RC 20497 (90854) 7/10/96
Computational Biology 33 pages

Research Report

Flexible 3D-Substructure Matching & Novel Conformer Derivation In Very Large Databases Of 3D-Molecular Information

Isidore Rigoutsos, Daniel Platt, Andrea Califano
IBM Research Division
T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Flexible 3D-Substructure Matching & Novel Conformer Derivation In Very Large Databases Of 3D-Molecular Information

Isidore Rigoutsos[1]
Daniel Platt
Andrea Califano

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598.

Abstract

We present and discuss a new method that allows us to carry out very efficient, flexible 3D substructure matching while at the same time deriving the conformers with the best atomic overlap with the query molecule. Given are a query molecule *C*, and a typically very large database *Đ* containing information about a plurality of three-dimensional molecular structures each of which is able to exhibit torsional flexibility about one or more existing covalent bonds. Each of the molecules from the knowledge base is represented by a *single* conformer in the database; in fact, for the purposes of the method *any which* conformer suffices. The method determines those molecules from *Đ* which contain substructures in common with substructures in the query molecule *C*, in the presence of rotatable bonds. Additionally, the method determines the set of rigid transformations (i.e. rigid rotations and translations) that will put the pair (database and query molecules) in the best possible structural registration. Note that application of these computed transformations to the database molecule produces a novel conformer for the molecule; this derived conformer will in general be different from the conformer that was entered in the database, and *no* sampling of the respective molecule's conformational space is required at any time during execution. The method makes efficient use of lookup tables, thus effectively trading space for computation and exhibits superb scalability characteristics

Keywords: hashing, hash or lookup table, invariant, preprocessing, matching/derivation, 3D substructure matching, conformational space, conformer, magic vector, magic pair, star-unit, conformation initiation, conformation augmentation, hypothesis threading, conformer derivation.

1. Introduction

As existing information repositories need to be processed more rapidly, and a greater variety of tools become available, the computer plays an increasingly important role in directing and streamlining the drug discovery and design process.

One of the basic ingredients of recent approaches to this line of research endeavor has been the desire to compute, catalog and search molecular properties that are involved in the most fundamental levels of drug interaction.

Specifically, computers can help researchers to quickly eliminate a priori unlikely candidates, thus avoiding long and expensive activity screenings. More important, they can allow researchers to identify new promising compounds based only on the available information about the receptor site, or about other lead compounds.

Being able to perform these tasks quickly and to recover information that can be immediately incorporated into the formulation of the drug search strategy is expected to greatly simplify this complex, multidisciplinary endeavor and to significantly increase the speed at which new and more effective drugs are identified, tested, and brought to the market.

[1] To whom correspondence should be addressed. E-Mail: rigoutso@watson.ibm.com. Phone: (914) 784 7968.

To this date, hundreds of protein structures have been determined via X-ray crystallography and nuclear magnetic resonance (NMR) methods. This data is readily available as a public resource of molecular structure data and allows pharmacologists and biologists to investigate various aspects of protein structures and their complex behavior. In addition to these public databases, a number of other (public and proprietary) databases of small organic molecules has been assembled through the efforts of numerous pharmaceutical and biotechnology companies, and research organizations.

There are several different scenarios that one is likely to encounter in the process of drug design:

- A pharmacophore model is proposed from several active molecules; one wishes to find other molecules that either support or refute the pharmacophore hypothesis.
- A number of untested molecules potentially exhibits biological activity; one wishes to exploit existing relationships between three-dimensional structure and activity to deduce potentially present biological properties.
- A certain conformation of a given ligand has been proposed to be biologically active; a three-dimensional search is supposed to identify other molecules which mimic the ligand.
- The three-dimensional structure of a protein or DNA binding site is available through crystallographic studies. Ligands that fit the binding site are sought in this case. The search can be augmented by using information about the electrostatic behavior for some of the atomic groups at the site. This technique has been successfully applied by several pharmaceutical companies such as Merck, American Cyanamid, Agouron, etc. to design ligands that bind to and inhibit the reverse transcriptase enzyme of HIV.

In each of these cases, the critical information that enables researchers to develop hypotheses, concerning potentially new molecular candidates for synthesis and testing, must be recovered through a search in a potentially very large database of relevant structural information. The following will concentrate on the case where the databases to be searched contain structural information pertaining to 3-dimensional atomic compounds.

Typically one is given a compound/molecule C in the form of a set of coordinates of the compound's atomic sites. Also given is a database $Đ$, i.e. a collection of sets D_j. $D_j=\{x_i, y_i, z_i\}$ is a collection of sets of coordinates of the atomic sites for each one of the member molecules. Connecting the various atomic sites in both C and the database members are bonds, some of which may be rotatable and thus allow for torsional flexibility. Torsional flexibility allows the groups of atoms rigidly attached at the two endpoints of a (rotatable) bond to rotate with respect to one another. Each compound/molecule may contain more than one rotatable bond, and thus the compound/molecule can assume any of an infinite number of conformations (three-dimensional configurations) via rotations around these bonds. Occasionally, steric constraints or energy considerations may limit the number of choices, but, the cardinality of the set of possible configurations remains infinite nonetheless. This conformational flexibility of molecular structures opens a broad range of possibilities in the quest for potential ligands, while at the same time rendering the problem exponentially more difficult. In addition to the (internal) torsional flexibility, the molecules are allowed to undergo rigid transformations in three-dimensional space, i.e. the molecule as a whole can rotate and translate. In what follows, the compound/molecule C will interchangeably be referred to as 'test compound' or 'test molecule' or 'query compound' or 'query molecule.'

Given a compound C and a database $Đ$ containing information about the 3-dimensional structure of a possibly large set of molecules, the following operations need to be defined and carried out [32]:

1. "structure insertion:" the ability to incorporate all available structural knowledge about compound C in the database $Đ$;
2. "structure membership:" determination of whether the compound C is already included in the database $Đ$;
3. "substructure search:" identify and report all the member compounds from $Đ$ that contain a particular substructure of compound C;
4. "similarity search:" identify and report all the member compounds from $Đ$ that are similar to compound C. In order to implement such an operation on the database $Đ$, a similarity measure $d(.,.)$ needs to be defined and available, and
5. "superstructure search:" identify and report all the member compounds from $Đ$ that are a substructure of compound C.

First of all, it is easy to see that the predicate of structure membership is subsumed by the substructure search operation. Furthermore, all of the search operations can be reduced to what we will refer to as "substructure similarity."

In what follows, we use the phrase "3D substructure similarity" to refer to the result of a single operation which when given a compound *C*, a database *Đ* and a similarity measure *d(.,.)* allows the determination of all the compound-members of *Đ* that contain a substructure that is similar to a substructure of *C*. The extent of the similarity between the molecules in question can be determined by the function *d(.,.)*. The understanding here is that the implied common substructure may not necessarily be a proper subset of *C*. The similarity function *d(.,.)* will remain unspecified but we assume it to be of a very general nature.

The problem of 3D substructure matching can be shown to be NP-complete by noting that it includes the problem of subgraph isomorphism as a special case. The real-world implication of this statement is that the time complexity for finding all optimum solutions is an exponential function of the length of the input, and thus no efficient (i.e. polynomial time complexity) algorithm exists for finding optimum solutions. The computational complexity of the problem is further compounded by allowing torsional flexibility around the molecule's covalent bonds.

An additional distinction should be noted. This is the distinction between 'identification' and 'recognition' of those molecules from the database *Đ* of molecules which are similar to the given test compound/molecule *C*. Identification restricts itself to reporting only the identities of the molecules from database *Đ* that match the test compound/molecule *C*. On the other hand, recognition entails not only the reporting of the identities of the matching molecules but also the determination and reporting of the necessary 3D transformations that will bring each of the identified matching molecules to "best registration" with the test compound/molecule. "Best registration" positions in three-dimensional space the atomic sites of the test molecule as well as those of matching database molecules in such a way that the atomic overlap is maximized. The necessary transformations include rotations and translations of the molecules under consideration as a whole but also rotations of structures within the molecules around the molecules' torsionally flexible bonds. Recognition is arguably a much more difficult problem than identification and this becomes particularly evident in the case of very large databases *Đ* containing molecules that are torsionally flexible. This happens because the number of possible transformations increases exponentially with the number of rotatable bonds that allow for the exhibited torsional flexibility: the computational considerations for finding and reporting the correct transformation increase with the number of possible transformations.

As the number of a molecule's rotatable bonds increases so does the number of possible conformations for the molecule. The method we present allows the determination of the best possible conformer for a database molecule which can be superimposed with the query molecule such that the atomic overlap is maximized. Note that the conformer that this method derives will in general be different than the single conformer that has been stored in the database. It is important to realize that the best conformer-answer will be derived *without* any search or sampling of the conformational space of the molecule under consideration; as a matter of fact, the performance of the method improves as the size of the conformational space increases, a rather counter-intuitive result!

Our method proceeds in two phases. During the first, *off-line* phase, and for each molecule in the knowledge base, a single conformer is selected. This selection can be made in an educated manner, but any conformer from the infinite set of conformers for the molecule under consideration can be selected and used to produce multiple entries in a lookup table. What is stored in the lookup table during this phase is implicit information about the molecule's ability to *assume* new conformations via rotations about its rotatable bonds.

In the second of the phases, which takes place *on-line*, the stored implicit information is exploited in a manner that is reminiscent of the pose clustering method and hypotheses are produced which correspond to placements in the three-dimensional space of various molecules' rigid groups. Substantial filtering takes place during this phase; moreover, the use of a lookup-table guarantees that this set of hypotheses is produced fast. During a final step, knowledge of the connectivity properties of each molecule's rigid groups, from the hypothesized set of molecules, is used to thread these hypotheses into full-fledged conformations for each molecule-answer. It should be stressed again that no search of the molecule's conformational space is required during the threading step. As a result, the database molecules for which a conformation exists that can exhibit maximum atomic overlap with the query molecule will be reported in the appropriate conformation in a very time-efficient manner.

