ELSEVIER

Clique-detection algorithms for matching three-dimensional molecular structures

Eleanor J. Gardiner,*†‡ Peter J. Artymiuk,*‡ and Peter Willett*†

**Krebs Institute for Biomolecular Research, †Department of Information Studies, ‡Department of Molecular Biology and Biotechnology, University of Sheffield, Western Bank, Sheffield, UK.*

The representation of chemical and biological molecules by means of graphs permits the use of a maximum common subgraph (MCS) isomorphism algorithm to identify the structural relationships existing between pairs of such molecular graphs. Clique detection provides an efficient way of implementing MCS detection, and this article reports a comparison of several different clique-detection algorithms when used for this purpose. Experiments with both small molecules and proteins demonstrate that the most efficient of these particular applications, which typically involve correspondence graphs with low edge densities, is the algorithm described by Carraghan and Pardalos. This is shown to be two to three times faster than the Bron–Kerbosch algorithm that has been used previously for MCS applications in chemistry and biology. However, the latter algorithm enables all substructures common to a pair of molecules to be identified, and not just the largest ones, as with the other algorithms considered here. The two algorithms can usefully be combined to increase the efficiency of database-searching systems that use the MCS as a measure of structural similarity. © 1998 by Elsevier Science Inc.

Keywords: Bron-Kerbosch algorithm, Carraghan-Pardalos algorithm, clique detection, graph matching, maximal common substructure, pharmacophore mapping

INTRODUCTION

Modern pharmaceutical and agrochemical research makes extensive use of database systems for the storage and retrieval of three-dimensional (3D) structural information.[1–3] These systems are based on the isomorphism techniques that have been developed for establishing the structural relationships that exist between pairs of graphs.[4–6] In this article, we report an evaluation of the efficiencies of several different algorithms for the identification of maximum common subgraph isomorphisms in 3D databases. The next section introduces the necessary graph-theoretic definitions, and this is followed by brief descriptions of the various algorithms that we have considered in the evaluation. We then report the results of an extended series of experiments using both chemical and biological 3D structures, and the article concludes with a summary of our principal findings.

Address reprint requests to: Prof. Willett, Department of Information Studies, University of Sheffield, Western Bank, S10 2TN, Sheffield, UK. E-mail: P.WILLETT@SHEFFIELD.AC.UK.
Received 30 September 1997; accepted 30 October 1997.

IDENTIFICATION OF MAXIMUM COMMON SUBGRAPHS

A graph, G, consists of a set of vertices, together with a set of edges connecting pairs of vertices, and two graphs, G_1 and G_2, are isomorphic if there is an exact correspondence between the vertices of G_1 and of G_2 such that adjacent pairs of vertices in G_1 are mapped to adjacent pairs of vertices in G_2. A *subgraph* of G is a subset, P, of the vertices of G together with a subset of the edges connecting pairs of vertices in P, and subgraph isomorphism exists if G_1 is a subgraph of G_2 (or vice versa). A common subgraph of two graphs G_1 and G_2 is defined as consisting of a subgraph g_1 of G_1 and a subgraph g_2 of G_2 such that g_1 is isomorphic to g_2; the maximum common subgraph (or MCS) is the largest such common subgraph.

Algorithms for the identification of the MCS between pairs of 3D chemical molecules, where the MCS is defined to be the largest set of atoms that have matching interatomic distances (to within user-defined tolerance values) in the two molecules that are being compared,[7] were first considered in the context of automated methods for the identification of pharmacophoric patterns. Crandell and Smith[8] noted that if one had two, structurally disparate molecules that both exhibited a biological activity of interest then an initial specification of the pharmacophoric pattern involved could be obtained from the MCS of the graphs representing these two molecules. Later, Brint and Willett[9] showed that such pharmacophores could be identified relatively efficiently using MCS algorithms that are based on clique detection.[10–13]

655 Avenue of the Americas, New York, NY 10010
1093-3263/97/$17.00
PII S1093-3263(97)00089-2

A clique is a subgraph of a graph in which every node is connected to every other node and that is not contained in any larger subgraph with this property, and the MCS algorithm operates by identifying the cliques in a *correspondence graph*. Given a pair of graphs, G_1 and G_2, the correspondence graph, C, can be formed by creating the set of all pairs of vertices, one from each of the two graphs, such that the vertices of each pair are of the same type: C is then the graph whose vertices are these pairs of vertices. Two correspondence graph vertices $\{G_1(I),G_2(X)\}$ and $\{G_1(J),G_2(Y)\}$ are connected in C if the values of the edges from $G_1(I)$ to $G_1(J)$ and $G_2(X)$ to $G_2(Y)$ are the same. Barrow and Burstall[11] showed that the MCSs of G_1 and G_2 then correspond to the cliques of C, so that the identification of the largest possible pharmacophore for a pair of 3D small molecules is equivalent to the identification of the maximum clique in the correspondence graph linking their two structures. Brint and Willett[9] compared the efficiencies of five clique-detection algorithms for 3D MCS detection, and found that the algorithm of Bron and Kerbosch[14] was by far the fastest for automated pharmacophore detection. A worked example of the operation of the Bron–Kerbosch algorithm is provided by Willett,[7] and it forms the basis for the DISCO (DIStance COmparisons) package for automatic pharmacophore detection that has been developed at Abbott Laboratories.[15]

