(v_i). The degree of a vertex v_i is the number of edges with which it is incident, symbolized by d(v_i). The ratio of the number of edges in a graph G relative to the maximum number of edges possible in a graph with |V(G)| vertices is referred to as the edge density (i.e., $2 \cdot |E(G)|/|V(G)| \cdot ((|V(G) - 1|))$).

A subgraph H of a graph G is a graph whose set of vertices and set of edges satisfy the relations: $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, and if it can be shown that $V(H) = V(G)$ and $E(H) = E(G)$ such that all adjacencies are preserved. If a subgraph H of G with the vertex set $V(H) \subseteq V(G)$ and edge set $E(H) \subseteq E(G)$ is such that the edge set E(H) contains all edges in E(G) that are incident on V(H) when each vertex in V(H) is

* Corresponding author phone: 44 1142-768 555 x5080; fax: 44 1142 780 300; e-mail: p.willett@sheffield.ac.uk.

10.1021/ci010381f CCC: $22.00
Published on Web 02/09/2002

mapped to a vertex in V(G), then subgraph H is an induced subgraph of G. An induced subgraph in which all vertices of the subgraph are incident on each other is called a clique. A maximal clique in a graph G is a clique of a graph that is not a subgraph of a larger clique in G. A maximum clique is the largest maximal clique induced in G. The size of the maximum clique is usually represented as $\omega(G)$.

A common induced subgraph between two graphs G_1 and G_2 is a subgraph H with vertices $V(H) \subseteq V(G_1)$ and $V(H) \subseteq V(G_2)$ such that the subgraphs induced in both G_1 and G_2 are isomorphic. The subgraph is maximal if it is not a subgraph of a larger subgraph possessing this property. The subgraph is referred to as a maximum common subgraph (MCS) if there is no other subgraph of greater cardinality meeting the aforementioned criteria.[12] Related to the MCS is the maximum common edge subgraph (MCES), which is a set of edges where $E(H) \subseteq E(G_1)$ and $E(H) \subseteq E(G_2)$ such that the edges induce isomorphic subgraphs in G_1 and G_2 of maximum cardinality.[12] Note that the MCS or MCES between two graphs is not necessarily a connected graph by definition. The MCES has also been referred to as the maximum overlapping set (MOS) in the chemical information community,[13] and any further descriptions of this problem will adopt this nomenclature.

MODULAR PRODUCT GRAPHS

The detection of the MCS between two graphs G_1 and G_2, MCS(G_1,G_2), can be reduced to one of determining the maximum clique in a compatibility graph. This concept has been independently discovered on several occasions. Levi[14] and Barrow and Burstall[15] appear to have been among the first to suggest the use of the compatibility graph for the determination of MCS(G_1,G_2) which later served as the impetus for the chemical MCS program of Cone et al.[9] Vizing,[16] independent of the aforementioned authors, identified the compatibility graph as a graph product. He dubbed this graph product as the modular product which will be denoted as $G_1 \diamond G_2$. The modular product graph is sometimes referred to as an association graph rather than a compatibility graph.[17] The modular product of two graphs G_1 and G_2 is defined on the vertex set $V(G_1 \diamond G_2) = V(G_1) \times V(G_2)$ with two vertices (u_i,v_i) and (u_j,v_j) being adjacent whenever

$$(u_i,u_j) \in E(G_1) \text{ and } (v_i,v_j) \in E(G_2)$$

or

$$(u_i,u_j) \notin E(G_1) \text{ and } (v_i,v_j) \notin E(G_2)$$

Figure 1 illustrates the modular product of two P_3 path graphs. In Figure 1, vertex (u_1,v_1) is adjacent to vertex (u_2,v_2) in the modular product graph since vertices u_1 and u_2 are adjacent in graph G_1, and vertices v_1 and v_2 are adjacent in graph G_2. Vertex (u_1,v_1) is also adjacent to vertex (u_3,v_3) since vertices u_1 and u_3 are not adjacent in graph G_1, and vertices v_1 and v_3 are not adjacent in graph G_2. However, vertex (u_1,v_1) is not adjacent to vertex (u_3,v_2) since vertices u_1 and u_3 are not adjacent in graph G_1, and vertices v_1 and v_2 are adjacent in graph G_2. It is also clear that vertex (u_1,v_1) is not adjacent to vertex (u_1,v_2) since in a chemical graph, a vertex cannot be incident on itself.

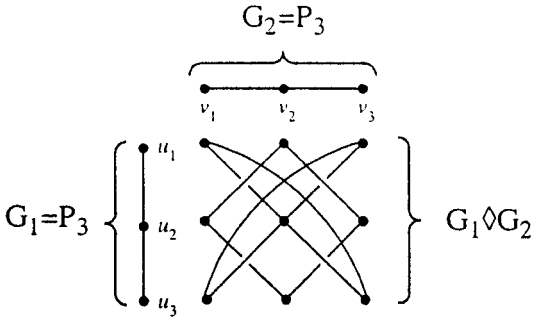

Figure 1. Modular product $P3 \diamond P3$.

The two maximum cliques of cardinality three in Figure 1 correspond to the two maximum vertex induced common subgraphs between graph G_1 and graph G_2, which, in this case, also happen to be isomorphic mappings. Skorobogatov and Bessonov developed a set of algorithms capable of detecting the maximum common subgraph between graphs based on the modular product concept.[18,19] Bessonov subsequently extended the modular product MCS method to 3D graphical representations of molecules accounting for translation and rotation.[20]

To apply the modular product to the problem of the MCS of two chemical graphs, it is first necessary to extend the definition of the modular product to allow for weighted graphs. Since each vertex corresponds to a labeled atom type in a chemical graph denoted $w(v_i)$, a vertex (u_i,v_i) exists in $V(G_1 \diamond G_2)$ if and only if $w(u_i) = w(v_i)$. Similarly for the edge labels $w(u_i,u_j)$ corresponding to bond types in a chemical graph, an edge exists between two vertices (u_i,v_i) and (u_j,v_j) in the modular product $G_1 \diamond G_2$ whenever

$$(u_i,u_j) \in E(G_1) \text{ and } (v_i,v_j) \in E(G_2) \text{ and } w(u_i,u_j) = w(v_i,v_j)$$

or

$$(u_i,u_j) \notin E(G_1) \text{ and } (v_i,v_j) \notin E(G_2)$$

The modular product of two simple chemical graphs is demonstrated in Figure 2(a),(b). The two MCS cliques of cardinality four are highlighted in Figure 2 [parts (a) and (b), respectively], by the hollow circles in the modular product graph. These correspond to the maximum common subgraphs indicated by the bold substructures in each chemical graph. This figure serves to illustrate one of the limitations of using a vertex induced MCS approach to chemical structure analysis. Note that both MCS structures contain four atoms, but the MCS in Figure 2(a) contains one more bond than does the MCS in Figure 2(b). From a chemistry perspective, the MCS in Figure 2(a) more adequately describes the similarity between the two molecules in question, but a vertex induced MCS procedure does not distinguish between subgraphs based upon edge set cardinality.

Another significant limitation imposed by the use of an MCS procedure is the fact that a significant majority of the atoms in a chemical graph are carbon.[21] This will cause the number of vertices in the modular product to increase dramatically as the sizes of the chemical graphs being compared increase since most of the atoms (i.e., carbon) in one molecule will correspond to most of the atoms in the

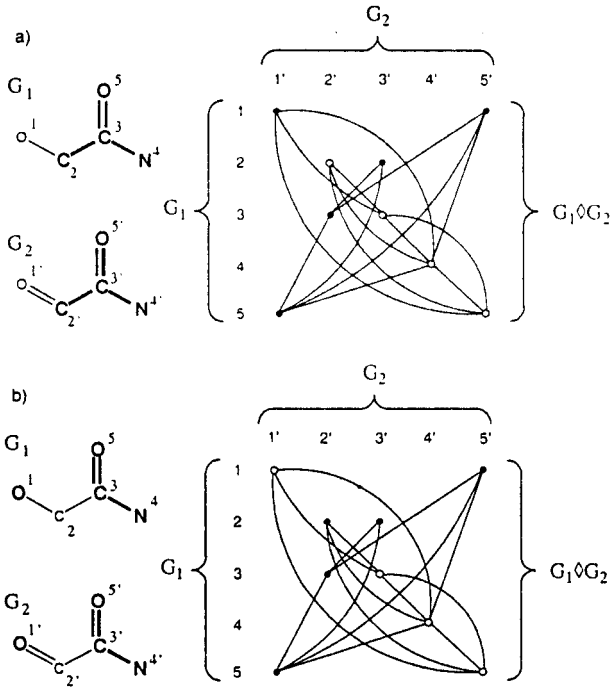

Figure 2. Modular product of chemical graphs (a) MOS = {(2,2′), (3,3′), (4,4′), (5,5′)} and (b) MOS = {(1,1′), (3,3′), (4,4′), (5,5′)}.