In the remaining sections we describe our method in detail. In Section 2, we review previous suggestions for tackling the problem. In Section 3, we introduce our method: the method is first presented and explained for the case of rigid molecules (no rotatable bonds); we then show how it directly extends to the case of molecules with rotatable bonds. Section 4 gives the general description of the algorithm, and Section 5 presents some results from experiments with the National Cancer Institute's databases of carcinogens containing in excess of 123,000 molecules. A discussion of the algorithm and of its ramification is given in Section 6. Various details of the algorithm that will facilitate the method's implementation are presented in Appendices 1 through 4. A very thorough description of the method and algorithm can also be found in [48].

2. Previous Suggestions For Tackling The Problem

The inherent computational complexity of the substructure similarity task has typically plagued all of the previously suggested approaches for tackling this problem. Even when one restricts the problem to the case of rigid molecules with no rotatable bonds, the problem remains computationally very demanding because of its three-dimensional nature.

In order to appreciate the complexity of the task, a one-dimensional analog from everyday life is presented. Given a shelf full of books and a sentence such as

"forming 3D queries that can accommodate certain flexibility in the target structures,"

a search task is defined as the need to find all occurrences of similar phrases in the set of available books.

Similar, in the most general case, means that a phrase such as

"we form a 3D search query so that it accommodates desired flexibility"

should be reported as being a valid match. In other words, operations such as replacement, insertion, and deletion of the most fundamental information elements (in this case the letters) are legitimate and thus allowed. (In the case of molecules the most fundamental information elements are the atoms of a molecule).

A straightforward approach to solving this task entails the scanning of the contents of all the books of the shelf in an exhaustive, linear fashion, i.e. left-to-right, top-to-bottom, in order to locate all similar (in this case one-dimensional) structures. Clearly, such a mode of operation will require increasingly more time as the number of books on the shelf (i.e. the size of the database) increases.

Of course, a number of heuristics can facilitate finding the answer. For example, certain operations may not be allowed, or the search could be restricted to a smaller, well-specified set; this immediately restricts the number of possible variants for a given phrase and makes precomputation and storing of alternative phrases plausible. When searching, the test phrase is compared against the set of all allowed, precomputed variants.

Alternatively, "keys" can be precomputed using a subset of words within a window of pre specified width, and stored. When presented with a query the system computes the set of keys for the query and uses them to search and find the set of keys which have been computed for all the phrases in all the books. In other words, instead of comparing the phrases directly with one another, their "representatives" are compared instead, again in a linear fashion.

A situation analogous to the above similar-phrase search task exists in the case of searching for similar structures in databases of three-dimensional molecular information. The following presentation of representative techniques intends to help identify the commonalities and differences of previously suggested approaches.

The various techniques that have been proposed over the years for searching three-dimensional molecular databases basically differ in their definition and their use of the similarity measure *d(.,.)* which was introduced above. Given a test compound *C* and a database *Đ*, the similarity measure *d(.,.)* evaluates the extent in which *C* and a given member of *Đ* are similar. The values produced by comparing *C* to each one of the member molecules of *Đ* will produce a "score" that can subsequently be used to rank candidate answers in order of decreasing quality.

For example, in the "atom-mapping" method [42] the Tanimoto coefficient is computed using the result of pair-wise comparisons of the rows from the distance matrices of two molecules. This coefficient is used as an entry to an intermolecular similarity matrix. This matrix is used in conjunction with a greedy algorithm to establish the degree of similarity between the two molecules. The calculation is repeated for all the combinations between a query molecule *C* and each of the molecules in the database *Đ*. As is the case with greedy algorithms, no guarantees exist that the algorithm will find all of the correct solutions. The approach is computationally very demanding and does not scale well with database size.

In the "clique-detection" method [26] a number of different orientations is generated for each of the molecules in the database prior to comparing it to the query molecule *C*. Each of the orientations is then overlaid on *C* and scored based on the presence or not of database atoms in the vicinity of an atom of *C*. All of the orientations that lead to score less than the maximum are discarded. The search then continues with the next molecule of the database. At any point during the search, the last *n* best scores are retained. This technique is at the core of the Mosaic molecular modeling system [37].

In other prior art techniques the molecular structures are represented as connection tables, and thus viewed as graphs. The vertices of each such graph correspond to the molecule's atomic sites. If a bond exists between two given atomic sites, then the corresponding graph will have an edge connecting the relevant nodes. If each molecule of the database *Đ* is represented by a graph, one can carry out a search for similar substructures by using subgraph isomorphism algorithms. As we have already mentioned above, the problem of subgraph isomorphism is NP-complete and thus no efficient algorithms exist. Recent work compared a number of different subgraph isomorphism algorithms and presented evidence for the usefulness of a backtracking search algorithm enhanced with the "refinement procedure" heuristic [52]. Graph-theoretic results are also used to develop similarity functions for comparing molecular fragments (substructures) [12].

A variation of the above scheme begins by clustering all of the molecules in the database *Đ* into several clusters. In this case, the similarity measure *d(.,.)* is first used to calculate intermolecular similarities for all the pairs that can be formed by molecules in *Đ*. Subsequently, a clustering step is used to group the various molecules into clusters based on the values produced by the pair wise comparisons. When presented with a query molecule *C*, this approach classifies *C* by identifying the cluster in which *C* belongs. The molecules of database *Đ* which best match the query molecule *C* are drawn from this cluster as well as the neighboring cluster (or clusters) [54].

So far, the assumption has been that the molecules under consideration are rigid three-dimensional structures. But most of the time, this is not the case. Usually, molecules have several internal rotatable bonds, and are thus able to assume a continuum of conformations, i.e., three dimensional configurations. Occasionally, steric constraints or energy considerations may limit the number of choices.

Treating the molecules of a database as rigid facilitates the search in 3D databases at the expense of discarding large numbers of valid candidates: although the stored conformation of a molecule may not exhibit the pharmacophoric pattern/model under consideration, a different conformation of the same molecule may be biologically active. Thus, conformational flexibility of molecular structures opens a broad range of possibilities in the quest for potential ligands. But, at the same time, it imposes a serious burden on the search component of the conventional approaches.

Given a database *Đ* of molecules, one straightforward approach that allows any search algorithm to carry out conformationally flexible searches in *Đ* entails the storing of all the conformations of each of the molecules in *Đ*.

In practice, given that there is a continuum of possible conformations, a large number of representative conformations is stored instead. The implications of such an approach are evident: the resulting databases will have overwhelming sizes and very long search times will be necessary. An alternative approach to storing all possible conformations involves storing each molecule in only one conformation. For example, the Concord-3D system [10] uses a set of rules to generate a single conformation using the molecule's connection tables. Midway to these two extremes lies another approach according to which a small number of carefully selected conformers (called "maximally dissimilar" conformers) replaces each molecule and account for the flexibility that the molecule exhibits; these smaller databases have been used to carry out flexible docking and also increase the utility of docking methods in finding new leads [25,35]. These approaches in essence belong to a class of methods that puts the flexibility in the database.

In an analogous manner, the Chem-X system [7], a variant of this method applies a set of rules (determined by performing a systematic conformational analysis on chains of various combinations of six backbone atoms) to examine the conformational space and retain only certain torsion angles for each rotatable bond: a set of "low-energy" conformations together with their corresponding "screens" is produced. These screens are subsequently used during the actual search of the database. In a related approach, a large number of conformations of a database molecule is generated during the search and compared with the pharmacophoric pattern [38]. This is a computationally heavy approach, and any attempts (through the use of heuristics) to reduce this burden have a direct impact on the quality of the produced results: otherwise valid matches will now be missed. Both of these methods are representative of a class of techniques that put the conformational flexibility into the search.

There is also a third approach, where the flexibility is put in the query itself [8,20]. The query in this case combines both rigid and flexible components and is iteratively refined by searching a database of compounds with "known" activity until the desired selectivity is obtained. Once the final query is available, it is used to search a database of compounds with "unknown" activities to identify potential leads.

The more successful search techniques attack the problem of conformational flexibility in a computationally demanding way; the conclusion of previous [9] work was that flexible three-dimensional searching using the approach developed by Clark et al. incurs a minimum of a 100-fold slowdown over the rigid matching case. This slowdown appears to be typical and independent of the actual technique that is employed [36].

In a comparison study by Haraki et. al. [21] it was shown that augmenting a database with multiple conformations of a given molecule enhances in general a search algorithm's performance. But the same study also established that the resulting effectiveness largely depends on the method which is used to generate the various conformations to be added in the database.

As an alternative to the multiple inclusion of a molecule in the database, a certain type of minimization is carried out in "discrepancy space" [24]. This approach is much faster but it requires certain relations between the number of structural constraints and the number of existing rotatable bonds; furthermore, it inherits all of the problems of non-linear optimization approaches.

Yet another type of technique tries, essentially, to perform rigid docking only on the rigid subparts of the molecule and then check the compatibility of the various docked parts in a post-processing phase. This technique is in general computationally very demanding [11].

A number of shortcuts in the form of search heuristics have been introduced to alleviate some of the computational burden, but not without adverse impact on the quality of the produced results. To counterbalance this statement, these heuristics are of a more general applicability and can also be used in the case where conformational flexibility is not one of the parameters of the problem.

In particular, some prior art [13,14] gives a very thorough presentation and carries out a comparative study of a number of descriptors for the purposes of database screening. The descriptors cover a large range of properties of the molecules in the database: physical, chemical, geometric as well as several combinations thereof. The discrimination ability of some of the proposed descriptors is encouraging but the results have been obtained using a small database with only a few thousand compounds.

Related work [39] introduces a two-stage method that in essence characterizes shape without the need for examining a multitude of docking orientations. In the first stage, a 2048-bit number is generated for each of the molecules in the database by setting appropriate bits in a bit-vector. The bits to set are selected based on a 32-bit encoding of each triangle formed by three atomic sites in the molecule. Clearly, the number captures geometric characteristics which are particular to each molecule; however, due to the way it is generated, the representation is not unique. During the second stage, a similar 2048-bit is generated for the test molecule and is compared with each of the stored signatures. For those molecules whose signatures exceed threshold, triplets of atomic sites are formed again and compared against the triplets in the test molecule for intersection. Although the method does not recover the relative orientation between a candidate and the test molecule it appears to be adequate as a screening step.

In all of the described prior art, the techniques either do not scale well with the size of the database (due to the need for a serial scan and processing of all the entries), or do not fully exploit the constraints imposed by the rotatable bonds in order to limit the extent of the search.