The MCS between two molecules provides an obvious measure of structural similarity. This idea forms the basis for 3D similarity searching systems, with the molecules in a database being ranked in decreasing order of structural overlap with the user's target structure,[16–18] and for our work on the use of MCS algorithms for identifying structural equivalences in 3D protein structures.[19,20] Here, the α-helix and β-strand secondary structure elements (hereafter SSEs) of a protein structure form the vertices of a graph, with the edges being inter-SSE angles and distances.[21] The Protein Data Bank (PDB)[22] can hence be represented by a set of labeled graphs that is searched using the program PROTEP,[20] which contains an implementation of the Bron–Kerbosch algorithm that retrieves all of the proteins in the PDB containing at least some minimum number of SSEs in the same geometric arrangement as in a user-defined target structure. This has proved to be an effective way of uncovering previously unknown structural relationships between proteins in the PDB that have little or no sequence homology, such as the resemblances we have discovered between the ribonuclease H and connection domains of human immunodeficiency virus (HIV) reverse transcriptase,[23] between leucine aminopeptidase and carboxypeptidase A,[24] and, most recently, between adenyl cyclase and the palm domain of DNA polymerase I.[25]

CLIQUE-DETECTION ALGORITHMS

Criteria for selection

An extensive literature search was carried out to identify novel clique-detection algorithms that had been reported since the earlier study by Brint and Willett.[9] The algorithms that were found fall into several, not necessarily disjoint, categories: they may be classified by the basic computational approach that is used, e.g., branch-and-bound or neural network; they may be serial or require some form of parallel (and often massively parallel) computer; they may be exact or approximate in nature, the latter generally being far faster in execution; they may be enumerative, listing all of the cliques in the correspondence graph, focus just on the maximal clique(s), i.e., those that are not contained in another, larger clique, or on the maximum clique(s), i.e., the largest possible maximal clique(s); and they may be restricted to some specific class of graph, such as interval graphs or permutation graphs.

As a result of the literature review, five algorithms were chosen for comparison with the Bron–Kerbosch algorithm. These were those described by Babel,[26] by Balas and Yu,[27] by Carraghan and Pardalos,[28] by Gendreau et al.,[29] and by Shindo and Tomita.[30] In the following, these will be abbreviated where appropriate to B, BY, CP, G, and ST, respectively, with the Bron–Kerbosch algorithm being referred to as BK.

All of the algorithms that we have tested are exact, require only a conventional, serial processor for their execution, and are quite simple in concept and thus easy to implement. In addition, none are stated as having been designed specifically for processing highly complex correspondence graphs. The complexity of a graph is generally described by its *edge density*,[26] which represents the probability that two vertices in the correspondence graph are connected. The density of a correspondence graph containing e vertices and k edges is given by Eq. (1):

$$\frac{2k}{e(e-1)} \tag{1}$$

A study of the correspondence graphs resulting from 3D molecules[31] shows that the former have very low edge densities when compared with those that are typically considered in the mainstream graph theoretic, computer science, and operations research literatures, and this finding hence informed our choice of algorithms for the evaluation reported here.

A final point is that the proposed database applications require algorithms that are efficient in operation, and we thus took account of any reported comparisons when choosing which algorithms should be included in our comparison. Thus, because BK provides our baseline of performance, it was natural to choose BY as Balas and Yu[27] report their algorithm as being faster than BK; similarly, both Babel[26] and Gendreau et al.[29] describe their algorithms as being faster than BY. CP is described as being fast by Lin[32] and was also selected as a benchmark for the 1993 DIMACS algorithm challenge.[13] Finally, ST was reported to be "constructed by adopting the technique of Bron and Kerbosch"[30] and is very simple in design, while still having a theoretical worst-case time complexity that is only slightly worse than the best known to date (the algorithm described by Tarjan and Trojanowski[33]).

The algorithms

We now give brief descriptions of the selected algorithms: the reader is referred to the original papers for full details.

Babel's algorithm for finding a maximum clique in an arbitrary graph is a partially enumerative, branch-and-bound algorithm. The bounds used to prune the tree are the size of the current maximum clique (lower bound) and the size of a minimum coloring (upper bound). Informally, colors (represented by positive integers) are assigned to vertices such that adjacent vertices are assigned different colors. It is fairly simple to derive a near-minimal value for the number of colors needed to color a graph, and this information is used in the calculation of the upper bound.

The Balas–Yu algorithm is a depth-first, branch-and-bound procedure in which the branching mechanism has since been adopted by other workers.[34,35] The basic idea of the algorithm is to use a class of subgraphs known as *triangulated subgraphs* that are known to be *perfect*, where a perfect graph is one for which a maximum clique can be found in polynomial time,[36,37] rather than the nonpolynomial time requirements that characterize general graphs. It is thus very simple to find a maximum clique for the triangulated subgraphs, and this clique then gives a lower bound on the size of the maximum clique for the correspondence graph. The algorithm is unusual in that no attempt is made to discover an upper bound for the maximum clique size, but it has still been found to be highly efficient in operation.