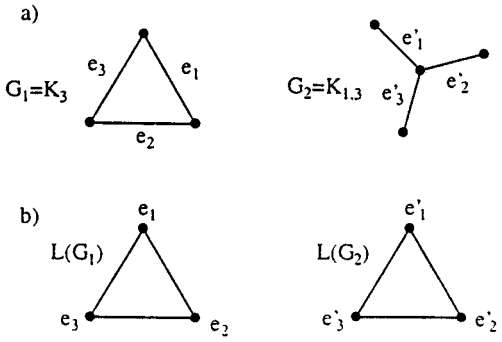

Figure 3. $K_3 \leftrightarrow K_{1,3}$ interchange.

other molecule. Since molecular graphs are also sparse, most atoms are not adjacent to the other atoms in the graph. Therefore, the term, $(u_i,u_j) \notin E(G_1)$ and $(v_i,v_j) \notin E(G_2)$, in the modular product will dominate, and the modular product graph will be dense as well as large.

Edge-Induced Isomorphism. As mentioned previously, it has been shown that the maximum common subgraph (MCS) problem and subsequently isomorphism and subgraph isomorphism can be reduced to operations on the modular product between two graphs.[14−16] This approach has two major limitations: the limitation of a vertex induced representation of the MCS and the dramatic increase in computational difficulty as the size of the molecules under consideration increase. Fortunately, there are techniques that can be employed that resolve these issues to some degree in a large number of cases.

One such technique is based upon the pioneering work of Whitney,[22] who proved that an edge isomorphism between two graphs G_1 and G_2 induces a vertex isomorphism provided that a $K_3 \leftrightarrow K_{1,3}$ interchange does not occur. This can be described with the aid of the example depicted in Figure 3. Figure 3(a) shows two graphs $G_1 = K_3$ and $G_2 = K_{1,3}$, respectively. It is evident by visual inspection that the two graphs in Figure 3(a) are not isomorphic.

A line graph $L(G_1)$ is a graph whose vertex set consists of the edge set of G_1; therefore, if (v_i,v_j) is an edge in G_1 it is also a vertex in $L(G_1)$. A pair of vertices in $L(G_1)$ is adjacent if the two corresponding edges in G_1 are incident on a common vertex. Figure 3(b) presents the line graphs of G_1 and G_2, respectively, and it is clear by inspection that the line graphs are isomorphic. This is called a $K_3 \leftrightarrow K_{1,3}$ interchange. Whitney proved that provided that a $K_3 \leftrightarrow K_{1,3}$ interchange does not occur, an isomorphism between two line graphs $L(G_1)$ and $L(G_2)$ induces an edge isomorphism between the root graphs (G_1 and G_2) of the two line graphs.

Nicholson et al.[23] first suggested the use of Whitney's theorem for use in applications involving the MCS of chemical graphs. Kvasnicka and Pospichal[24] extended this idea and published a very readable implementation of the resulting theorem for application to the MCS problem. This line graph induced isomorphism concept has served as the basis for the development of the MCS program TopSim.[25,26] Independently of that work, Chen and Yun[27] have also developed an algorithm based on these principles, but they were apparently unaware of the work of Whitney,[22] Nicholson et al.,[23] and Kvasnicka and Pospichal.[24]

For the purposes of clarity and consistency, it must be stated that the maximum clique in the modular product of line graphs does not yield the MCS of two root graphs G_1 and G_2. The maximum clique obtained from the modular product of two line graphs is more appropriately described as the maximum common edge subgraph (MCES) or the maximum overlapping set (MOS) of the two root graphs G_1 and G_2.

The distinction is illustrated with the example in Figure 4. The chemical graphs G_1 and G_2 used in the example of Figure 2 are presented in Figure 4(a), and their respective line graphs are depicted in Figure 4(b). Since molecular graphs are weighted (labeled), their respective line graphs are also weighted. Each vertex in the line graph $L(G_1)$ is weighted by its respective edge and vertex endpoint weights in G_1. Take for instance edge number 1 in graph G_1 of Figure 4(a). This corresponds to vertex number 1 in the line graph $L(G_1)$ in Figure 4(b). This vertex is weighted by the bond pair C−O.

The edges in a weighted line graph are also weighted. Each edge between two vertices, v_i and v_j, in a weighted line graph $L(G_1)$ is labeled with the vertex label of the incident vertex in G_1 between the two edges in G_1 corresponding to v_i and v_j. For instance, edges 3 and 4 in graph G_1 of Figure 4(a) correspond to vertices 3 and 4 in the line graph $L(G_1)$ in Figure 4(b). Since edges 3 and 4 in G_1 are incident on an atom of type carbon (C), then the corresponding edge in $L(G_1)$ is also labeled as type carbon.

Since a weighted line graph can then simply be assumed to be an arbitrary weighted graph, the modular product can be constructed as previously described. The modular product graph for the line graphs depicted in Figure 4(b) is presented in Figure 4(c). Since no $K_3 \leftrightarrow K_{1,3}$ interchange has occurred, the clique depicted in the modular product graph corresponds to the maximum overlapping set (MOS) between the two chemical graphs in Figure 4(a). The MOS is highlighted in bold face in the chemical graphs.

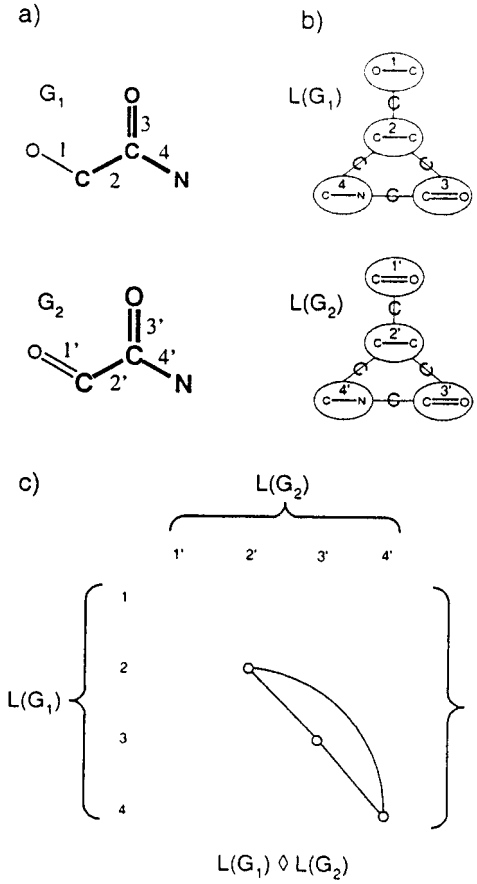

Figure 4. Modular product of line graphs.

This example illustrates two desirable features of the line graph approach. First, it determines the MOS between two graphs rather than the MCS. Since the MOS tends to more adequately describe chemical structure similarity, it appears to have advantages over the vertex induced approach. Second, as can be seen by the disparity in complexity between the modular products depicted in Figures 2 and 4, the line graph technique has the potential to reduce significantly the size of the modular product graph. Since the maximum clique problem is NP-complete and all known exact algorithms are of exponential complexity, this can mean the difference between being able to detect the maximum clique efficiently and not being able to calculate it at all.

Since it is well-known that the complexity of clique detection increases dramatically as the number of vertices and/or the number of edges in a graph increase, the simplest and most effective heuristics would obviously be those that involved the deletion of vertices or edges in the modular product graph prior to initiating clique detection. A previous attempt by Brown[13] to make the modular product approach tractable in practice categorized carbon atoms into three distinct types: (1) chain atoms, (2) ring atoms of connectivity two, and (3) ring atoms of connectivity three. Carbon atoms of one type were then not allowed to correspond to carbon atoms of another type. Even using this restrictive constraint with the TopSim[25] algorithm, Brown found that the exact MOS approach was so slow that he abandoned it for an approximate structure-matching method based upon genetic algorithms.[28]

Brown's heuristic is too restrictive to be useful in practice, as it is not difficult to find molecules where the MOS solution

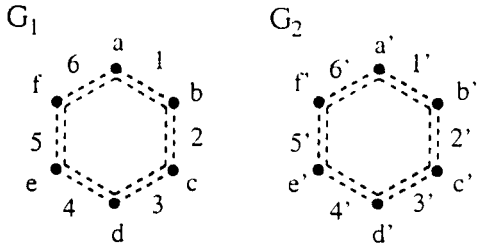

Figure 5. Benzene rings.