Unlike the techniques that require a linear scan of the database *Đ*, hashing techniques [6,28,27] are based on the identification of certain invariant descriptors (indices) that can be used to store in a lookup table a partial representation of, say, a molecule. Compatible molecules can be retrieved by computing the indices from a test input [15,16,49,40], retrieving the partial representation from the lookup table, and integrating the evidence directly, thereby eliminating the need to scan the entire database for one or more matches. For molecules, indices can be formed by using tuples of atoms (e.g., triplets) with atom properties or tuples of small surface patches associated with their normals and the chemical properties at the surface (two independent patches are sufficient in this case.

In previous work it has been argued that using indices of a high dimensional nature (with a large number of distinct values) is crucial for the correct behavior of these techniques when the size of the database becomes large [5,6,46]. Two main issues contribute to this very general result. First, bins in a lookup table with a larger set of bins will be, on the average, less crowded. And second, coarser quantization can be used along each of the index dimensions thus increasing the probability of recovering the same index during retrieval of similar entities.

Unlike the scan-based techniques though, the class of hashing algorithms has increased storage requirements. In particular, the various incarnations of the algorithm derive their speed by precomputing results and storing them in appropriately constructed lookup-tables. This precomputation can be performed off-line, is done only once, and the results are stored on disk and used when needed. The hashing approach in essence trades space for computation; in the face of decreasing slow-storage costs, the tradeoff is becoming increasingly justifiable and reasonable.

3. The Method

In this section, we will describe the suggested method in detail. We begin the presentation with the definition of the problem.

3.1 The Problem Definition

We are given a query molecule *C* in the form of a set of coordinates of the compound's atomic sites. Also given is a database *Đ* which is a collection of the sets of the atomic sites' coordinates for each one of the member molecules. Connecting the various atomic sites in both *C* and the database members are bonds, some of which may be rotatable and thus allow for torsional flexibility: the groups of atoms attached at the two endpoints of a rotatable bond can rotate with respect to one another. A molecule may contain no rotatable bonds (*rigid* molecule). Alternatively, a molecule can contain one or more rotatable bonds (*torsionally-flexible* molecule), and can thus assume any of a possibly infinite number of conformations via rotations around such bonds. The molecules, whether rigid or torsionally-flexible, are also allowed to undergo rigid transformations in three-dimensional space, i.e. they can rotate and translate. Figure 1 shows a molecule with two rotatable bonds 18a and 18b, connecting the three rigid groups 10, 20 and 30; the molecule comprises a set of atomic sites that have coordinates in the global coordinate system 35.

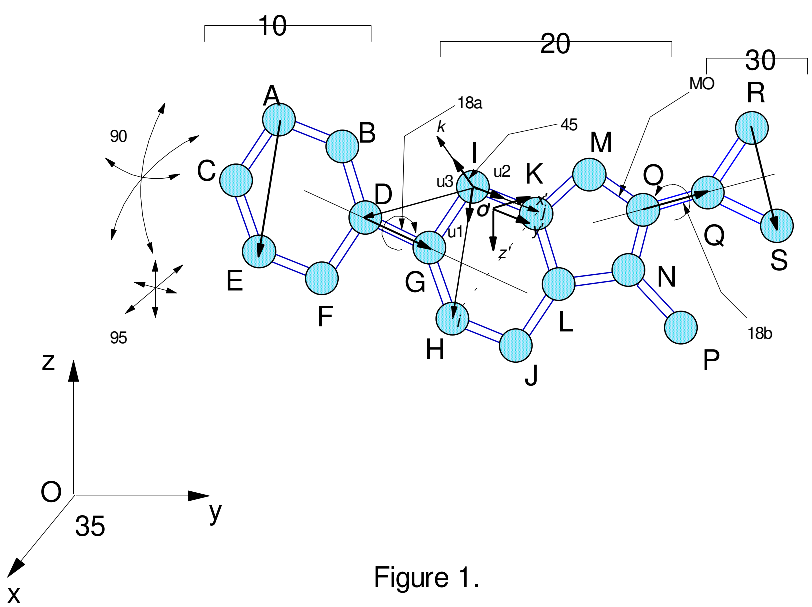


Figure 1.

The task at hand is to determine, for the given compound *C*, database *Đ*, and similarity measure *d(.,.)*, all the compound-members of *Đ* that contain one or more substructures similar to one or more substructures of *C,* the extent of similarity being determined by the function *d(.,.)*. An implicit assumption here is that the implied common substructures may not necessarily be a proper subset of the molecule *C*. Also, the similarity function *d(.,.)* can be assumed to be of a very general nature.[1]

For each database compound found, the method we present will in essence identify *all* substructures that are in common to both the query and some database molecule and derive a novel conformer for the latter; the derived conformer is placed in three-dimensional space in a way that will maximize the atomic overlap between the query molecule *C* and each answer molecule from *Đ*, in the presence of possible rotatable bonds in either or both. As such, the method to be described implements a set of operations that subsumes the entire set described at the bottom of page 2.

3.2 A High-Level Description

The method we propose herein has two stages. During a preprocessing stage, information pertaining to the molecules in the database is stored in a hash table. This stage occurs *off-line* and since it will take place only once, it is allowed to be computationally expensive. Performance is critical during the second stage, that of (structural) matching/derivation. During this latter stage which is performed *on-line*, a query molecule is used to

[1] It should be noted that the concept of substructure as used in this work does not refer to degenerate substructures of one atom. Indeed, the *smallest* possible common substructure is directly dependent on the size of the smallest set of atoms needed to define a structural invariant (index); in our case, the smallest such set includes a minimum of 3 atoms.

access locations of the database's lookup table in order to determine both the identity and the necessary transformations of those molecules in the database that can best match the query molecule. What is meant here by *best match* is that the method will determine the identity of database molecules and their placement in three-dimensional space with respect to the query molecule so that any and all substructures that are shared by both the query and each database molecule will occupy the same positions in space. In doing so, the method in essence derives a novel conformer for each molecule/answer (a conformer in general different that the one used to make the entries in the lookup table) *without* searching the respective molecules' conformational spaces; each derived conformation will have maximum atomic overlap with the query molecule.

The fact that the method is a two-stage one is reminiscent of the geometric hashing approach from where we borrow ideas, but it is distinct from traditional geometric hashing in several aspects. In particular, let C be the query compound, and D a member of the database Đ that shares one or more substructures with C. Let N_C and N respectively be the numbers of the atomic sites comprising each molecule. Both C and D contain bonds, but for the moment we assume the molecules to be structurally rigid.

Let us consider the set of all possible tuples t_j that one can form by considering the atomic sites of molecule D. These tuples could be 2-tuples (i.e. comprising 2 atomic sites), 3-tuples, or, in the general case k-tuples, where $k=|t_j|$ is the cardinality of the tuple. For each one of these tuples, t_j, the traditional geometric hashing approach will form a Cartesian coordinate system in which it expresses the coordinates of the remaining $N - |t_j|$ sites. It is these coordinates that are used to access appropriate locations of a hash table; in the accessed locations information about the identity of the model D, the atomic site whose coordinate was computed, and the tuple t_j is stored. The number of the entries stored in the hash table for molecule D is bounded above by $\binom{N}{|t_j|}\left(N-|t_j|\right)$.

In another variation of the algorithm [40], two-site tuples are used: in particular, ordered pairs of atomic pairs are formed and combined with other atomic sites of the molecule under consideration. The appropriate hash table locations to access are now determined by the lengths of the sides of the formed triangles; again, the information that is stored in the bin includes information about the identity of the model D, the atomic site whose coordinate was computed, and the ordered-pair t_j. The benefits from such a choice are a decrease in the computational complexity of the method and the required storage space, at the expense, of course, of discriminatory power.

During the recognition phase of geometric hashing, accumulators are set up for all the tuples t_j of each of the database molecules: a tuple t_c of the same cardinality as in the preprocessing phase is formed and the remaining atomic sites' coordinates are expressed in the coordinate system of the tuple's members and the corresponding hash table bins are accessed. The identities of the tuples found in the accessed bins are used to increment the respective accumulators. The highest accumulator counts are then determined: for each one of those tuples, the transformation which will bring it in registration with the tuple t_s is computed and used to rotate and translate the respective molecule; the molecule in its new position and orientation in space is then verified against the test molecule. If the verification succeeds, the tuple corresponding to the accumulator under consideration becomes a *candidate match.* Then the procedure is repeated for another tuple from the test molecule. When *all* possible tuples t_c of the test molecule are exhausted[1] the consistent *candidate matches* are clustered together based on the proximity of the respective translation and rotation parameters. Each of the formed clusters corresponds to an answer to the question of which transformation the molecule D must undergo in order to best match the test molecule C.

The geometric hashing method operating as described above cannot handle molecules that exhibit torsional flexibility. Extensions to the method have been proposed which can cope with *the hinge*-like flexibility exhibited by large molecules such as proteins, but these extensions to the geometric hashing framework have a strong Hough-transform flavor; consequently, the performance benefits stemming from the use of lookup tables are greatly diminished.

[1] Occasionally, not all tuples need to be exhausted, but a percentage of all possible ones suffices.

Generally speaking, our work borrows from both pose clustering [50] and geometric hashing. Indeed, the method uses a lookup table to quickly generate hypotheses about those positions and orientations that would reveal any matching substructures between the test and database molecules. The generated hypotheses are then clustered together using a variation of the pose clustering approach, but, unlike the latter, our method does *not* require the exhaustive search of the information pertaining to *all* of the database molecules.

Another very important characteristic distinguishing our method from all previous molecular matching work has to do with the fact that our formulation and the resulting algorithm treat *uniformly* both the rigid and the torsionally-flexible molecule cases.

3.3 The Case Of Rigid Molecules

We will begin the description of the method by first discussing the case of rigid molecules, and will then proceed by generalizing to the case of molecules with any number of rotatable bonds.