The Carraghan–Pardalos algorithm is a very simple, partially enumerative procedure. Assume that the vertices in the correspondence graph are labeled $1 \ldots n$. The neighbors of a vertex, i, that have higher labels than i are referred to as its *successors*, and a vertex is *expanded* by taking its successors and putting them in a list at depth 2. The algorithm starts by expanding the first vertex, v_1, and finding a largest clique in the correspondence graph, C, that contains v_1. Then v_2 is expanded and the algorithm finds a largest clique in $C\backslash\{1\}$ that contains v_2, where $C\backslash\{1\}$ denotes the graph C after removal of vertex v_1. Then v_3 is expanded and the algorithm finds a largest clique in $C\backslash\{1,2\}$, and so on. A backtracking search procedure is used with an upper bound condition to prune the search tree. This extremely simple condition states that the algorithm should backtrack when processing vertex v_k if that vertex has insufficient successors to form a clique larger than the best already found.

The algorithm described by Gendreau et al. is a depth-first, branch-and-bound, partially enumerative procedure. The algorithm is based on a fairly straightforward greedy heuristic, which selects at each step the vertex of maximum degree among the remaining candidates to extend the current complete subgraph. Gendreau et al. note that this heuristic can produce very bad solutions in some cases, and they accordingly define the *triangle degree* of a vertex. This is the number of triangles to which a vertex belongs, and the algorithm seeks to choose vertices of maximum triangle degree, with the pruning of vertices for which the triangle degree is too small.

Finally, the Shindo–Tomita algorithm is a simple, partially enumerative algorithm to find a single maximum clique. The cited article[30] is principally concerned with the worst-case time complexity, $O(2^{n/2.863})$, of the algorithm and only passing mention is made of some pruning techniques that had been described previously[38]; however, the algorithm in Ref. 30 is very slow in practice unless these techniques are included. It is interesting to note that one of them, the bound condition, is actually the same as that used by CP and given above; indeed, the basic version of ST is the same as CP if no pruning is carried out. The main question to be answered is thus whether their modifications, which improve the theoretical worst-case performance, are efficient in the context of the correspondence graphs studied here. The first modification, which Shindo and Tomita call "identification-of-partial-forest rule," is based on the observation that if a vertex, v_k, belongs to a clique then none of its neighbors can belong to a larger clique unless the latter contains a nonneighbor of v_k. This observation permits an ordering of the vertices, and a consequent prioritization of the vertices as the search tree is explored. The second modification, which Shindo and Tomita call "identification restriction," is used to decide whether or not to reapply the identification-of-partial-forest rule at a lower depth in the tree.

EXPERIMENTAL DETAILS AND RESULTS

The algorithms described above, modified where necessary to find all maximum cliques rather than just a single such clique, were coded in C and run on a Silicon Graphics Indigo workstation with an R4000 processor and 48MB of memory (with the exception of the results shown in Table 5, which were obtained with a DEC Alphastation 600). Experiments were carried out involving correspondence graphs linking pairs of small molecules and correspondence graphs linking pairs of proteins. Each run was repeated 50 times and the run times quoted in Tables 1–5 are the mean values when averaged over the set of runs. A run was terminated if a maximum clique had not been identified within 100 s of CPU time.

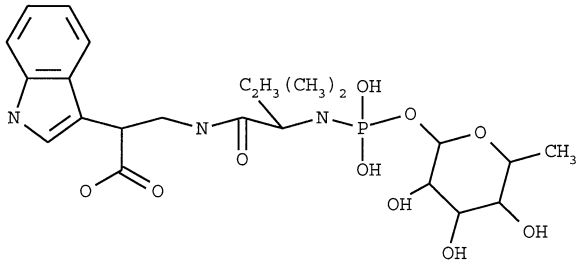
Small-molecule structures

The small-molecule test data consisted of the three pairs of molecules shown in Figure 1: **I** and **II** are angiotensin-converting enzyme (ACE) inhibitors[39] containing 37 and 33 nonhydrogen atoms, respectively; **III** and **IV** are cholecystokinin A (CCK-A) antagonists,[40] both containing 30 nonhydrogens; and **V** and **VI** are anticoccidial triazines,[41] both containing 26 nonhydrogens. The molecules were represented by CONCORD structures and these were used to generate the correspondence graph in each case. Vertices in this graph are joined by an edge if the interatomic distances between the pairs of atoms are the same to within a user-defined tolerance.

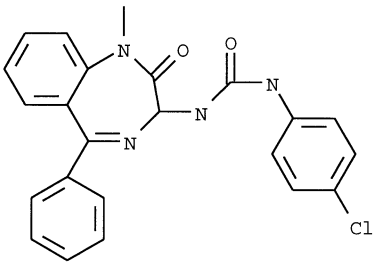
Ten correspondence graphs were created for each pair of molecules by varying the tolerance from 1.0 to 0.1 Å. This has the effect of progressively deleting edges from the correspondence graph, but tends to preserve the maximum clique until the tolerance becomes very low. For example, in the case of **III** and **IV** the maximum clique size does not change from 29 until the tolerance reaches 0.4 Å. A further set of eight correspondence graphs was created for each pair of molecules by distorting the first molecule while keeping the tolerance fixed at 0.5 Å. This was done by multiplying the x coordinate of the molecule by a translation factor of between 1.5 and 5.0, thus "stretching" the molecule along the x axis. The similarity between a pair of molecules (as measured by the size of a maximum clique) is altered more easily using this method than when the tolerance is varied.