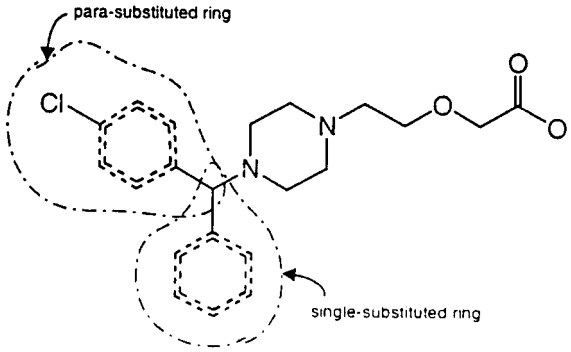

Figure 6. Cetirizine.

calculated using this heuristic is arbitrarily poor with respect to the true MOS solution. The objective of the rest of this paper is to develop heuristics to simplify the modular product without placing unrealistic restrictions on the potential MOS between two molecular graphs. Fortunately, chemical graphs possess some characteristics that facilitate the development of the vertex and edge deletion heuristics in this paper.

NODE DELETION HEURISTICS

Node deletion heuristics are among the most advantageous pruning heuristics because they reduce not only the number of vertices but also any incident edges as well. This can have a dramatic increase on the efficiency of clique detection.

Benzene Ring Symmetry. One common constituent in chemical structures is the benzene ring. In this context the benzene ring can be assumed to represent a fixed local structure within a molecule. The symmetry of a benzene ring provides an ideal mechanism for pruning vertices from the modular product graph. Take, for instance, the two benzene rings in Figure 5. Ring G_1 can be mapped onto G_2 in 12 different ways. This includes the six mappings obtained by rotating graph G_1 onto G_2 in the plane of the paper (e.g., (1,1′), (2,2′), (3,3′), (4,4′), (5,5′), (6,6′) and (1,2′), (2,3′), (3,4′), (4,5′), (5,6′), (6,1′)). Additionally other mappings can be obtained by rotating the ring about an axis out of the plane of the paper at each of these six mappings. For example, the mapping (1,1′),(2,2′),(3,3′),(4,4′),(5,5′),(6,6′) will yield (1,6′),(2,5′),(3,4′),(4,3′),(5,2′),(6,1′) if it is rotated 180° out of the plane of the paper along the axis defined by a line running from vertex *a* to vertex *d* in G_1 and *a*′ to *d*′ in G_2.

Many molecules contain benzene rings in the form of a singly substituted or para-substituted benzene ring. Figure 6 depicts the antihistamine compound cetirizine possessing both of these ring types. Note that both rings have the property that the rotation out of the plane of the paper previously described is redundant. Let us assume that it is desired to map the para-substituted benzene ring in Figure

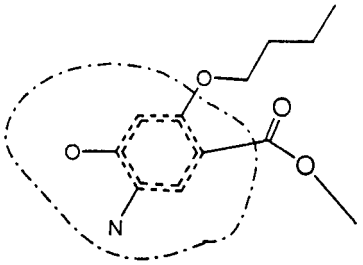

Figure 7. Orthocaine.

6 onto the benzene ring in the molecule orthocaine depicted in Figure 7.

Figure 8 illustrates two potential mappings of the cetirizine benzene ring onto the orthocaine benzene ring. Both are symmetric with respect to rotation out of the plane of the paper. Clearly, the two mappings are redundant in that it is physically impossible to distinguish between the two mappings in their respective parent molecules. Therefore, it is only necessary to consider one of the mappings when attempting to establish the MOS between cetirizine and orthocaine. Concerning the modular product, this means that it is only necessary to retain the nodes corresponding to one of these mappings within the modular product graph. If it is arbitrarily assumed that the first mapping is to be retained, then the vertices in the modular product graph corresponding to (6,1′), (5,2′), (4,3′), (3,4′), (2,5′), and (1,6′) could be deleted without affecting the MOS between the two molecular graphs provided that they are not required for some other valid benzene-ring-to-benzene-ring mapping. This heuristic is basically an attempt at addressing the concept of localized equivalence classes in an MOS algorithm.

This analysis has assumed that both rings were benzene rings, but the same procedure is valid for many other types of rings exhibiting similar symmetry. It is also equally valid if one of the rings contains a heteroatom. Since one ring will consist of all carbon atoms and the other contains a heteroatom such as nitrogen, none of the bonds (edges) incident on the heteroatom in a mapping from the benzene ring to the heteroatom ring will match regardless of the orientation of the mapping. It can also be used if one of the rings happens to be a fused aromatic ring.

Ring Substitution Alignment. Since the valence of carbon is four, any carbon in an aromatic ring can have at most one other atom not in the aromatic ring attached to it. In other words, it can have only one substitution. This observation presents an effective node pruning procedure for aromatic rings that is independent of whether the ring is singly or para-substituted.

If a benzene ring subgraph in a graph G_1 and a benzene ring subgraph in a graph G_2 both have a substitution of the same atom type, then only those benzene-ring-to-benzene-ring mappings which maintain a common atom substituent type need be considered valid. Nodes in the modular product graph corresponding to invalid mappings can be deleted provided that they do not also pertain to some other valid mapping. If the two rings have no common substituent atom types in common, then any one arbitrary mapping needs to be retained while discarding the other potential mappings since the maximum mapping that could possibly occur is simply the aromatic ring itself without any substituents.

Figure 9 demonstrates the ring alignment heuristic. Revisiting the two benzene rings of Figure 8, Figure 9 depicts the 12 possible ring mappings. Since both rings have a carbon atom substitution, the aforementioned heuristic is applicable. Of the possible 12 mappings, only mappings ⑤ and ⑥ preserve a substituent atom type (i.e., carbon); therefore, the other mappings can be discarded. For each discarded mapping, it must be determined which bond-to-bond matchings can be eliminated from the modular product graph by checking to make sure that it is not required for some other valid ring mapping. For instance, bond to bond matching (4,1′) in mapping ① is not present in either of the two valid mappings, ⑤ and ⑥. The node corresponding to matching (4,1′) can then be deleted from the modular product graph without affecting the MOS.

Note that the list of potential mappings in Figure 9 can be further pruned by employing the previous benzene ring symmetry heuristic. Mappings ⑤ and ⑥ correspond to a 180° rotation out of the plane of the paper for a para-substituted benzene ring. Therefore if we arbitrarily select mapping ⑤ as a default valid mapping, then the nodes in the modular product graph corresponding to bond-to-bond mappings (1,6′), (2,5′), (3,4′), (4,3′), (5,2′), and (6,1′) from mapping ⑥ can potentially be deleted. Since mapping ⑤ is the only valid mapping left, it is clear that these nodes are not required for another valid mapping. They can then be deleted from the modular product graph.

It is evident upon inspection that establishing this locally optimal mapping cannot result in a sub-optimal global mapping. Since all benzene-ring-to-benzene-ring mappings that preserve substituted edges (i.e., (7,7′) in ⑤) are conserved in the proposed heuristic, any globally optimal solution that requires the specified benzene-ring-to-benzene-ring mapping will remain valid. If, for instance, the benzene-ring-to-benzene-ring mapping in the globally optimal solution does not include any of the substituted edges (i.e., the optimal solution does not preserve a substituted edge), then the optimal benzene-ring-to-benzene ring mapping is actually a subgraph of the conserved mapping in ⑤. This simply means that there is a maximum clique in the modular product that does not include the node specified by (7,7′); therefore, the benzene ring is still completely conserved in the MOS.

As with the previous heuristic, the ring alignment pruning procedure can be adapted to situations where one of the rings

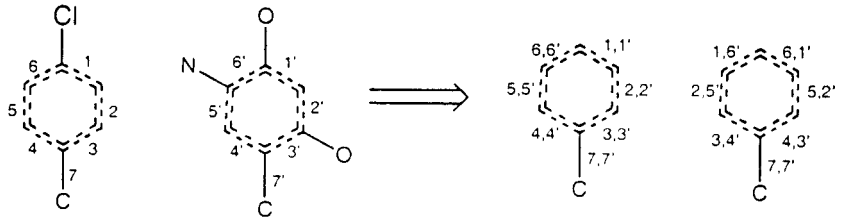

Figure 8. Redundant mapping.