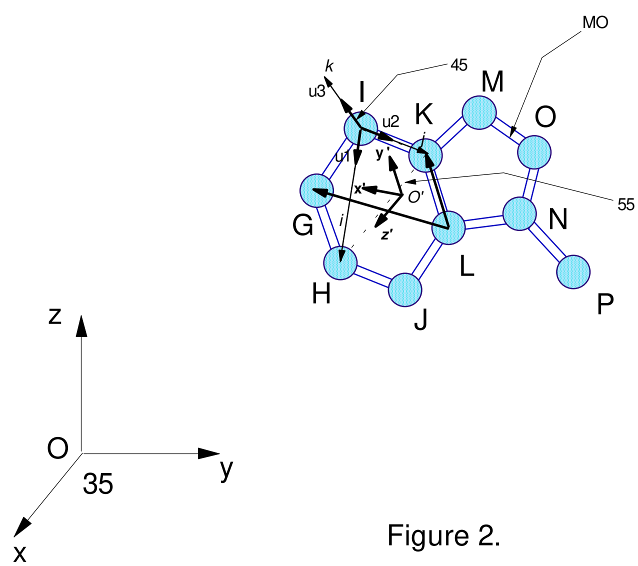


Figure 2.

In Figure 2, a rigid molecule is shown. This particular molecule was selected for clarity purposes; it consists of one rigid substructure comprising the atomic sites with labels G through P. The atoms comprising the molecule in question are connected with one another through chemical bonds, typically MO. We will be designating bonds by a pair of letters that correspond to the labels of the respective atomic sites. The coordinates of the various atomic sites are expressed in the global coordinate frame 35. Groups of atoms such as P are not considered to be separate from the group G-O despite the fact that the bond NP may be rotatable. The reason for this is that any rotation around the bond NP will not change the position of P in the global coordinate system. Moreover, any such rotation will not change the relative placement of P with respect to the group of G-O either.

Referring to Figure 2, we can see that defining the coordinates (x, y, z) of three or more (non-collinear) atoms in the global coordinate frame suffices to define a global position and a global orientation (*O'x'*, *O'y'*, *O'z'*) for the rigid molecule in the global coordinate frame. Furthermore, the selected three sites define a, possibly skewed, local coordinate frame 45. Every remaining atomic site of the rigid molecule has a set of coordinates (i.e. a position) in the local coordinate frame; these coordinates remain unchanged when the molecule undergoes a three-dimensional rigid transformation (rotation and translation).

Also depicted in Figure 2 is what we refer to as a *magic*[1] *pair* comprising two *magic vectors,* namely the vectors $\overrightarrow{LK}$ and $\overrightarrow{LG}$. These two vectors are shown defined in terms of two pairs of atomic sites. Although the particular vectors shown in this figure have a common tail (atomic site L) this does not have to be the case; for example, the magic pair could comprise any two non-collinear vectors such as $\overrightarrow{LO}$ and $\overrightarrow{JG}$. In general, each magic vector is defined in terms of endpoints E_i, $i = 1, 2$ that can be uniquely determined as a function of the coordinates of the atoms comprising the rigid structure with which the magic vectors are associated. It is important to realize that positioning the pair of magic vectors in the three-dimensional space immediately positions the entire rigid molecule; this is a direct consequence of the way the vectors were constructed.

Given the two endpoints defining a magic vector, the vector's direction can be decided in a number of alternative ways; one choice for the positive direction could for example be from the atom with the lexicographically smaller label to the one with the higher label.

Also, it is clear that the two non-collinear magic vectors form a coordinate frame, the *magic coordinate frame* (this is frame 55 in Figure 2). The frame is assumed to have its origin at the center of mass of the four endpoints defining the respective magic vectors. The frame's X and Y principal axes are defined by translating the two magic vectors so that their origins coincide with the origin of the magic frame. The non-collinearity assumption guarantees that the their cross product exists and thus it forms the Z axis of the frame. Finally, it should be pointed out that since the magic vectors are not orthonormal, the coordinate frame will be generally skewed.

The two magic vectors are by definition rigidly attached to the associated rigid structure and thus move together with the molecule whenever the latter undergoes rigid transformations. As such, each of the magic vectors has a fixed position and orientation in any *local* coordinate frame (e.g. 45 in Figure 2) that can be defined by means of three (non-collinear) atoms. I.e. for the $j-th$ magic vector, there is translation vectors T_{ij} and rotation matrices R_{ij} that uniquely determine the position of the two endpoints $E_i, i = 1, 2$ in each local coordinate frame. Given the definition of any one local coordinate frame and the appropriate set of translation vectors and rotation matrices, the magic pair associated with the molecule can be reconstructed unambiguously. Moreover, since the same three sites that define the local coordinate frame have a fixed position and orientation in the global coordinate frame, the translation vectors T_{ij} and rotation matrices R_{ij} allow the unique determination of the position and orientation of the magic pair (and thus of the molecule) in the *global* coordinate frame.

Given a rigid molecule with *N* atoms, a magic pair is defined for the molecule and fixed. Triplets of non-collinear atomic sites can be identified and selected in the molecule; at most, $\binom{N}{3}$ unordered and $\binom{N}{3}3!$ ordered triplets can be formed. To each formed triplet we associate a set of translation and rotation matrices representing the position and orientation of the magic pair in the respective local coordinate frame. Alternatively, any other, equivalent to the set of translation and rotation matrices, representation of the magic pair could be used.

The association of each local coordinate frame and the appropriate translation and rotation matrices is accomplished with the help of a lookup table. The lookup table consists of a number of hash bins that can be indexed by means of the geometric properties of the triplet of atoms forming the local frame. Occasionally, physical, chemical or other properties of the triplet of atoms (or of the molecule) may be used to generate the index. Given a triplet of atoms, the identity of the appropriate hash bin is determined and an entry is made in the bin. The entry contains information about the identity of the molecule, the translation and rotation matrices (or equivalent information) that allow the reconstruction of the magic pair, and possibly other pertinent information. A given hash bin will in general contain more than one such entry, the latter corresponding to one or more distinct molecules.

---

[1] The choice of the adjective *magic* was dictated by what will become apparent further in our discussion: despite its apparent simplicity, the concept of the two magic vectors proves to be a powerful tool that allows the efficient handling of a computationally demanding task when defined and exploited in an appropriate manner.

The preprocessing stage of the algorithm completes when all of the triplets in all of the molecules in the database of recognizable (rigid) molecules have been associated with the appropriate translation and rotation matrices corresponding to each molecule's magic pair. At the end of the processing, a lookup table that encodes all of these associations is available for the second stage of the method, that of matching/derivation.

The matching/derivation stage is an on-line process. The input to this stage is a query molecule *C* comprising *N* atoms; the molecule is assumed rigid. Non-collinear triplets of atoms are again formed by selecting from among the atoms of *C*. For each such triplet an index is produced and used to access the contents of a hash bin. Every entry found in the accessed bin contains information about the translation and rotation matrices of some molecule in the database; these matrices are used to reconstruct a *hypothesis* for the position and orientation (in the global coordinate frame) of the respective molecule's magic pair.

The lookup table in essence associates a triplet of atoms in the query molecule with a triplet of atoms in a molecule of the database. It is clear that by design the employed lookup table has a dual functionality:

- during the preprocessing stage, each bin of the lookup table effectively clusters together those database molecules that contain a triplet of atoms with the same properties. The properties could be structural, physical, chemical etc. and are captured by the index that corresponds to the respective hash bin.
- during the matching/derivation stage, the lookup table helps associate each atom triplet of the query molecule with all those database molecules containing an atom triplet with the same (structural, etc.) properties. This association is achieved by indexing into bins of the lookup table using properties of triplets of atoms from the *query* molecule. A common substructure between the query molecule and a molecule in the database will give rise to reconstructed hypotheses that are *consistent* with one another and corroborate the same position and orientation for the magic vector of the database molecule.

Recall that the position and orientation of the magic pair uniquely identifies the position and orientation of the entire molecule in the global coordinate frame. Thus, these reconstructed hypotheses for the position and the orientation of a magic pair are in essence hypotheses for the position and orientation of the respective entire molecule. We will be making use of this observation later when discussing the case of torsionally-flexible molecules: the reconstructed hypotheses will be referring to the position and orientation of a molecule's *rigid segment* instead.

After all of the query molecule's triplets have been exhausted, there will be a collection of hypothesized (Molecule-Id, Magic-Pair-Position-and-Orientation) tuples not all of which will be distinct. In other words, a given tuple may appear in the collection more than once. The degree of repetition is analogous to the number of atom triplets made up of atoms participating in the substructure which is common between the query and the molecule indicated by the tuple: all these triplets will be reconstructing the same (within tolerance) hypothesis for the position and orientation of the candidate molecule's magic pair.

In particular, since it may happen that the common substructure as it appears in the query molecule is a slightly distorted version of the substructure in the respective database molecule, the tuples (Molecule-Id, Magic-Pair-Position-and-Orientation) that are reconstructed by atom triplets in the common substructure will not be identical; instead, there will be slight differences with respect to one another. Figure 3 exemplifies this statement. Despite this observation, the expected differences do not prevent the method from clustering all such hypotheses into one with support equal to the cardinality of the contributing set. This has also been observed experimentally.

Once all of the consistent hypotheses have been clustered[1] they can be ranked in order of decreasing support. At the top of the list, a tuple (Id-1, Position-1 and Orientation-1) with the largest support will be found. This in essence translates into the following statement: the molecule with Id-1, when placed in the global coordinate frame in such a way that its magic pair have the position and orientation dictated by (Position-1 and Orientation-1), will exhibit the maximum degree of structural overlap with the query molecule *C*. The remaining

[1] The clustering can be achieved easily with a histograming operation.

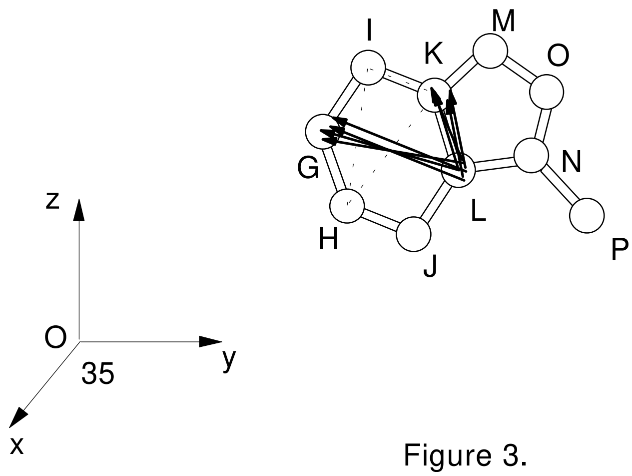


Figure 3.

items on the list will correspond to molecules with decreasing degrees of structural overlap (smaller-size common substructures).