The first set of runs used **I** and **II** with the tolerance varied from 1.0 to 0.1 Å and with several different versions of each of the algorithms; in all, these initial experiments involved a total of 17 versions of the 6 algorithms (BK, B, CP, G, BY, and ST; BK finds all maximal cliques, so a modified version was tested here that found just the maximum cliques and that was thus comparable to the other algorithms). The results led us to exclude the algorithm of Gendreau et al. from further consideration, because all runs with tolerances greater than 0.5 Å were aborted after running for 100 s of CPU time, and to identify the most efficient version of each of the other algorithms for additional testing.

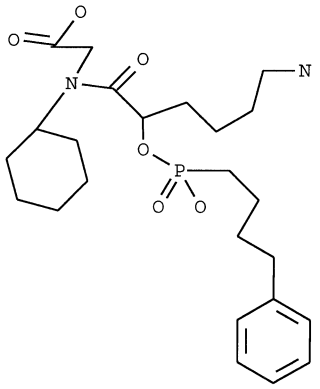
Table 1 details the results obtained when the best versions



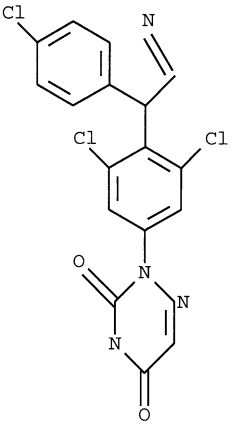
I



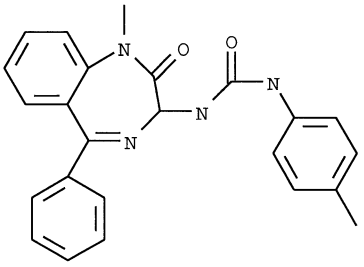
IV



II



V



III

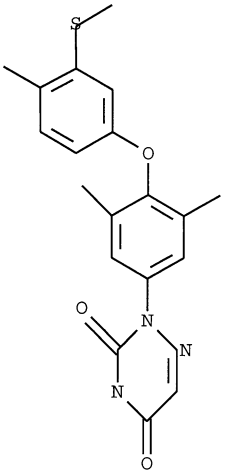


Figure 1. Small-molecule test data: **I** *and* **II**, *ACE inhibitors;* **III** *and* **IV**, *CCK-A antagonists;* **V** *and* **VI**, *anticoccidial triazines.*

were applied to all three pairs of molecules and when the tolerance was varied systematically. In Table 1A–C, T is the tolerance in angstroms, ρ is the edge density of the graph (as defined previously), ω is the size of the maximum clique, and M is the number of such cliques present in the correspondence graph. Table 2 contains the corresponding results when one of the molecules in each pair was translated by the factor, X. An inspection of the entries in Tables 1 and 2 shows clearly that CP is the only algorithm that is able to outperform BK. ST seems comparable to BK but is certainly not as fast as CP.

Table 1. Mean run times for matching molecules I and II, III and IV, and V and VI, when the distance tolerance *T* is varied

T	ρ	ω	*M*	B	BK	CP	BY	ST
				A. Molecules I and II				
1.0	0.192	15	3	42.9	12.7	7.0	13.5	14.5
0.9	0.171	15	1	24.9	7.7	3.9	15.1	7.9
0.8	0.153	14	1	21.3	5.4	2.2	11.0	5.1
0.7	0.136	13	2	12.7	3.9	1.5	9.0	3.6
0.6	0.118	12	2	5.6	2.8	1.0	7.2	2.4
0.5	0.102	11	4	3.2	2.0	0.7	5.2	1.6
0.4	0.084	11	2	1.6	1.4	0.5	5.7	1.0
0.3	0.065	10	2	0.8	1.0	0.3	4.4	0.7
0.2	0.047	8	3	0.6	0.7	0.2	3.6	0.5
0.1	0.025	6	17	0.4	0.5	0.2	2.2	0.4
				B. Molecules III and IV				
1.0	0.221	29	1	0.4	10.6	61.8	1.5	7.1
0.9	0.202	29	1	0.3	6.7	15.4	0.9	4.3
0.8	0.182	29	1	0.3	4.3	4.8	0.8	2.5
0.7	0.165	29	1	0.3	3.1	1.7	0.6	1.7
0.6	0.146	29	1	0.3	2.1	0.9	0.5	1.2
0.5	0.126	29	1	0.3	1.6	0.6	0.5	0.8
0.4	0.107	28	2	0.3	1.2	0.4	0.4	0.6
0.3	0.084	25	6	0.3	0.9	0.4	0.7	0.5
0.2	0.063	23	2	0.3	0.6	0.2	0.5	0.3
0.1	0.038	21	1	0.2	0.4	0.1	0.4	0.2
				C. Molecules V and VI				
1.0	0.189	20	2	0.2	0.9	2.2	1.8	1.1
0.9	0.170	20	2	0.1	0.7	1.6	1.6	0.8
0.8	0.153	20	2	0.3	0.5	1.0	1.7	0.6
0.7	0.139	20	1	0.2	0.4	0.7	0.8	0.4
0.6	0.127	20	1	0.1	0.3	0.3	0.7	0.2
0.5	0.112	20	1	0.1	0.3	0.2	0.8	0.2
0.4	0.097	20	1	0.1	0.2	0.1	0.6	0.2
0.3	0.077	20	1	0.1	0.2	0.1	0.3	0.1
0.2	0.060	18	2	0.2	0.2	0.1	0.3	0.1
0.1	0.038	14	2	0.1	0.1	0.1	0.4	0.1