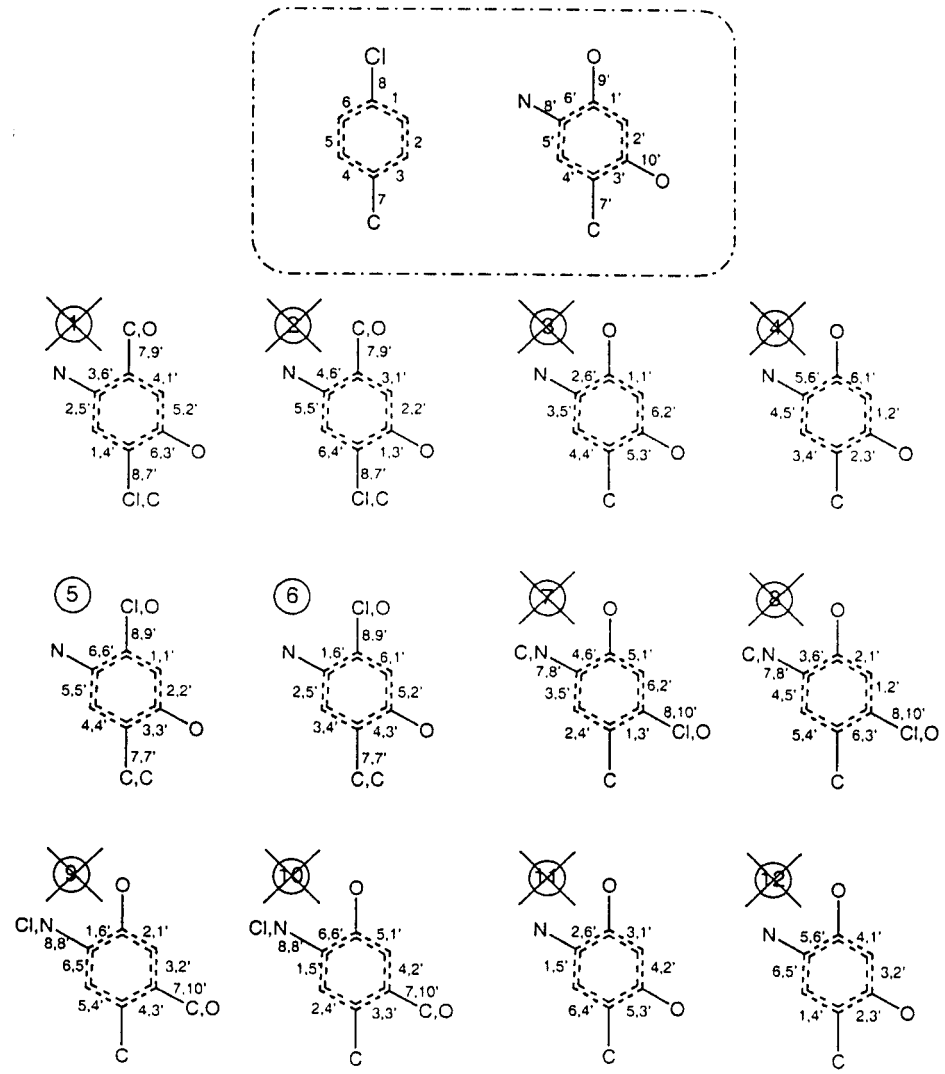

Figure 9. Ring alignment mapping.

is a nonbenzenoid ring such as pyrrole and/or the rings are fused. It can also be used when the rings are not necessarily the same size.

EDGE DELETION HEURISTICS

The previous node deletion heuristics took advantage of the properties of a molecular graph in order to reduce the size of the modular product graph. The following edge deletion heuristics approach the problem from the perspective of the molecular structure rather than the chemical graph.

Weak Ring Heuristic. Ring systems are of considerable importance in molecular structures, especially those of pharmaceutical and environmental interest, providing a skeletal framework on which to place active substructures or serving as pharmacophoric substructures in their own right.

The proposed weak ring technique considers those rings in a molecule with seven or fewer members and considers a potential edge between two vertices (u_i,v_i) and (u_j,v_j) in the modular product graph. The weak ring heuristic takes effect if (1) both edges in graph G_1 (respectively, G_2) corresponding to u_i and u_j (respectively, v_i and v_j) are ring bonds in the *same* ring of seven or fewer members and (2) either u_i or u_j (respectively, v_i or v_j) is an unfused bond (indicating the two bonds correspond to a single unambiguous ring), and (3) both edges in graph G_2 (respectively, G_1) corresponding to v_i and v_j (respectively, u_i and u_j) are ring bonds in rings of seven or fewer members, and (4) either v_i or v_j (respectively, u_i or u_j) is an unfused bond.

If the preceding requirements are met between two arbitrary vertices (u_i,v_i) and (u_j,v_j) in the modular product graph, then the weak ring heuristic states that both edges v_i and v_j (respectively, u_i and u_j) in graph G_1 (respectively, G_2) must also be in the *same* ring with seven or fewer members. If this is not the case, then there is no edge between the vertices (u_i,v_i) and (u_j,v_j) in the modular product graph. This procedure is illustrated in Figure 10. The pairs of vertices (1,1′), (2,2′) and (1,1′), (2,3′) would have corresponding edges in the modular product graph, whereas the pair of vertices (1,1′), (2,4′) would not have a corresponding edge. Vertices (1,1′) and (2,4′) fulfill requirements 1−4 above when $u_i = 1$, $u_j = 2$, $v_i = 1'$, and $v_j = 4'$. Since v_i and v_j are not both in the same ring with seven or less members, there is no corresponding edge in the modular product graph.

It should be noted that this heuristic can occasionally result in MOS calculations that differ from the pure graph theoretic MOS. Figure 11 points out this minor discrepancy. The MOS that would be calculated using the proposed ring heuristic is highlighted in bold in graphs G_1 and G_2. However, a true

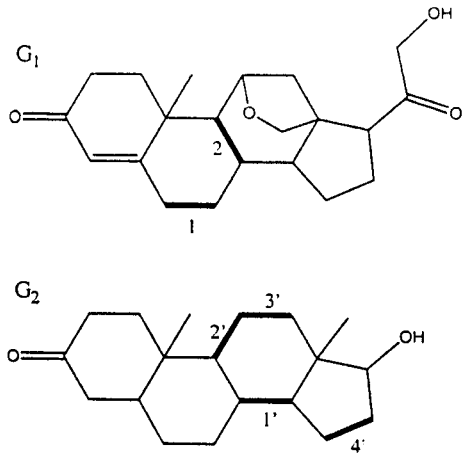

Figure 10. Weak ring heuristic.

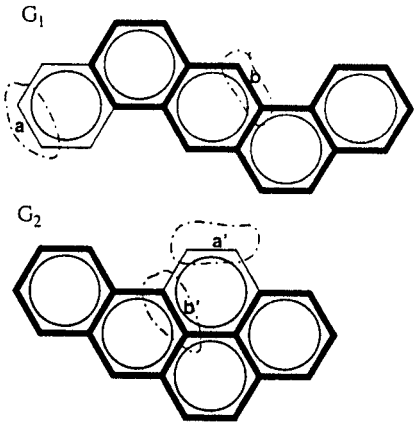

Figure 11. Weak ring heuristic exception.

graph theoretic MOS would also have edge a in G_1 mapped to edge a' in G_2. The reason for this discrepancy can be illustrated by the vertices (a,a′) and (b,b′) in the modular product. Since a' and b' are both in the same ring in G_2, and a and b are not, there is no edge between (a,a′) and (b,b′) in the modular product.

Throughout the course of many trial comparisons using widely varying chemical structures, this "missing edge" drawback occurred very infrequently. When it did occur, the missing edge had negligible effect on the MOS between the two molecules being compared. As such, we believe that this minor limitation is far outweighed by the significant improvements in run time (as discussed further below).

To implement the proposed ring heuristics, it is necessary to enumerate rings with seven or fewer members. There is no theoretical basis for the size limit of seven atoms, and this number can be increased or decreased as necessary. It was chosen on the assumption that a pair of bonds contained solely within a ring with seven or fewer members is significantly more spatially constrained relative to a pair of bonds not contained in a ring with seven or fewer members.