3.4 Extending The Method To The Case Of Torsionally-Flexible Molecules

A straightforward extension of the above-described method allows us to handle the general case where the involved molecules exhibit torsional flexibility.

Let us recall again the case of the molecule in Figure 1. The molecule consists of three *rigid* groups denoted 10, 20 and 30. The rigid groups are connected with one another via the rotatable bonds 18a and 18b that allow for the torsional flexibility of the molecule.

The two rotatable bonds allow a rigid group to rotate with respect to the neighboring group; during such rotations, the distance between any two atoms that belong to the same rigid group remains the same. However, triplets made out of atoms belonging to more than one rigid group have structural properties that, in general, will change when one of these rigid groups rotates around one of the rotatable bonds emanating from it. Since such triplets of atoms do not exhibit structural properties that are fixed, any indices produced by these triplets cannot be used as an invariant for accessing the lookup table built during preprocessing.

Having stated this, let us concentrate for the moment on the rigid group 20 of Figure 1, ignoring the rotatable bonds 18a and 18b emanating from it. It should be clear that if we augment the rigid group 20 by also including the atoms labeled D and Q, the new group will still be rigid. Indeed, notice that independent of any rotation around the bond D-G of rigid group 20 with respect to the rigid group 10, the position of atom D relative to any of the atoms G through P remains *unchanged*. A similar argument holds for atom Q. We will be referring to this new set of atoms as the *augmented* rigid group 20.

Again, ignoring the presence of the two rotatable bonds 18a and 18b, we can next view the augmented rigid group 20 as a rigid *molecule* in its own right. As such, a magic pair can be defined using atoms belonging to the augmented set {D,G,H,I,J,K,L,M,N,O,P}.

As already explained above, the magic pair can be defined with the help of any set of atoms in a rigid molecule, as long as the resulting vectors are not collinear. Consequently, there is nothing that prevents us from defining the magic pair associated with the rigid molecule in question (i.e., the augmented rigid group 20) as comprising the two vectors $\overrightarrow{DG}$ and $\overrightarrow{OQ}$.[1]

An analogous situation exists for the two rigid groups 10 and 30. Let us concentrate on rigid group 10. If we augment this group by also including the atom labeled G, the new group will still be rigid. Indeed, notice that independent of any rotation around the bond D-G of rigid group 20 with respect to the rigid group 10, the position of atom G relative to any of the atoms A through F remains *unchanged*. Consequently, we can view the augmented rigid group 10 as a rigid molecule in its own right, and magic vectors can be defined using any of the atoms belonging to the set {A,B,C,D,E,F,G. We define the magic pair for rigid group 10 as comprising the vector $\overrightarrow{DG}$ and any other non-collinear vector defined using two atoms of the augmented group. Note, for example, that the vector $\overrightarrow{GB}$ would be a valid choice. Similarly, the augmented group 30 will contain the atom with label O in addition to the atoms Q, R and S; the group's associated magic pair will comprise the vector $\overrightarrow{OQ}$ and any other non-collinear vector defined using two atoms of the augmented group.[2]

Having performed all these augmentations, the method for the case of rigid molecules can be extended to this general case with the following two modifications:

- during the preprocessing, phase lookup table entries are built that contain information about the identity of the molecule, the identity of the rigid group under consideration, the translation and rotation matrices (or equivalent information) that allow the reconstruction of the magic pair associated with the rigid group, and possibly other pertinent information. A given hash bin will in general contain more than one such entry corresponding to one or more distinct rigid group and/or molecules.
- during the matching/derivation phase, the hypothesized tuple which for the case of rigid molecules had the form (Molecule-Id, Magic-Pair-Position-and-Orientation) will now be modified to be (Molecule-Id, Rigid-Group-Id, Group's-Magic-Pair-Position-and-Orientation); this latter form of the hypothesis takes into account the fact that our concept of a rigid molecule is now equivalent to a specific rigid group in a given molecule.

As before, when presented with a query molecule (possibly containing rotatable bonds), triplets of atoms are formed by selecting from among the atoms forming the query molecule. It is not necessary to distinguish rigid groups in the query molecule, although one is certainly not prevented from doing so. Each of the formed atom triplets will give rise to an index that will be used to access a bin in the lookup table; for each of the entries in the accessed hash bin a hypothesis (Molecule-Id, Rigid-Group-Id, Group's-Magic-Pair-Position-and-Orientation) will be reconstructed.

After all of the query molecule's triplets have been exhausted, there will be a collection of hypothesized (Molecule-Id, Rigid-Group-Id, Group's-Magic-Pair-Position-and-Orientation) tuples not all of which will be distinct. As before, a given tuple is likely to appear in the collection more than once. The degree of repetition is analogous to the number of atom triplets made up of atoms participating in the substructure which is common between the query and the indicated rigid group of the molecule specified by the tuple: all these triplets will be reconstructing the same (within tolerance) hypothesis for the position and orientation of the magic pair for the candidate rigid group and molecule combination.

Having produced this collection of hypotheses for the positions and orientations of various rigid groups in different molecules, we next set out to answer the question of whether there is a substructure common to both the

[1] Here, we used the lexicographic ordering of the atom labels to determine the direction of the magic vectors.
[2] In Appendix 2 we examine the issue of magic pair generation in more detail.

query and some torsionally-flexible database molecule. It is a maximum-size substructure that we are seeking and this substructure will generally span more than one rigid groups. It should be evident that since the database molecules are assumed to be torsionally-flexible, a maximum-size common substructure will reveal itself only if the database molecule is placed in the appropriate conformation.

Each element of the collection of the hypothesized tuples (Molecule-Id, Rigid-Group-Id, Group's-Magic-Pair-Position-and-Orientation) in essence states the following: if the rigid group *Rigid-Group-Id* of molecule *Molecule-Id* is positioned in the three-dimensional space so that its associated magic pair assumes the position and orientation indicated by the tuple's *Group's-Magic-Pair-Position-and-Orientation*, then the molecule *Molecule-Id* and the query molecule will overlap; the amount of structural overlap will be analogous to the number of atom triplets that have supported the tuple in question.

But each of these tuples provides additional information: indeed, it also dictates the position and orientation of the respective rigid group's magic pair. Recall at this point that we have defined the magic pair for the rigid group 20 of Figure 1 by means of the two rotatable bonds emanating from it. Any hypothesis tuples produced for this group will in essence dictate possible position and orientation choices for the *rotatable bonds* 18a and 18b such that any existing structural overlap with the query molecule is revealed. A similar statement holds true for the hypothesis tuples corresponding to the rigid group 30 whose magic pair includes the rotatable bond 18b as one

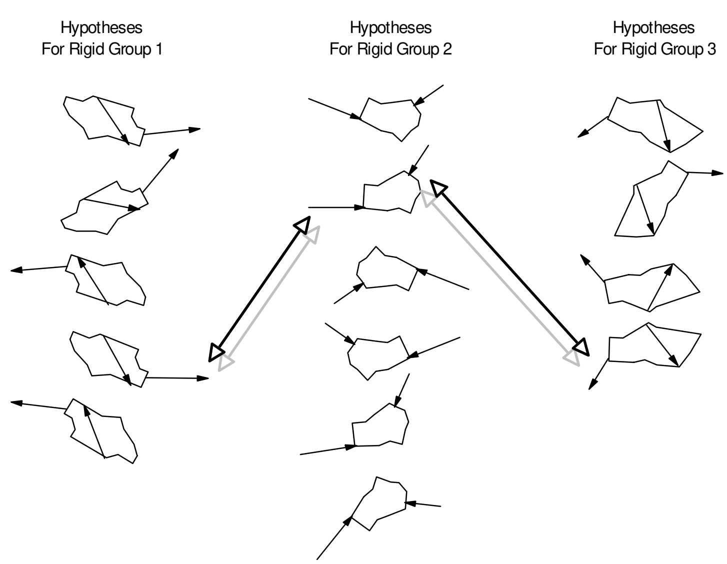


Figure 4.

of its magic vectors. It should now be clear that any structural match between the query and rigid group 20 can be extended to *also* include any existing structural match between the query and rigid group 30 if there is a hypothesis (Molecule-Id, 20, Group's-Magic-Pair-Position-and-Orientation) as well as a hypothesis (Molecule-Id, 30, Group's-Magic-Pair-Position-and-Orientation) both placing the common magic vector (i.e. the rotatable bond 18b) in the same position and orientation in the three-dimensional space. Similar considerations are in place when attempting to extend a matching substructure to also include any existing structural match from the rigid group 10.

The situation is similar to what is depicted schematically in Figure 4. The figure shows hypotheses (Molecule-Id, Rigid-Group-Id, Group's-Magic-Pair-Position-and-Orientation) for a molecule such as the one in Figure 1 with three rigid groups. Each rigid group is depicted as an irregular closed contour. Also shown attached to each group's contour are the hypothesized positions and orientations of the vectors of the magic pair associated with the rigid group.[1] Hypotheses referring to two distinct groups that are connected through a rotatable bond are consistent with one another if they both agree on the three-dimensional position and orientation of the magic pair corresponding to the joining bond. And if a common substructure exists between the query molecule and the molecule under consideration that incorporates all the rigid groups of the latter, then a *path* such as the one shown in Figure 4 must exist. The formed path has the following properties:

- each hypothesis participating in the path corresponds to a position and an orientation of the respective group that is consistent with the position and orientation of any other neighboring group;
- hypotheses corresponding to two rigid groups that should be connected by a rotatable bond place the respective groups in the three-dimensional space in a way that agrees with the existence of the rotatable bond;
- the path itself corresponds to a *valid*, *novel conformation* of a database molecule that exhibits the maximum possible structural overlap with the query molecule;
- if for no possible conformation of the database molecule does a common substructure exist that spans all of the molecule's rigid groups, then no path will be complete: a partial path is a path that comprises hypotheses that do not include all of the rigid groups of the molecule.

Although we described the approach by making reference to the molecule of Figure 1, the statements hold for any database molecule as well.