Protein structures

The programs were also tested on two protein data sets that are known to exhibit substantial 3D resemblances at the level of the arrangement of their SSEs in 3D space. The first pair included the ribonuclease H and connection domains of HIV reverse transcriptase[23] and the second pair consisted of leucine aminopeptidase and carboxypeptidase A.[24] In each case, the 3D structures (PDB codes[22] 1HRH for the two domains in HIV reverse transcriptase, and 5CPA and 1LAP for the two peptidase structures) were extracted from the PDB and then clique detection was carried out using a range of tolerances. Specifically, the distance tolerance was fixed at the minimum of 40% of the distance of closest approach or 4 Å, the angle between the vectors representing a pair of SSEs was varied between 25 and 45%, and the sequence either was, or was not, taken into account when deciding whether two vertices matched.

Initial experiments using the full range of algorithm variants on the two reverse transcriptase domains identified the same subset of versions for further investigation as in the small-molecule experiments described previously. An inspection of the results in Table 3 shows that they are very similar to those for small molecules in that CP consistently outperforms BK by a factor of 2 or 3, ST gives results that are similar to BK, and both B and BY are significantly worse.

Use of the Carraghan–Pardalos algorithm

The results presented in Tables 1–3 suggest that CP is the most efficient algorithm of those tested here, with its relative per-

Table 2. Mean run times for matching molecules I and II, III and IV, and V and VI, when the second molecule is translated by a factor *X*

X	ρ	ω	*M*	B	BK	CP	BY	ST
				A. Molecules I and II				
1.5	0.088	10	4	2.6	1.5	0.5	5.9	1.2
2.0	0.072	8	5	2.0	1.1	0.4	8.0	1.2
2.5	0.061	8	3	1.2	0.9	0.3	4.3	0.9
3.0	0.054	9	1	0.6	0.8	0.3	4.5	0.9
3.5	0.047	8	1	0.6	0.7	0.3	3.3	0.7
4.0	0.042	7	1	0.6	0.6	0.2	3.9	0.6
4.5	0.038	6	1	0.7	0.6	0.2	3.2	0.6
5.0	0.035	6	4	0.6	0.5	0.2	3.3	0.6
				B. Molecules III and IV				
1.5	0.117	13	1	2.6	1.8	0.7	5.7	1.6
2.0	0.109	9	8	3.7	1.7	0.6	3.7	1.6
2.5	0.097	9	5	2.0	1.3	0.5	4.4	1.3
2.5	0.087	9	2	1.9	1.1	0.4	4.2	1.3
3.5	0.076	7	6	1.7	1.0	0.3	4.6	1.1
4.0	0.068	7	1	1.2	0.8	0.3	3.6	1.0
4.5	0.061	6	10	1.2	0.7	0.3	3.5	0.8
5.0	0.054	6	2	1.0	0.6	0.2	3.1	0.7
				C. Molecules V and VI				
1.5	0.101	13	1	0.2	0.3	0.1	0.8	0.3
2.0	0.098	9	2	0.3	0.3	0.1	2.2	0.4
2.5	0.087	8	3	0.4	0.2	0.1	1.3	0.3
2.5	0.080	6	27	0.6	0.2	0.1	2.1	0.4
3.5	0.073	6	5	0.4	0.2	0.1	1.1	0.3
4.0	0.069	6	3	0.4	0.2	0.1	1.5	0.3
4.5	0.065	6	4	0.4	0.2	0.1	1.6	0.3
5.0	0.062	5	45	0.4	0.2	0.1	2.1	0.3

formance being poor only in some of the runs involving the very highest edge densities. One might hence conclude that CP is the algorithm of choice, and could, with benefit, replace BK in systems such as DISCO[15] or PROTEP.[20] Unfortunately, all of the new algorithms considered here, including CP, only find maximum cliques in a correspondence graph, whereas BK normally finds all the maximal cliques (as noted previously, the version of BK used here found just the maximum cliques to allow comparison with the other algorithms). Thus, if one wishes to find *all* structural equivalences between a pair of 3D structures, rather than just the largest equivalences (for example, to find equivalences that are in addition to one, very large, obvious structural commonality, as might occur with a set of analogs in a QSAR investigation), then BK continues to be the algorithm of choice. That said, the speed of CP can be exploited by using it as a screen to ascertain whether it is necessary to invoke the more time-consuming BK in a database search. Specifically, CP can be used to determine the size of the MCS that each database structure has in common with the target structure; only those database structures with sufficiently large MCSs are then submitted to a second match involving BK.