In practice since most molecules have a unique SSSR (smallest set of smallest rings), ring perception can be accomplished using one of the more efficient SSSR algorithms.[29−31] However to be theoretically complete, it is necessary to enumerate all rings of seven or fewer members since the SSSR is not necessarily unique. Algorithms such as the Hanser[32] and Syslo[33] algorithms which enumerate all rings in arbitrary and planar graphs, respectively, can be

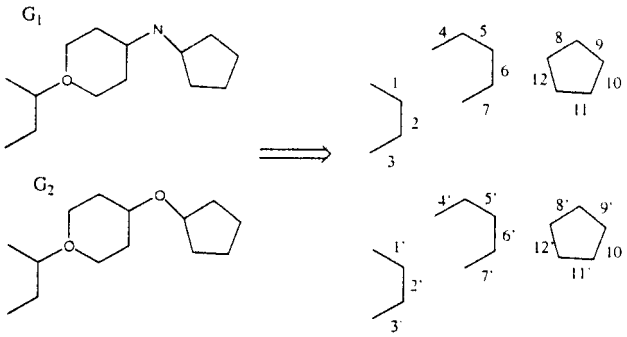

Figure 12. C−C induced subgraphs.

modified to enumerate all rings with seven or fewer atoms. It can also be achieved using a different approach since Alon[34] and Richards[35] have published algorithms capable of finding rings of size k (i.e., C_k) in polynomial time. In the experiments reported below, we used the SSSR algorithm described by Figueras[29] for implementing all ring-based heuristics.

Strong Ring Heuristic. The strong ring heuristic is a slight modification of the weak ring heuristic applied to aromatic rings. If a pair of vertices (u_i,v_i) and (u_j,v_j) in the modular product pass the weak ring heuristic criteria and both rings are aromatic rings, then they must also have a path of the same length in common in both rings. This is assumed regardless of whether the aromatic ring is benzenoid or nonbenzenoid.

C−C Induced Subgraph Heuristic. This edge deletion heuristic is based on the observation that most organic molecules contain moderately large substructures comprised solely of carbon single bonded to carbon (C−C) pairs.

The proposed C−C heuristic can be explained in a stepwise fashion. First, isolate all C−C bonds in a given molecule revealing all connected C−C edge induced subgraphs in the molecular graph. The C−C heuristic operates on this set of induced C−C subgraphs. In this heuristic, any pair of edges, both of which are in a single C−C induced subgraph of molecular graph G_1 that is matched to a pair of edges both of which are in a single C−C induced subgraph in a molecular graph G_2, must be connected by a path of the same length in their respective C−C induced subgraphs in G_1 and G_2.

Figure 12 demonstrates the isolation of the C−C edge induced subgraphs in a pair of graphs G_1 and G_2. Two valid vertices in the modular product are (1,4′) and (3,7′). Since edge 1 and edge 3 in graph G_1 are not incident and edge 4′ and edge 7′ in graph G_2 are not incident, there would be an edge (1,4′), (3,7′) in the modular product using the formal definition. Using the C−C heuristic, however, there is no edge between these two vertices in the modular product. Since edge 1 and edge 3 in graph G_1 are separated by a C−C path of length 2 and edge 4′ and edge 7′ are separated by a C−C path of length 3 in graph G_2, they cannot be connected in the MOS. Thus, there is no corresponding edge in the modular product. In contrast, there is an edge between vertices (1,4′) and (3,6′) in the modular product.

To implement the C−C heuristic, it is necessary to enumerate all paths between two edges in each C−C induced subgraph. When constructing the modular product, two nodes corresponding to pairs of C−C edges that are members of one induced C−C subgraph in graphs G_1 and G_2, respec-

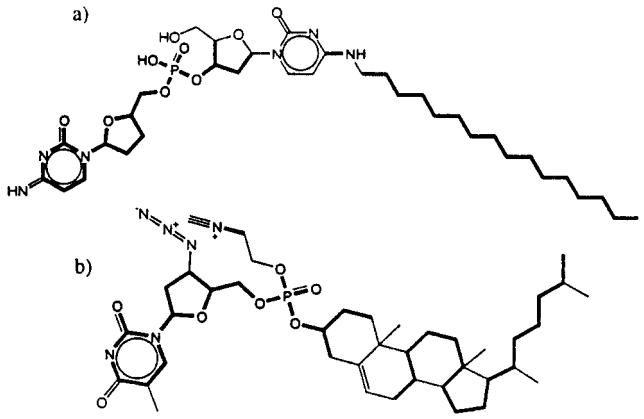

Figure 13. MOS of AIDS actives.

tively, are compatible provided that they both have a C−C path of the same length in common. Since the path enumeration problem is NP-complete, it is necessary that an efficient algorithm be used to enumerate the C−C induced paths. The practical difficulty with this heuristic is that the published algorithms for enumerating paths in graphs are typically very inefficient.[36−40]

Fortunately since molecular graphs are sparse, it is possible to take advantage of the connectivity of molecular graphs when enumerating simple paths. One quite effective yet very simple approach to enumerating these paths involves using the Ullmann subgraph isomorphism algorithm.[41] Since a graph with |V(G)| vertices can have at most a path of length |V(G)| − 1, it is possible to enumerate all paths between pairs of vertices by using the Ullmann algorithm to compare the $P_{|V(G)|-1}$ path graph with the molecular graph of interest. Simply by recording the path at each depth in the depth-first search, all path lengths in the C−C induced subgraph can be enumerated. We have found that this simple method has proven dramatically superior to the Aziz et al. algorithm[38] in direct comparisons by the authors. Interestingly enough, it has been found that this method is algorithmically equivalent to that of Randic et al.[42] Another possible technique for path enumeration involves a path-specific modification of the Hanser ring perception algorithm.[32]

Note that the proposed C−C heuristic does not preclude multiple connected C−C substructures in one molecular graph from being present in another C−C substructure in the other molecular graph. It simply does not allow a single C−C induced subgraph to be disconnected when mapped entirely to another single C−C induced subgraph. Figure 13 demonstrates the application of the proposed heuristic. The two molecules depicted are both active compounds taken from the NCI AIDS database with their respective MOS highlighted in bold. The figure exemplifies the situation where it is necessary to map chain bonds from one chemical graph to ring bonds in another chemical graph to explain adequately the molecular similarity in terms of the MOS.

EXPERIMENTAL EVALUATION

The data set selected for an initial evaluation of the effectiveness of the proposed modular product simplification heuristics consisted of a collection of 200 compounds including steroids, melatonins, statins, antileukemics, and other various compounds from miscellaneous activity classes. The average number of nodes is 25.6 with a standard deviation of 6.7. To investigate the efficiency and robustness of the proposed algorithm, all possible pairwise MOS comparisons between two molecules in a data set were examined. For a data set of size N, this results in $\binom{N}{2}$ comparisons or 19 900 comparisons for this data set. In practice, it may not be necessary to perform all pairwise comparisons to cluster a set of graphs as the MOS-based graph similarity measure used in this paper is a metric obeying the triangle-inequality.[43] Therefore, it may be possible to use information gained from previous comparisons to determine whether it is necessary to perform any given comparison. This remains a question for future research; here, we have computed the complete set of all distinct pairwise similarities as previously stated.

The modular product graphs for each pairwise comparison were constructed with and without using the proposed set of simplification heuristics. The percent reduction of vertices and edge density was calculated using $100 \times (N_{wo} - N_w)/N_{wo}$ where N_{wo} and N_w denote the number without using and the number using the simplification heuristics, respectively. The results indicate an average percent reduction in the number of modular product vertices of 26.2% with a standard deviation of 22.3% and an average reduction in the edge density of 7.8% with a standard deviation of 8.7%. Although these may not appear to be dramatic reductions, we show below that they can yield significant increases in efficiency. This is primarily due to the exponential dependence on the number of vertices and edge density of all known NP-complete algorithms.

The RASCAL algorithm,[11] in addition to determining an MOS between a pair of graphs, returns a measure of the similarity of the two graphs being compared. Although other graph metrics based on the MCS/MOS have been published,[5,6,44,45] RASCAL uses the measure given below[7]

$$sim(G_1,G_2) = \frac{(|V(MOS)| + |E(MOS)|)^2}{(|V(G_1)| + |E(G_1)|)\cdot(|V(G_2)| + |E(G_2)|)}$$

where MOS denotes the maximum common edge subgraph between G_1 and G_2.

RASCAL accepts as input a minimum similarity index (MSI) value (i.e., a minimum acceptable value for $sim(G_1,G_2)$) which is used to improve search efficiency. The specification of an acceptable MSI value will depend in large part on the application as well as the structure of the graphs being considered. It will also be dependent upon the number of vertices and density of the graphs. Figure 14 demonstrates a simple example depicting the $sim(G_1,G_2)$ values calculated for the comparison of ganciclovir with a set of other small nucleoside antiviral agents.