Figure 4 also graphically shows how the magic pair is selected and fixed for each group. In particular, and as mentioned previously, the hypotheses for the rigid groups 1 and 3 involve magic vectors that are *interior* to the groups. Unlike the rigid group 20 of the molecule in Figure 1 which has two (non-colinear) rotatable bonds emanating from it, the rigid groups 10 and 30 have only one bond. For such groups, the second vector of the magic pair will be defined using atoms other than those participating in the rotatable bond: any vector that is not collinear with the one rotatable bond emanating from the unit will suffice. In the Appendix we elaborate on the issue of defining magic vectors.

4. The Algorithm

In this section we recapitulate the steps that we described in the discussion so far. The steps are given in a flowchart mode similar to the description in [40].

First we begin with the preprocessing stage during which we build the lookup table to be used. We assume the availability of a database of molecules that are, in general, torsionally flexible and for which we will be implementing a system able to carry out the operations outlined on page 2.

The *off-line* preprocessing stage populates the lookup table using the properties of the database molecules. Figure 5 shows a flowchart of the stage which begins with the selection of a molecule from the database. We assume that, if deemed necessary, an educated selection of the unique conformer to be used has already taken place; recall that any conformer from the conformational space of the molecule under consideration is equally good to use at this stage. The various rigid groups of the molecule are then identified together with the rotatable bonds emanating from each one of them.

From the identified set of the molecule's rigid groups one is chosen, and, one (or more) magic pair is associated with the rigid group. For the selected rigid group, we next generate the reference-tuple selection set. This is the set of all atomic sites that will participate in triplets generating the various indices; group augmentation with the incorporation of atoms belonging to other groups and participating in rotatable bonds that emanate from the group under consideration also takes place at this phase. It should be noted that occasionally one may wish to form higher-order tuples (e.g. quintuples) of atomic sites from which an index will be generated; for example, we recently devised a new invariant which operates on quintuples of points in three dimensional space and spreads all the produced lookup table entries evenly in a three-dimensional lookup table [47]. Other schemes are also

[1] It should be reiterated at this point that, although the position and orientations choices for each magic pair are shown as a two-dimensional drawing, in reality they are three-dimensional entities. Figure 4 is only meant as a visualization aid.

possible. But for the sake of clarity, the rest of the discussion will assume that the indices are based on three atoms.

Given the reference-tuple selection set, a number of non-collinear triplets can be formed by sub-selecting from this set. Since each such triplet is used as the core for a reference frame in which the magic pair's position and orientation will be expressed, we refer to them as the reference-frame tuple. The method proceeds by selecting one reference-frame tuple and building the respective (skewed) local reference frame. Subsequently, the magic pair associated with the rigid group is represented in the local reference frame; the representation is in terms of a

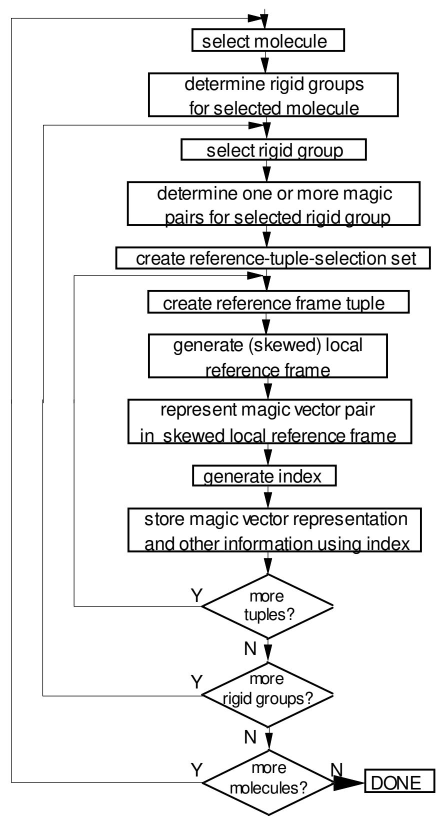


Figure 5.

translation and rotation matrix (see also previous discussion).

For the reference-frame tuple under consideration an index is generated (see also Appendix 1) and an entry made in the respective hash bin. The entry will contain the following information: the identities of the molecule and rigid group under consideration, identifiers for the atoms forming the reference-frame tuple, the representation of the magic pair(s) in the local reference frame, and possibly other information.

The above process will repeat as is necessary until all of the reference-frame tuples, rigid groups and database molecules have been considered. Upon termination, the lookup table contains all of the rigid groups and molecules *classified* according to the attributes of the tuple used to determine the hashing index.

In Figure 6, we are showing the steps for the *on-line* matching/derivation stage of the method. This stage assumes the availability of the lookup table generated during preprocessing. Matching/derivation begins with the selection of one of the query molecules. Although it is not necessary, it may frequently be beneficial nonetheless to determine any rotatable bonds that are present in the query molecule and the respective rigid groups.

Independent of whether this step takes place, one (entire molecule) or more (the various rigid groups) collections of atoms will be identified in the query molecule.

The method proceeds with the selection of one of the identified groups, and the formation of the matching-tuple selection set for the selected group. Similarly to the reference-tuple selection set, the matching-tuple selection set

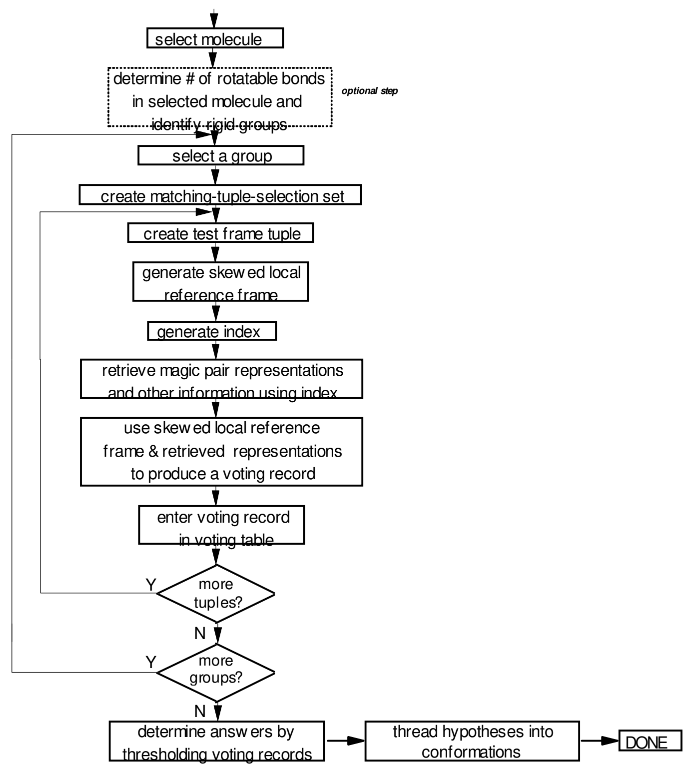


Figure 6.

includes all of the atoms that can participate in the creation of triplets (or higher-order tuples if appropriate). A tuple is selected and the respective (skewed) local coordinate frame is generated. Using the properties of the selected tuple an index into the lookup table is produced in a manner similar to that used during the preprocessing phase. The index will identify a hash bin containing multiple entries; for each of the found entries, the representation of the magic pair(s) contained therein is used in conjunction with the active local reference frame to produce an entry in a voting table. The entry is a hypothesis about the position and orientation (in the *global* coordinate frame[1]) of a rigid group of some molecule from the database (see also Figure 4).

The previous steps repeat until all the test frame tuples in all of the identified groups of the query molecule have been exhausted.

Upon termination, the voting table is likely to contain many entries with a multiplicity higher than 1. These entries will correspond to more than one frame tuples that corroborate a certain placement in the global coordinate frame of a certain molecule's rigid group. The multiplicity of occurrence reflects the extent of similarity (i.e. the size of the common substructure) between the query molecule and the named rigid group of the named molecule. A simple histograming operation of the voting table's contents will determine the identities of the rigid groups and respective molecules in order of decreasing support; attached to each such answer is the

[1] This can be easily achieved using the knowledge of the position in the global coordinate frame of each atomic site of the query molecule that participates in the test frame tuple.

position and orientation of the group's magic pair that will place the group in the best overlapping registration with the query molecule.

At this point, we have answered the question of which database molecules' rigid groups have common substructures with the query molecule. For some molecules of the database it is likely that there exist more than one rigid groups sharing substructures with the query molecule. In such situations, we can proceed even further and identify the conformation of the database molecule which is in agreement with the individual results produced by the histograming. This deduction is facilitated by the availability of the magic pairs' positions and orientations.

Let us assume that database molecule with identifier *m* is listed as having more than one rigid groups sharing substructures with the query molecule. We sub-select from the histogram those answers that involve molecule *m* and partition them into categories: the *i*-th category contains the answers corresponding to the *i*-th rigid group of *m* (see also Figure 4). These answers can now be seen as hypotheses for the purpose of producing a conformation for the molecule *m* which will exhibit the maximum possible overlap with the query molecule.

Since molecule *m* is one of the molecules of our database we have access to information about its topology and thus know which rigid groups are connected with one another. A hypothesis from the *i*-th category can be augmented by joining it with a hypothesis from the *j*-th category if and only if (a) the *i*-th and *j*-th rigid group of molecule *m* are connected, and, (b) both hypotheses agree on the position and orientation of the rotatable bond that they share. Joining these two hypotheses will "consume" the magic vector that corresponds to the common rotatable bond and will also form a conformation {i,j} of length 1. A conformation {i,j} can be further augmented by attaching a not-already-consumed magic vector from, say, its *i* end to a hypothesis from the *k*-th category; this augmentation is possible if and only if (a) the *i*-th and *k*-th rigid group are connected ,and, (b) the selected magic vector from the existing conformation and the hypothesis from the *k*-th group agree on the position and orientation of the common rotatable bond. The common magic vector is subsequently consumed and the current conformation {i,j,k} has now length 2. This procedure continues until a conformation incorporates consistent hypotheses from *all* the rigid groups of molecule *m*, or it cannot be augmented anymore. Notice, that the produced conformation may be branching at those nodes that correspond to rigid groups with more than two rotatable bonds emanating from them. The issue of conformation augmentation is pursued further in Appendix 4. It should be stressed that, once a maximal set {i,j,...} is built as above, generating the final (maximal overlap) conformation for molecule *m* is straightforward: indeed, positioning each of the molecule's rigid groups so that the respective magic pairs have the position and orientation by which they are listed in the maximal set is sufficient.