Assume that a user-defined target structure is to be matched against each of the structures in a database, and that the average time required for each such match using the two algorithms is denoted by t_{BK} and t_{CP}. Then a database search involving just BK will be slower than one in which CP is used as an initial screen if

$$t_{BK} \geq t_{CP} + \alpha t_{BK} \qquad (2)$$

where α is the fraction of the file that is found to require the second-stage, BK search when CP is used as a screen.

A large number of experiments were carried out, using both the small molecules and the protein structures, in which the times were recorded for CP (in the version described by the original authors that finds just a single maximum clique and that would suffice for use in a screening system of the sort suggested above) and for the full version of BK (which finds all maximal cliques and that is thus the normal mode of operation

Table 3. Mean run times for matching carboxypeptidase A and leucine aminopeptidase, and the ribonuclease H and connection domains of HIV reverse transcriptase[a]

T	*S*	ρ	ω	*M*	B	BK	CP	BY	ST
				A. Carboxypeptidase A and leucine aminopeptidase					
25	n	0.049	9	2	0.40	0.38	0.15	2.50	0.34
30	n	0.059	10	1	0.40	0.45	0.17	2.80	0.49
35	n	0.069	10	2	0.67	0.54	0.20	2.80	0.51
40	n	0.080	11	1	0.94	0.65	0.23	3.40	0.58
45	n	0.090	11	1	0.88	0.77	0.28	3.80	0.72
25	y	0.025	9	2	0.32	0.27	0.10	1.30	0.20
30	y	0.030	10	2	0.32	0.29	0.12	1.60	0.22
35	y	0.035	10	2	0.32	0.31	0.13	2.10	0.26
40	y	0.041	11	1	0.32	0.33	0.13	2.10	0.28
45	y	0.046	11	1	0.33	0.36	0.14	2.50	0.32
				B. Ribonuclease H and connection domains of HIV RT					
25	n	0.031	9	1	0.19	0.05	0.02	0.38	0.07
30	n	0.034	9	1	0.19	0.06	0.02	1.50	0.08
35	n	0.039	9	1	0.19	0.06	0.03	0.49	0.09
40	n	0.043	9	1	0.19	0.05	0.02	0.06	0.08
45	n	0.047	9	1	0.19	0.05	0.02	0.40	0.09
25	y	0.016	9	1	0.19	0.04	0.02	0.28	0.03
30	y	0.018	9	1	0.19	0.04	0.03	0.28	0.04
35	y	0.021	9	1	0.19	0.04	0.03	0.28	0.04
40	y	0.023	9	1	0.19	0.04	0.02	0.28	0.03
45	y	0.025	9	1	0.19	0.04	0.02	0.28	0.03

[a] The distance tolerance is fixed at the minimum of 40% of the closest approach distance or 4 Å. Here, *T* refers to the angular tolerance and *S* denotes whether (y) or not (n) the sequence order of the secondary structure elements is taken into account when deciding that a match has been obtained.

for pharmacophore detection or database search). Values for the ratio t_{BK}/t_{CP} were found to lie within the range 2.5–3.7, this representing values for α in Eq. (2) of 0.60 and 0.73, respectively. Accordingly, we would expect to achieve an increase in the speed of a database searching program if the inclusion of the CP screen suffices to eliminate about two-thirds, or more, of the search file.

We have tested this conclusion by developing a new version of our program PROTEP, which searches the PDB for MCS-based matches with a user-defined target protein. Timed searches were carried out using the normal version, in which PDB structures are retrieved if they have a clique of at least some minimum size in common with the target structure, and using a version in which the full, BK-based match was invoked only if an initial, CP-based match showed that a sufficiently large clique was, in fact, present. The results of these experiments are shown in Table 4, where it may be seen that the inclusion of the initial screen approximately doubles the search

Table 4. PROTEP searches using a range of target structures[a]

Target	*T*	ω	t_{BK}	t_{CPBK}	t_{BK}/t_{CPBK}
1SMG	5	6	87	36	2.42
1BTV	6	12	172	90	1.91
1KBC	7	14	232	114	2.04
1AB8	7	9	241	121	1.99
1QIL	7	24	647	274	2.36
2MAS	8	29	566	290	1.95
1CJL	8	24	908	415	2.19

[a] Here, *T* is the minimal number of matching secondary structure elements for a PDB structure to be retrieved and ω, t_{BK} and t_{CPBK} are the run times for searches using Bron–Kerbosch alone and using Bron–Kerbosch with the initial Carraghan–Pardalos screen, respectively.

Table 5. ASPROTE searches using a six-side chain pattern[a]

T	t_{BK}		t_{CPBK}		t_{BK}/t_{CPBK}	
3	1 055	1 339	626	1 139	1.69	1.15
5	1 050	1 340	220	399	4.77	3.36
6	1 050	1 343	220	401	4.77	3.35

[a] Here, T is the number of side chains matching the pattern when a distance tolerance of 1.5 Å and the first and second entries in each column of the main body of the table refer to the use of specific and generic side-chain types, respectively.

speed (with, of course, no change in the matching structures that are retrieved).