For the comparison of the simplified version to the unsimplified modular product, six separate trial simulations using the RASCAL algorithm were run at MSI values ranging from 0.60 to 0.85 in increments of 0.05. All simulations were run using Visual C++ 6.0 or a 400 MHz Intel Celeron processor with 128 MB RAM under Windows 98. Table 1 presents the average per-comparison time results of implementing the RASCAL algorithm on all 19 900 comparisons with and without using the proposed modular product simplification heuristics. For this comparison, no "time-out" feature was implemented, and the reported times reflect the actual time to allow each comparison to

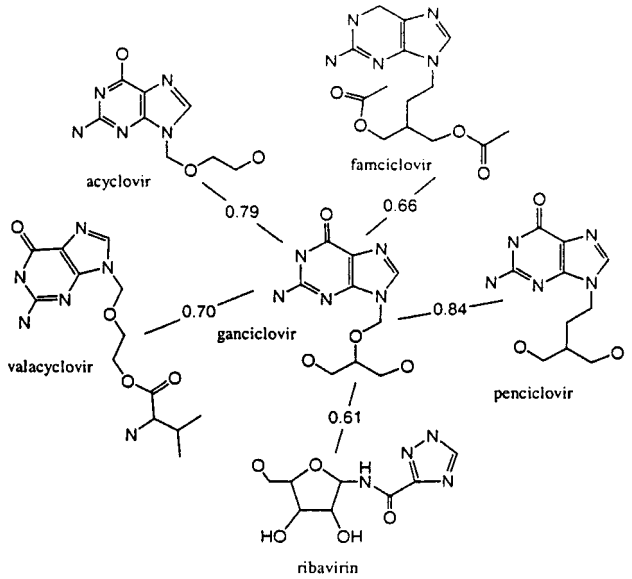

Figure 14. Example of chemical graph similarity.

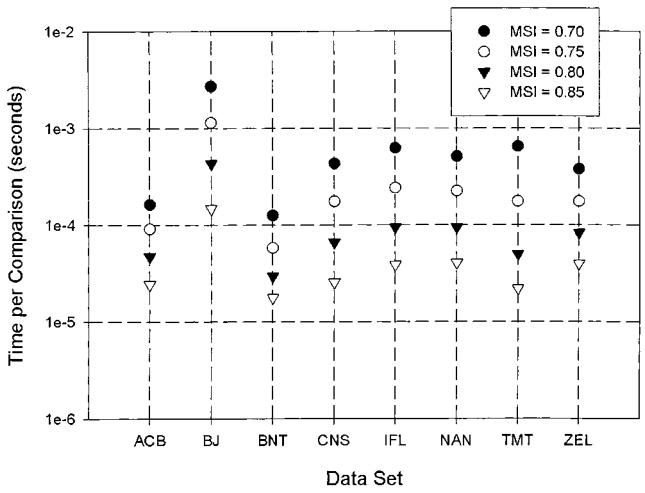

Figure 15. Time per comparison results.

run to completion. It has been assumed that when applying an algorithm to a practical instance of an NP-complete problem, an algorithm is only as good as its worst real-world instance.

An inspection of Table 1 shows that the use of the heuristics proposed in this paper can bring about substantial increases in the efficiency of MOS detection. It will be seen that when close structural relationships (i.e., high MSI values) are sought, even the original RASCAL algorithm is extremely fast. However, the efficiency of the unsimplified algorithm decreases substantially as the MSI threshold is lowered, thus allowing the identification of more distant, but arguably more interesting structural relationships between pairs of molecules. The heuristics described above serve to alleviate the reduction in search efficiency that are observed with the unsimplified algorithm; in the case of the lowest MSI value studied here (0.60), the simplified algorithm is over six times faster than the basic RASCAL procedure.

Given the success of this initial evaluation, an attempt was made to accumulate a wide variety of sets of publicly available drug-like compounds so as to sufficiently test the proposed heuristics combined with the RASCAL algorithm for similarity searching. All of the compounds used in the following simulations are either freely accessible downloadable files from the vendor via the Internet or are available upon request.[46] The relevant statistics regarding the eight selected test sets are listed in Table 2. As with the previous test set, all possible pairwise comparisons were considered with no "time-out" feature. The simulations were run at the MSI threshold values of 0.7, 0.75, 0.8, and 0.85.

The average time per comparison for each test set at each MSI value is presented in Table 2 and Figure 15. Figure 15 reveals that the time per comparison for data sets ACB, BNT, CNS, IFL, NAN, TMT, and ZEL are fairly consistent. The explanation for the larger than average time per comparison for the BJ data set is due not to the size of its respective molecular graph (see Table 1) but rather to the high degree of symmetry characterizing these structures. A high degree of topological symmetry can affect the efficiency of an MOS algorithm by introducing extreme levels of degeneracy into the MOS solution, as illustrated in Figure 16. Figure 16(a) depicts two symmetric molecular graphs with the corresponding MOS highlighted in bold. Figure 16(b) depicts just

Table 1. Comparison of Simplified and Unsimplified Modular Product [Total (per Comparison) in Seconds (Milliseconds)]

	MSI					
source	0.60	0.65	0.70	0.75	0.80	0.85
simplified	120.6 (6.06)	81.0 (4.07)	43.0 (2.16)	22.4 (1.13)	10.9 (0.55)	7.0 (0.35)
unsimplified	735.2 (36.94)	473.1 (23.77)	74.4 (3.74)	31.3 (1.57)	13.4 (0.67)	7.5 (0.38)

Table 2. Chemical Structure Data Set Per-Comparison Time Results[a]

source	no. of structures	no. of comparisons	ID	av no. of atoms	SD	av symmetry (S)	SD	MSI 0.70	MSI 0.75	MSI 0.80	MSI 0.85
ACB blocks	4233	8 973 966	ACB	16.3	4.0	0.038	0.053	24.4 (0.163)	13.7 (0.0914)	7.12 (0.0476)	3.65 (0.0244)
Bakken/Jurs	609	185 136	BJ	26.6	6.5	0.102	0.083	8.35 (2.71)	3.52 (1.14)	1.33 (0.433)	0.45 (0.147)
Bionet	2257	2 545 896	BNT	23.0	5.3	0.057	0.050	5.32 (0.125)	2.45 (0.0578)	1.25 (0.0294)	0.75 (0.0177)
ChemBridge	16 000	127 992 000	CNS	23.3	4.7	0.056	0.049	922 (0.432)	370 (0.174)	141 (0.0661)	54.5 (0.0255)
IFLab	4999	12 492 501	IFL	25.4	6.6	0.071	0.064	130 (0.625)	50.3 (0.241)	19.5 (0.0938)	8.10 (0.0389)
Nanosyn	2715	3 684 255	NAN	26.3	5.5	0.045	0.043	31.4 (0.511)	13.8 (0.224)	5.82 (0.0946)	2.50 (0.0406)
TimTec	5000	12 497 500	TMT	23.2	7.5	0.069	0.074	135 (0.647)	36.5 (0.175)	10.3 (0.0495)	4.57 (0.0219)
Zelinsky Inst.	4999	12 492 501	ZEL	25.2	6.1	0.043	0.051	78.9 (0.379)	36.5 (0.175)	17.3 (0.0828)	8.27 (0.0397)
		180 863 755		23.3	6.7			(0.443)	(0.175)	(0.0675)	(0.0274)

[a] Two figures are listed in each of the MSI columns: the first is the total time in minutes and the second is the time per comparison in milliseconds.

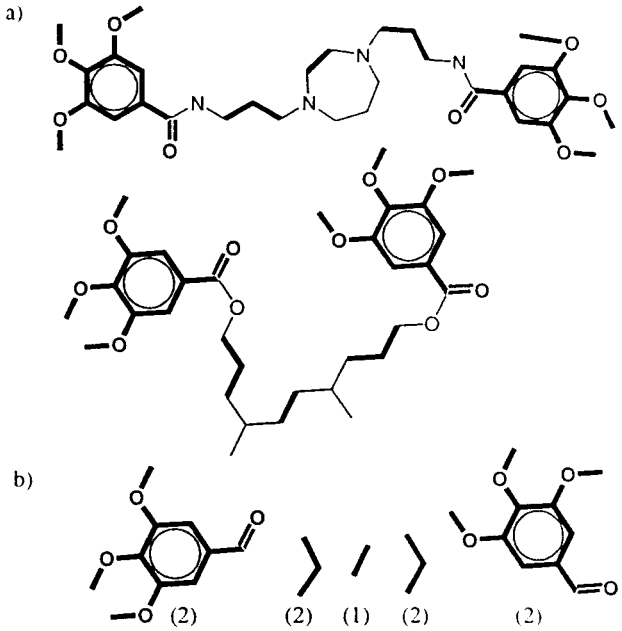

Figure 16. Example of a topologically degenerate MOS.

the MOS between the two molecules. The numbers in parentheses in Figure 16(b) represent the number of ways each distinct subgraph can be mapped onto itself (i.e., the automorphism number). The number of topologically distinct representations of this particular MOS is then $2 \times 2 \times 1 \times 2 \times 2 = 16$. The presence of such degenerate solutions can have a negative impact on the efficiency of MOS detection.