Note that at the end of the threading step we have just described, the algorithm produces a *novel* conformer for the molecule *m*[1] with the property that it maximizes the atomic overlap with the query molecule, and all of this in the presence of molecule *m*'s torsional flexibility.

The threading procedure described above can be repeated for all the database molecules appearing in the histogram of the voting table's entries with more than one rigid group in order to their conformers that will exhibit maximum atomic overlap with the query molecule.

5. Experimental Results

In this section, we present preliminary results from a prototype parallel implementation of the method we have described.

The database used was the publicly available segment of the National Cancer Institute's database of carcinogenic substances [34]. After excluding those of the entries which (a) corresponded to disconnected molecules, and (b)

---

[1] Indeed, the derived conformer will in general be different than the one used to produce the lookup table entries for *m*.

had no three-dimensional information listed, we generated a database containing 123,065 molecules. The table below shows the statistics in terms of the number of rotatable bonds present in a database molecule. As can be seen, a small number of these molecules were rigid, whereas most contained at least a handful of rotatable bonds.

Rotatable Bonds	0	1	2	3	4	5	6	7	8	9	10	11	≥12
# of Molecules	11,865	13,341	17,791	17,569	16,557	11,637	9,502	5,994	4,906	3,154	2,500	1,821	6,428

A lookup table was produced for this database using the above described method. The size of the lookup table was approximately 3.5 Gbytes and the table was shared equally among the 8 processors of an IBM SP/2 super-computer. In this particular implementation, triplets of atoms were used to form tuples.

From each triplet we generated a quadruplet of numbers: three of the numbers corresponded to properties of the triangle formed by the selected atoms (see Apppendix 1), whereas the fourth number corresponded to the chemical type of one of the atoms; in this first prototype, we employed the 41 chemical types used in Tripos' SYBYL system [51]. The produced quadruplet of numbers was subsequently quantized and used to generate the index of a hash bin where an entry was made. During the selection of triplets, we only allowed triplets whose respective

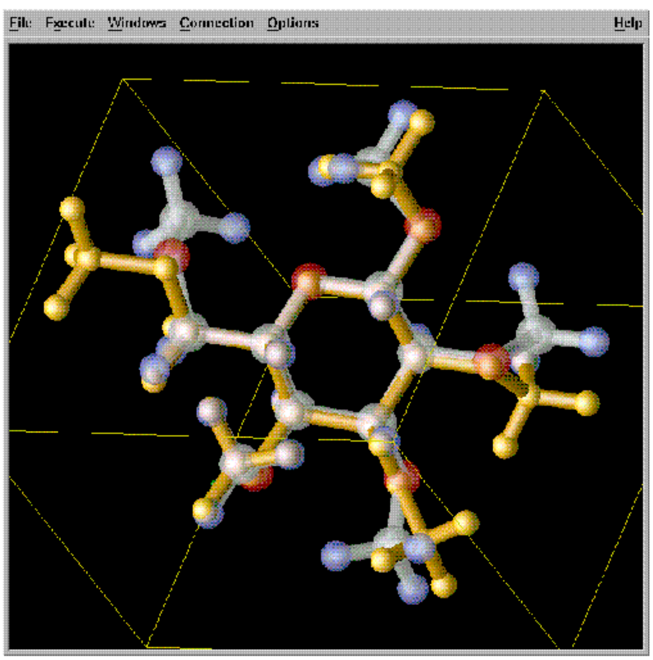


Figure 7.

triangles' sides had lengths in the range $[0.9, 5.0]\,\AA$. We made no attempts to optimize the lookup table generation phase which required approximately 10 hours on the 8-processor SP/2.

For the matching/derivation phase, we used molecules of varying sizes. The search times showed little variation and were in the order of 5 minutes for a query against all 123,065 molecules of the NCI database. Here, we show some representative results. Figure 7 shows a match that extends over several rigid groups.

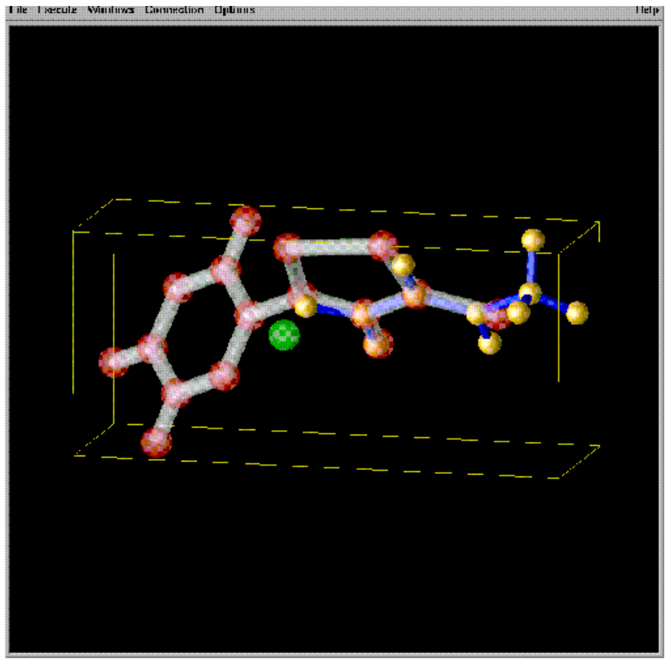


Figure 8.

In Figure 8 we show one of the partial matches that the method can identify. The query molecule was the linear molecule shown as solid. The database molecule is depicted semi-transparent so that the existing overlap is easily visible. The point to be made here is that the method is powerful enough to solve the described problem in an efficient manner but without compromising the sensitivity needed to identify weak matches.

The reported results should be viewed as preliminary. An exhaustive series of tests is necessary to determine the capabilities and limitations of the approach and is currently underway.

6. Discussion

In this work, we introduced and discussed a new method for carrying out flexible 3D substructure matching and conformer derivation in three-dimensional databases of molecules that exhibit torsional flexibility around their rotatable bonds.

The method we have presented comprises two-phases. The first of the two phases takes place *off-line* and a lookup table is built that contains information pertaining to the set of molecules to be recognized. Despite the fact that the molecules under consideration exhibit torsional flexibility, and thus can assume an infinity of possible conformations, information about *only one* of those conformations needs to be stored in the lookup table. In fact, there is complete freedom in the choice of which conformation can be used during this phase. Since only one conformation needs to be used to produce entries in the lookup table, the method also makes an optimal use of disk storage.

During this first phase, we store in the lookup table implicit information about a molecule's ability to assume new conformations via rotations around its rotatable bonds. We carry this step at the level of each rigid group of the molecule rigid group. A careful selection of what exactly is stored in the lookup table will guarantee optimal storage requirements and minimal computational requirement during the second of the phases.

The implicit information that we store during the off-line phase is exploited during the second phase which takes place *on-line*. In a manner, reminiscent of the pose clustering technique, we generate and cluster hypotheses that correspond to placements in the three-dimensional space for the various rigid groups of the database molecules.

Since these hypotheses are generated with the help of a lookup table: (a) they are produced fast, and (b) a substantial *filtering* takes place that rejects irrelevant hypotheses. A thresholding operation can further reduce the number of rigid group placement hypotheses.

At this point, and as a consequence of the information that was originally stored in the lookup table and later reconstructed during evidence accumulation, additional three-dimensional constraints are available; the latter dictate which of the hypotheses can be potentially joined to produce larger sets of database substructures common with the query molecule.

What is most important is that this augmentation step can take place in time that is a very small fraction of the time that would otherwise be needed to sample a molecule's conformational space. We wish to stress that our method does not require the generation of any conformers for a molecule-answer and as such it is not dependent on any conformer-generation or conformer-sampling strategies. Instead, in our approach we reduce the task at hand to a set of consistency checks (= augmentation step) for already available, valid hypotheses and it consequently is very efficient computationally.

The end result of this augmentation step is a derived conformer for the molecule-answer. This conformer is produced in one step and is not the result of any optimization approach as in other methods. Moreover, the derived conformer suffers from *no torsional-resolution* limitations: methods that rely on either explicit or implicit sampling of the conformation space, using intervals which follow a deterministic schedule, in essence impose a quantization step. As a result, the resulting discrimination power of any such method is quantized in the sense that it collapses more that one valid conformations into one representative. To contrast this, our method's resolution ability is only limited by machine precision.

From the description of the algorithm, it should be clear that the threading stage is at liberty to derive conformations that are not at or near a minimum energy configuration; this is a desirable property especially if the query molecule is taken from the context of a receptor and is in its bound form.

Another interesting aspect of our algorithm is that is becomes more efficient as the number of a molecule's rotatable bonds increases! This is a counterintuitive result since increase of the number of torsionally flexible bonds increases the size of the conformation space for the molecule. It is easy to see that for a fixed number of atoms in a molecule, an increase in the number of rotatable bonds will result in a decrease of the number of this molecule's entries in the lookup table. This consequently increases the efficiency of the hypothesis generation step. Moreover, since fewer hypotheses will be generated for the molecule in question, the threading of the rigid group hypotheses into molecular conformations will proceed much faster.

And last but not least, the described algorithm is characterized by superb scalability properties. Doubling of the available processing power has been shown to more than double the performance of the algorithm. In an analogous manner, doubling the size of the database produced a slowdown by a factor of 2. We have experimented and verified these observations using databases with sizes that ranged from a few hundred molecules to the size of the publicly available part of NCI's database (over 123,000 molecules).

We have found that the described method is particularly suitable for large databases of small molecules (drug molecules) and will also be valid for all molecules where the exhibited flexibility can be captured by the torsionally flexible bond model that we have assumed throughout our analysis.

By building the prototype system above we have also identified areas where further exploration is warranted. One such area is related to the observed non-uniform occupancy of the various hash bins. This is a direct consequence of the indexing scheme that is used (see also Appendix) and imposes limitations in the scalability of the system. In this direction, we have already devised a new hashing function exhibiting all desirable properties [47] and are in the process of incorporating it in the prototype.

A second area relates to the used atom typing scheme. We are currently exploring alternative ways of atom typing (different than SYBYL's 41 atom types) and their impact on the system's behavior.