We have achieved similar increases in efficiency with a further retrieval system we have developed for searching the PDB. The program ASPROTE is derived from an earlier program, called ASSAM,[42] which characterizes 3D protein structures by graphs in which the nodes are the amino acid side chains (rather than the SSEs as in PROTEP). ASPROTE uses an MCS algorithm to find those proteins in the PDB that have patterns of side chains in common with a user-defined target structure. The side chains defined in the pattern may either be specific (e.g., leucine, alanine) or generic (e.g., acidic, basic). Run times with the current version of ASPROTE, which is based on the Bron–Kerbosch algorithm, can be quite extended, owing to the density of the resulting correspondence graphs, and we have thus developed a new version incorporating an initial screening stage based on CP. The results of specific and generic searches for a 6-side chain pattern in a 570-protein subset of the PDB are shown in Table 5, which again demonstrates the substantial increases in search speeds that can result from the use of both CP and BK for database searching. The speed-up is limited at the lowest threshold, of just three side chains, but is then noticeably larger as the matching criterion becomes more stringent. In fact, still greater speed-ups (of 7.41 and 4.43 for the specific and generic cases, respectively) are obtained for these searches when a tolerance value is used (30% on the pattern distances) for which there are no matching structures in the file.

CONCLUSIONS

Clique detection provides a simple way of overlaying pairs of graphs describing 3D molecular structures, and hence of identifying the structural features that they have in common. In this article we have compared the efficiency of several different clique-detection algorithms for aligning both small molecules and proteins. Our results show that the algorithm of Carraghan and Pardalos is the most efficient for finding the maximum cliques in a graph, and is several times faster than the Bron–Kerbosch algorithm, which has previously been the method of choice for the identification of 3D MCSs. However, the latter algorithm is not restricted to the identification of maximum cliques and can thus be used, for example, to find all maximal cliques greater than some user-defined size.

Our experiments hence suggest that Bron–Kerbosch remains the algorithm of choice for those applications where there is a need to identify all maximal cliques between a pair of 3D structures; where just the maximum clique (or cliques) are required then the Carraghan–Pardalos algorithm would appear to be consistently superior. The relative merits of the two algorithms are well illustrated by the new versions of PROTEP and ASPROTE we have developed, where the inclusion of the initial screen provides a simple way to increase search efficiency without any consequent deterioration in search effectiveness.

Finally, we must emphasize the fact that our conclusions regarding algorithmic efficiency relate only to the particular types of chemical and biological correspondence graphs used here; these graphs have low edge densities, and algorithms other than the Bron–Kerbosch and Carraghan–Paradalos algorithms may be more appropriate for different types of correspondence graph.

ACKNOWLEDGMENTS

We thank the Engineering and Physical Sciences Research Council for funding EMG, and Dr. Luitpold Babel for bringing the article by Pardalos and Xue[12] to our attention. The Krebs Institute for Biomolecular Research is a designated Biomolecular Sciences Centre of the Biotechnology and Biological Sciences Research Council.

REFERENCES

1 Ash, J.E., Warr, W.A., and Willett, P. (eds.). *Chemical Structure Systems.* Ellis Horwood, Chichester, 1991
2 Good, A.C. and Mason, J.S. Three-dimensional structure database searches. *Rev. Comput. Chem.* 1996, **7,** 67
3 Martin, Y.C. and Willett, P. (eds.). *Designing Bioactive Molecules: Three-Dimensional Techniques and Applications.* American Chemical Society, Washington, D.C., in press
4 Read, R.C. and Corneil, D.G. The graph isomorphism disease. *J. Graph Theory* 1977, **1,** 339
5 Gati, G. Further annotated bibliography on the isomorphism disease. *J. Graph Theory* 1979, **3,** 95
6 McGregor, J.J. Backtrack search algorithms and the maximal common subgraph problem. *Software Pract. Experience* 1982, **12,** 23
7 Willett, P. *Three-Dimensional Chemical Structure Handling.* Research Studies Press, Taunton, 1991
8 Crandell, C.W. and Smith, D.H. Computer-assisted examination of compounds for common three-dimensional substructures. *J. Chem. Inf. Comput. Sci.* 1983, **23,** 186
9 Brint, A.T. and Willett, P. Algorithms for the identification of three-dimensional maximal common substructures. *J. Chem. Inf. Comput. Sci.* 1987, **27,** 152
10 Levi, G. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo* 1972, **9,** 341
11 Barrow, H.G. and Burstall, R.M. Subgraph isomor-