To quantify the relative difference in symmetry of the molecular graphs in each data set, the average symmetry for each data set was calculated using the symmetry measure, $S = -\log(EQ/N)/\log(N)$, where EQ is the number of topologically equivalent edges (i.e., edge equivalence classes) and N is the total number edges in the chemical graph. Since EQ can range from N to 1, it is clear from the definition that S ranges in value from 0 to 1 with 0 being completely unsymmetrical and 1 being maximally symmetric. For instance, S equals 0 for CFClBr, and S equals 1 for CF_3.

The average value of S for each data set is listed in Table 2. From Table 2, it is evident that the BJ data set contains molecules that are more symmetric on average than the other data sets, and the standard deviation of the symmetry measure is larger for the BJ data set indicating that BJ contains more structures with more extreme values of symmetry. This is especially apparent when considering that the symmetry measure was normalized using a log scale.

In this paper, we have only attempted to test the effect of the combination of all of the proposed heuristics, rather than to test each heuristic individually, as the effectiveness of each is highly dependent upon the type of molecular graph being considered. For instance, the edge deletion heuristics would be most effective on molecules such as steroids, whereas the node deletion heuristics would be most effective when applied to structures possessing substituted benzene rings. The objective was therefore to develop a set of simplification heuristics tailored to specific molecular graph types that when used in combination will have a significant impact on the average per-comparison simulation time of any realistic molecular graph.

Table 3. Error Analysis of Proposed Heuristics

data subset	%DC (Total)	%DC (>MSI)	%DS (>MSI)	SD %DS
ACB[a]	0.0079	0.045	5.97	0.0
BJ[a]	0.35	2.8	4.30	0.88
BNT[a]	0.0040	0.36	4.21	0.0
CNS[a]	0.060	3.1	4.85	0.85
IFL[a]	0.32	2.6	4.32	0.93
NAN[a]	0.067	1.5	3.91	0.46
TMT[a]	0.048	1.2	4.87	0.34
ZEL[a]	0.060	0.74	3.47	0.35
mean:	0.11	1.5	4.3	0.9

[a] A subset of 200 molecules with MSI = 0.7.

The results in Table 2 and Figure 15 indicate that the proposed heuristics used in conjunction with the RASCAL algorithm are an effective method for calculating the similarity of chemical graphs and reporting the resultant MOS. It remains, however, to determine the effect that the proposed heuristics have on the resulting MOS solutions. While the proposed node deletion heuristics will not affect the MOS solution, it is theoretically possible for the edge deletion heuristics to result in an MOS solution that is different from the actual graph theoretic solution. To avoid any conflicts associated with discerning whether the difference in MOS solutions can be justified using the notion of chemical sensibility, this analysis will concentrate simply on the effect of this event on the calculated value of $sim(G_1,G_2)$.

For this analysis, a subset of 200 structures arbitrarily selected from each of the data sets presented in Table 2 was investigated, resulting in 25 200 pairwise comparisons per data subset. The results of this investigation are presented in Table 3. The column identifiers for Table 3 are given below:

- %DC (Total): This indicates the percent difference of comparisons which result in a theoretically incorrect value for $sim(G_1,G_2)$ relative to the total number of comparisons. The percent difference is calculated as

$$\%\ Diff = 100 \times \frac{N_{error}}{N_{total}}$$

where N_{error} denotes the number of comparison resulting in a difference in the calculated MOS with and without using the proposed heuristics and N_{total} denotes the total number of pairwise comparisons (i.e., 25 200).

- %DC (>MSI): This statistic is identical to %DC (Total) except that the percent difference is calculated using the number of pairwise comparisons ($N_{>MSI}$) whose value of $sim(G_1,G_2)$ exceeds the specified MSI of 0.7 rather than the total number of pairwise comparisons (N_{total}).
- %DS (>MSI): This number represents the average percent difference in the value of $sim(G_1,G_2)$ in the instances where a discrepancy occurs for each data subset. For instance, the value of 4.30 for the BJ subset indicates that the average difference between $sim(G_1,G_2)$ calculated without using the proposed heuristics and the value of $sim(G_1,G_2)$ calculated using the heuristics is 4.30% for those 2.8% of relevant cases.
- SD %DS: This value represents the standard deviation in %DS (>MSI).

Table 3 shows that, on average, only 0.11% of all comparisons result in a theoretical discrepancy in the calculated MOS using the proposed heuristics, and only 1.5%

of comparisons that exceed the specified MSI threshold of 0.7 result in an inconsistency. Of those few comparisons that do result in a discrepancy, the average percent difference between the two calculated values of $sim(G_1,G_2)$ is 4.3% with a standard deviation of 0.9%. This shows that in addition to reducing the average time per-comparison, the proposed heuristics have a negligible effect on the resulting similarity calculation.

CONCLUSION

Current systems for similarity searching in chemical databases normally employ fingerprint-based measures of structural similarity. Although highly efficient in operation, such measures are limited in that the fragments encoded in a fingerprint describe only local substructural information and thus can provide only an approximate estimate of global similarity; in addition, such measures provide biased estimates of similarity arising from the natures of the similarity coefficients that are used.[47,48] Similarity measures based on the detection of common subgraphs do not suffer from such limitations but have attracted much less attention owing to the NP-complete nature of the available algorithms. We have recently described an algorithm for the detection of maximum common edge subgraphs (or maximum overlapping sets), called RASCAL, that appears to be substantially more efficient than other algorithms available for this purpose. In this paper, we have described several heuristics that draw upon the specific characteristics of chemical graphs and have shown that the inclusion of these heuristics in the algorithm results in increases in the efficiency of chemical graph matching. These increases are particularly noticeable when there is a need to explore remote, nonobvious structural relationships, such as are required in the initial stages of lead-discovery programs.

The algorithmic developments reported here and in our previous paper mean that it is now possible to consider the large-scale implementation of chemical similarity searching based on inexact graph-matching, rather than on the fingerprints that form the structural representations underlying current similarity searching systems. We are hence now conducting a detailed comparison of the effectiveness of retrieval provided by these two methods for similarity searching. The results of this work will be reported shortly; other possible applications include the implementation of graph-based cluster analysis and the extension of our techniques to 3D chemical graphs.

ACKNOWLEDGMENT

The authors would like to extend their gratitude to Pfizer (Ann Arbor) for funding of this project and thank Christine Humblet, Alain Calvet, David Wild, and Eric Gifford of Pfizer and Mark Johnson of Pannanugget Consulting for their support. The Krebs Institute for Biomolecular Research is a designated center of the Biotechnology and Biological Sciences Research Council. We would also like to thank the reviewers for their useful comments and suggestions.

Note Added after ASAP: Erroneous slants were deleted from equations in the Modular Product Graphs section. The corrected version was released to ASAP on February 18, 2002. The print and final web version are correct.