phism, matching relational structures and maximal cliques. *Inf. Proc. Lett.* 1976, **4,** 83
12 Pardalos, P.M. and Xue, J. The maximum clique problem. *J. Global Optimization* 1994, **4,** 301
13 Johnson, D.S. and Trick, M.A. *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*. Bellcore and the American Mathematical Society, 1996
14 Bron, C. and Kerbosch, J. Algorithm 457. Finding all cliques of an undirected graph. *Commun. ACM* 1973, **16,** 575
15 Martin, Y.C., Bures, M.G., Danaher, E.A., DeLazzer, J., Lico, I., and Pavlik, P.A. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. Comput.-Aided Mol. Design* 1993, **7,** 83
16 Brint, A.T. and Willett, P. Upperbound procedures for the identification of similar three-dimensional chemical structures. *J. Comput.-Aided Mol. Design* 1988, **2,** 311
17 Moon, J.B. and Howe, W.J. 3D database searching and *de novo* construction methods in molecular design. *Tetrahedron Comput. Methodol.* 1990, **3,** 697
18 Ho, C.M.W. and Marshall, G.R. FOUNDATION: A program to retrieve all possible structures containing a user-defined minimum number of matching query elements from three-dimensional databases. *J. Comput.-Aided Mol. Design* 1993, **7,** 3
19 Grindley, H.M., Artymiuk, P.J., Rice, D.W., and Willett, P. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* 1993, **229,** 707
20 Artymiuk, P.J., Grindley, H.M., MacKenzie, A.B., Rice, D.W., Ujah, E.C., and Willett, P. PROTEP: A program for graph-theoretic similarity searching of the 3-D structures in the Protein Data Bank. In: *Molecular Similarity and Reactivity: from Quantum Chemical to Phenomenological Approaches* (R. Carbo, ed.). Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995, pp. 123–140
21 Mitchell, E.M., Artymiuk, P.J., Rice, D.W., and Willett, P. Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *J. Mol. Biol.* 1990, **212,** 151
22 Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, M., and Tasumi, M. The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.* 1977, **112,** 535
23 Artymiuk, P.J., Grindley, H.M., Kumar, K., Rice, D.W., and Willett, P. Three-dimensional structural resemblance between the ribonuclease H and connection domains of HIV reverse-transcriptase revealed using graph theoretical techniques. *FEBS Lett.* 1993, **324,** 15
24 Artymiuk, P.J., Grindley, H.M., Park, J.E., Rice, D.W., and Willett, P. Three-dimensional structural resemblance between leucine aminopeptidase and carboxypeptidase A revealed by graph-theoretical techniques. *FEBS Lett.* 1992, **303,** 48
25 Artymiuk, P.J., Poirrette, A.R., Rice, D.W., and Willett, P. A polymerase I palm in adenyl cyclase? *Nature (London)* 1997, **388,** 33
26 Babel, L. Finding maximum cliques in arbitrary and special graphs. *Computing* 1991, **46,** 321
27 Balas, E. and Yu, C.S. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.* 1986, **15,** 1054
28 Carraghan, R. and Pardalos, P.M. Exact algorithm for the maximum clique problem. *Operations Res. Lett.* 1990, **9,** 375
29 Gendreau, M., Picard, J.C., and Zubieta, L. An efficient implicit enumeration algorithm for the maximum clique problem. *Lecture Notes Econ. Math. Syst.* 1988, **304,** 70
30 Shindo, M. and Tomita, E. Simple algorithm for finding a maximum clique and its worst-case time complexity. *Syst. Comput. Jpn.* 1990, **21**(3), 1
31 Gardiner, E.J. *An Evaluation of a New Clique-Detection Procedure for Matching Chemical Structures.* MSc dissertation. University of Sheffield, Sheffield, UK, 1996
32 Lin, F. Parallel computation network for the maximum clique problem. In: *Proceedings of the IEEE International Symposium on Circuits and Systems.* IEEE, Piscataway, NJ, 1993, pp. 2549–2552
33 Tarjan, R.E. and Trojanowski, A.E. Finding a maximum independent set. *SIAM J. Comput.* 1977, **6,** 537
34 Friden, C., Hertz, A., and de Werra, D. Tabaris: An exact algorithm based on tabu search for finding a maximum independent set in a graph. *Comput. Operations Res.* 1990, **17,** 437
35 Manino, C. and Sassano, A. An exact algorithm for the maximum stable set problem. *J. Comput. Optimization Appl.* 1993, **3,** 243
36 Golumbic, M.C. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980
37 Grötschel, M., Lovász, L., and Schrijver, A. Polynomial algorithms for perfect graphs. *Ann. Discrete Math.* 1984, **21,** 325
38 Tomita, E. and Fujii, E. Efficient method of finding a maximum clique and its experimental evaluation. *J. Soc. Signal Proc.* 1985, **J68**-D(3), 221
39 DePriest, S.A., Mayer, D., Naylor, C.B., and Marshall, G.A. 3D-QSAR of angiotensin-converting enzyme and thermolysin inhibitors: A comparison of CoMFA models based on deduced and experimentally determined active site geometries. *J. Am. Chem. Soc.* 1993, **115,** 5372
40 Rault, S., Bureau, R., Pilo, J.C., and Robba, M. Comparative molecular field analysis of CCK-A antagonists using field-fit as an alignment technique. A convenient guide to design new CCK-A ligands. *J. Comput.-Aided Mol. Design* 1992, **6,** 553
41 McFarland, J.W. Comparative molecular field analysis of anticoccidial triazines. *J. Med. Chem.* 1992, **35,** 2543
42 Artymiuk, P.J., Poirrette, A.R., Grindley, H.M., Rice, D.W., and Willett, P. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J. Mol. Biol.* 1994, **243,** 327