REFERENCES AND NOTES

(1) Johnson, M.; Maggiora, G. *Concepts and Applications of Molecular Similarity*; 1990.
(2) Willett, P.; Barnard, J.; Downs, G. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(3) Dean, P. M. *Molecular Similarity in Drug Design*; Chapman and Hall: 1994.
(4) Sanfeliu, A.; Fu, K. S. A Distance Measure between Attributed Relational Graphs for Pattern Recognition. *IEEE Trans. Syst. Man, Cybern.* **1983**, *13*, 353−362.
(5) Bunke, H.; Shearer, K. A Graph Distance Metric Based on the Maximal Common Subgraph. *Pattern Recogn. Lett.* **1998**, *19*, 255−259.
(6) Balaz, V.; Jaroslav, K.; Kvasnicka, V.; Milan, S. A Metric for Graphs. *Casopis Pest. Mater.* **1986**, *111*, 431−433, 435.
(7) Johnson, M.; Naim, M.; Nicholson, V.; Tsai, C. Unique Mathematical Features of the Substructure Metric Approach to Quantitative Molecular Similarity Analysis. In *Graph Theory and Topology in Chemistry*; King, R. B., Rouvray, D. H., Eds.; Elsevier Science Publishers: 1987; pp 219−225.
(8) Bersohn, M. An Algorithm for Finding the Intersection of Molecular Structures. *J. Chem. Soc. Perk. Trans. 1* **1982**, *1*, 631−637.
(9) Cone, M.; Venkataraghavan, R.; McLafferty, F. Molecular Structure Comparison Program for the Identification of Maximal Common Substructures. *J. Am. Chem. Soc.* **1977**, *99*, 7668−7671.
(10) Takahashi, Y.; Satoh, Y.; Sasaki, S. Recognition of Largest Common Structural Fragment Among a Variety of Chemical Structures. *Anal. Sci.* **1987**, *3*, 23−28.
(11) Raymond, J.; Gardiner, E.; Willett, P. RASCAL: Calculation of Graph Similarity Using Maximum Common Edge Subgraphs. *Comput. J.* submitted for publication.
(12) Kann, V. *On the Approximability of NP−Complete Optimization Problems*; Ph.D. Thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology: Stockholm, Sweden, 1992.
(13) Brown, R. D. *A Hyperstructure Model for Chemical Structure Handling*; Ph.D. Thesis, Department of Information Studies, University of Sheffield: Sheffield, U.K., 1993.
(14) Levi, G. A Note on the Derivation of Maximal Common Subgraphs of Two Directed or Undirected Graphs. *Calcolo* **1972**, *9*, 341−352.
(15) Barrow, H.; Burstall, R. Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques. *Inf. Proc. Lett.* **1976**, *4*, 83−84.
(16) Vizing, V. G. Reduction of the Problem of Isomorphism and Isomorphic Entrance to the Task of Finding the Nondensity of a Graph (in Russian). In *Third All-Union Conference on Problems of Theoretical Cybernetics*; Novosibirsk, 1974; p 124.
(17) Pelillo, M.; Siddiqi, K.; Zucker, S. W. Matching Hierarchical Structures Using Association Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 1105−1120.
(18) Bessonov, Y. E. On the Solution of a Problem on the Search for the Best Intersection of Graphs on the Basis of an Analysis of the Projections of the Subgraphs of the Modular Product (in Russian). *Vychisl. Sistemy* **1985**, *3*−*22*, 121.
(19) Skorobogatov, V. A. Determination of Common Parts in Families of Graphs (in Russian). In *Applied Problems on Graphs and Networks*; Akad. Nauk SSSR Sibirsk, Otdel.: Vychisl. Tsentr, 1981; pp 117−132.
(20) Bessonov, Y. E. Determination of the Intersections of Geometric Graphs by Means of the Operation of the Modular Product (in Russian). *Vychisl. Sistemy* **1987**, *43*−*48*, 114.
(21) Adamson, G.; Lambourne, D.; Lynch, M. Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-based File. Part III. Statistical Association of Fragment Incidence. *J. Chem. Soc., Perkin Trans. 1* **1972**, 2428−2433.
(22) Whitney, H. Congruent Graphs and the Connectivity of Graphs. *Am. J. Math.* **1932**, *54*, 150−168.
(23) Nicholson, V.; Tsai, C.; Johnson, M.; Naim, M. A Subgraph Isomorphism Theorem for Molecular Graphs. In *Graph Theory and Topology in Chemistry*; King, R. B., Rouvray, D. H., Eds.; Elsevier: 1987; pp 226−230.
(24) Kvasnicka, V.; Pospichal, J. Maximal Common Subgraphs of Molecular Graphs. *Reports in Molecular Theory* **1990**, *1*, 99−106.
(25) Durand, P. *An Improved Program for Topological Similarity Analysis of Molecules*; M.S. Thesis, Department of Mathematics and Computer Science, Kent State: Toledo, OH, 1996.
(26) Durand, P.; Pasari, R.; Baker, J.; Tsai, C. An Efficient Algorithm for Similarity Analysis of Molecules. *Internet J. Chem.* **1999**, *2*, 1−12.
(27) Chen, C. K.; Yun, D. Y. Unifying Graph Matching Problems with a Practical Solution. In *International Conference on Systems, Signals, Control, Computer*; Durban, South Africa, 1998.
(28) Brown, R. D.; Jones, G.; Willett, P.; Glen, R. Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 63−70.

(29) Figueras, J. Ring Perception Using Breadth-First Search. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 986−991.
(30) Takahashi, Y. Automatic Extraction of Ring Substructures from a Chemical Structure. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 167−170.
(31) Chaudhuri, P. A Self-Stabilizing Algorithm for Detecting Fundamental Cycles in a Graph. *J. Comput. System Sci.* **1999**, *59*, 84−93.
(32) Hanser, T.; Jauffret, P.; Kaufman, G. A New Algorithm for Exhaustive Ring Perception in a Molecular Graph. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1146−1152.
(33) Syslo, M. An Efficient Cycle Vector Space Algorithm for Listing All Cycles of a Planar Graph. *SIAM J. Comput.* **1981**, *10*, 797−808.
(34) Alon, N.; Yuster, R.; Zwick, U. Finding and Counting Given Length Cycles. *Algorithmica* **1997**, *17*, 209−223.
(35) Richards, D. Finding Cycles of a Given Length. In *Cycles in Graphs*; Burnaby, B. C., Ed.; North-Holland, 1982; pp 249−255.
(36) Kahn, A.; Singh, H. Petri Net Approach to Enumerate All Simple Paths in a Graph. *Electron. Lett.* **1980**, *16*, 291−292.
(37) Misra, R. An Algorithm for Enumerating All Simple Paths in a Communication Network. *Microelectron. Reliab.* **1979**, *19*, 363−366.
(38) Aziz, M.; Sobham, M.; Samad, M. Fast Enumeration of Every Path in a Reliability Graph Using Subgraphs. *Microelectron. Reliab.* **1994**, *34*, 1395−1396.
(39) Aihara, K. Approach to Enumerating Elementary Paths and Cutsets by Gaussian Elimination Method. *Electron. Commun. Jpn.* **1975**, *58*, 1−10.
(40) Cahit, I.; Cahit, R. Generation of Simple Paths of Graph by Decomposition. *Electron. Lett.* **1974**, *10*, 13−14.
(41) Ullmann, J. R. An Algorithm for Subgraph Isomorphism. *J. ACM* **1976**, *23*, 31−42.
(42) Randic, M.; Brissey, G. M.; Spencer, R. B.; Wilkins, C. L. Search for All Self-Avoiding Paths for Molecular Graphs. *Comput. Chem.* **1979**, *3*, 5−13.
(43) Johnson, M. A Review and Examination of the Mathematical Spaces Underlying Molecular Similarity Analysis. *J. Math. Chem.* **1989**, *3*, 117−145.
(44) Fernandez, M. L.; Valiente, G. *A Graph Distance Metric Combining Maximum Common Subgraph and Minimum Common Supergraph*; Research Report, LSI-01-2-R; Universitat Politecnica De Catalunya, 2001.
(45) Wallis, W. D.; Shoubridge, P.; Kraetz, M.; Ray, D. Graph Distances Using Graph Union. *Pattern Recogn. Lett.* **2001**, *22*, 701−704.
(46) ACB Blocks Ltd.; Moscow, Russia; (www.acbblock.com). Bakken, G. A.; Jurs, P. C. Classification of Multidrug-Resistance Reversal Agents Using Structure-Based Descriptors and Linear Discriminant Analysis. *J. Med. Chem.* **2000**, *43*, 4534−4541. Bionet Research; Camelford, U.K..; (www.bionetresearch.co.uk). ChemBridge; San Diego, CA, U.S.A.; (www.chembridge.com). IFLab; Kiev, Russia; (www.iflab.kiev.ua). Nanosyn; Mountain View, CA, U.S.A.; (www.nanosyn.com). TimTec; Newark, DE, U.S.A.; (www.timtec.net). Zelinsky Institute; Moscow, Russia; (www.zelinsky.com).
(47) Flower, D. R. On the Properties of Bit String-Based Measures of Chemical Similarity. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 379−386.
(48) Godden, J.; Xue, L.; Bajorath, J. Combinatorial Preferences Affect Molecular Similarity/Diversity Calculations Using Binary Fingerprints and Tanimoto Coefficients. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 163−166.

CI010381F