A third area relates to the presence of clusters of atoms that exhibit internal symmetries (e.g. ring systems, methylene arrangements). These structures impose an unnecessary computational burden during the threading of the group-placement hypotheses into molecular conformations that can be alleviated by additional bookkeeping during preprocessing. We are at the moment evaluating several approaches that are expected to further improve the system's performance while at the same time decreasing storage requirements.

Another area concerns the used scoring function. At the moment, the scoring function is a measure of the extent of the atomic overlap between the query molecule and each proposed answer; it does not account for electrostatic and/or van der Waals interaction. We are in the process of investigating various enhancements that incorporate such measurements.

We conclude by pointing out that the algorithm can be modified in a straightforward manner to address the pharmacophore problem and work toward this end is currently under way.

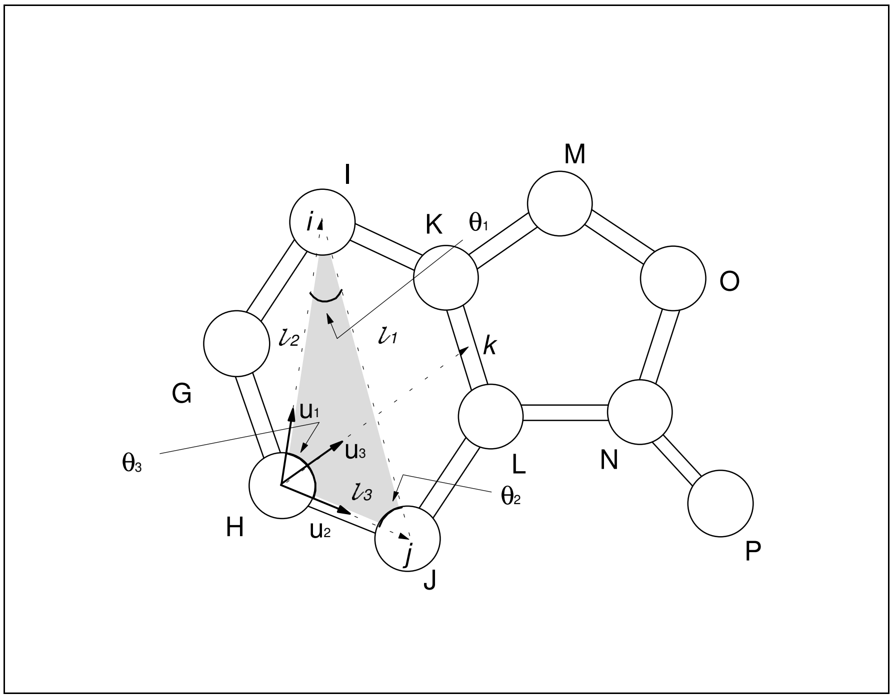
7. Acknowledgments

We would like to thank David Silverman and Mike Pitman for providing a stimulating environment where this system could be developed, and also for providing a first database on which various concepts were tried. We would also like to thank Haim Wolfson, Ruth Nussinov and Dani Fischer for numerous conversations over the years. Finally, our thanks go to the National Cancer Institute in Bethesda, Maryland for making available the public segment of their database without which the development of this system would not have progressed equally fast.

Appendix 1

Issues In Index Generation

An issue for which we gave a high-level description in the above presentation is that of the index generation. Given a triplet of atoms in a molecule we need to generate an index that captures various characteristics of the triplet. The index is used to access a location of the lookup table both during preprocessing and also matching/derivation. As already stated the index may capture a lot of characteristics pertaining to the triplet itself as well as the respective molecule.



For the purposes of the method we have described a minimum requirement for the set of characteristics would be the inclusion of the geometric properties of the triplet. Let us refer to the above figure: we assume that the triplet of atoms under consideration is H-I-J. For simplicity, we also assume that the triplet is ordered. The triplet of atoms forms a triangle (shaded region) with the following properties: the three sides of the triangle have lengths ℓ_1, ℓ_2 and ℓ_3 whereas the angles at the three vertices are equal to ∂_1, ∂_2 and ∂_3 respectively. By appropriately quantizing the values of any subset of these properties we can produce an index that is particular to the triplet under consideration. One subset that we found worked well when we experimented with the public part of the National Cancer Institute's database (123,065 molecules) was ℓ_1, ℓ_2 and ∂_1.

It should be pointed out that these 6 numbers are correlated; any subset of 2 or more will produce indices exhibiting preference for certain regions of the range of indices. In other words, the probability of making an entry in a given hash bin depends on the bin and is *not* constant for all bins. This in turn translates into lookup tables with bins containing greatly varying numbers of entries. The observation is not peculiar to the above way of producing indices but instead reflects a property endemic to all indexing schemes. In fact, it has been proven to hold true for a variety of data types and problem domains.

This preference for certain regions of the range of the hashing function has important ramifications for systems that manipulate very large databases. Indeed, and for a given size of the lookup table, uneven occupancy of the various hash bins translates into a diminished net discrimination ability. Additionally, sharing of the involved very large databases among several hard drives incurs additional workload to some of the storage devices thus affecting performance. And finally, in parallel processing environments, an unevenly distributed set of entries in the table's hash bins directly impacts the load sharing properties among the processors.

This problem of non-uniform distribution of the entries across the range of possible index values can be alleviated in a number of ways. First, one can attempt to form indices by selecting among the available variables those that are as little correlated as possible.

Second, a Huffman encoding approach based on the frequency of appearance of the various indices can be employed. Third, a rehashing function can be computed using the frequency of appearance of the various indices (in essence one computes the cumulative distribution corresponding to the probability density function of the hashing function that corresponds to the indexing scheme in effect) [45].

Finally, a fourth method for coping with the problem involves the design of alternative hashing functions with improved hash bin occupancy characteristics. Along these lines, we recently designed and implemented a new invariant producing function with the following properties:

- when presented with the positions of a five-tuple of points in three-dimensions, the function generates a triplet of real numbers (u, v, w);
- each of the u, v, and w assumes values in the interval $[0, 1]$;
- the expected frequency of occurrence of a given triplet is constant and independent of the values of u, v, and w.

In essence, the function can be used to implement a three-dimensional lookup table with an expected constant occupancy frequency across the entire set of bins. More details on this work can be found in [47].

Although we concentrated only on the geometrical characteristics of a set of atoms, it is clear that additional information can also be used to form the index. This information may be capturing other properties (e.g. physical, chemical etc.) pertaining to the atoms forming the tuple and/or the molecule to which they belong. For example, a value can be attached to each of the three atoms with labels H, I and J which corresponds to their chemical type; this value can be used as an additional element of information when forming the index of a hash bin.

It should be cleat that incorporating more information in the index generation process makes the index more descriptive than before. However, and depending on the nature of the problem at hand, this increase in discrimination power may occur at the cost of the resulting sensitivity; thus, caution should be used when trying to strike a balance between these two requirements.

Appendix 2

Defining Magic Vector Pairs

In this appendix, we discuss the concept of magic vectors in more detail. In particular, we present various situations and provide suggestions for how to handle them.

The simplest situation is that of a rigid molecule which contains no rotatable bonds. The distance between any two atoms of the molecule remains constant under any possible rotations and translations of the molecule in three-dimensional space. In this case, the magic pair can be defined using any two non-collinear vectors that are defined using pairs of atoms belonging to the molecule. For example, one magic vector could be defined with the help of the two atomic sites that are the furthest apart in the molecule; the second magic vector could be the one defined using one of the already selected atomic sites and the atomic site that is the furthest from the axis of the first magic vector. In the case of ties, the labels of the atomic sites (and the implied lexicographic order) can be used.

When the molecule under consideration is not rigid, we have seen that the method calls for one magic pair for each of the rigid groups of the molecule. For rigid groups that have two rotatable bonds emanating from them, the magic pair will comprise the two vectors that these bonds define. The labels of the respective atomic sites can be used to determine a consistent direction for the two vectors.

If the rigid group has only one rotatable bond emanating from it (i.e. an "end-group") then the magic pair will comprise the vector that corresponds to the one rotatable bond and one non-collinear vector defined using any two atoms from the group. Referring again to Figure 1, the additional vector for group 10 could for example be $\overrightarrow{AE}$, whereas the additional vector for group 30 could be $\overrightarrow{RS}$.

Frequently, a rigid group may have more than two rotatable bonds emanating from it. We frequently refer to such rigid groups as "star-groups" or "star-units." Any two non-collinear bonds can serve as the basis for defining the magic pair needed to position and orient the group. For the purposes of generating the path shown in Figure 4 additional bookkeeping will be necessary, of course. Indeed, once the hypotheses for the star-unit in question are available, the positions and orientations of the rotatable bonds not participating in the magic pair will need to be computed dynamically in order to determine whether the current path can be further extended by adjoining to it consistent hypotheses for rigid groups that are *accessible* through these latter bonds. Occasionally, it may be desirable to associate more than one magic pair to a star unit; for example, for a star unit with N rotatable bonds, one choice could be a properly selected subset from the $\binom{N}{2}$ possible pairs.

There is always the possibility that a rigid group has exactly two collinear (or almost collinear) rotatable bonds emanating from it. As per our method, the respective magic pair would comprise both of these bonds and thus be degenerate. Although the collinearity condition allows us to recover the translation matrix it prevents the computation of the rotation matrix that would put the rigid group in registration with the query molecule.

Two are the relevant issues here: the *identification* of the appropriate database model and the *alignment* of the two molecules. It is clear that identification could be carried out even in this degenerate case. Indeed, extending an existing path (see Figure 4) by adjoining to it a consistent hypothesis of another rigid group requires the position and orientation of the rotatable bond over which we wish to extend the path.

On the other hand, in order to be able to perform recognition (see previous discussion) we need to align pairs of rigid groups and for this a rotation matrix is required. What is needed is the ability to recover the appropriate rotation matrix that will correctly position the rigid group under consideration. A simple solution to this problem involves replacing *every* degenerate magic pair by *two* new magic pairs. This is in essence a hybrid of the modification that is needed to handle end-units and calls for the definition of a third vector that is not collinear with either of the two original magic vectors. This third vector can be defined with the help of a direction that is of relevance to the group under consideration. Once the third vector is selected, the original magic pair is replaced by the two magic pairs formed by the third vector and each of the original magic vectors. Clearly, additional bookkeeping is necessary in this case.

A final comment concerns the situation where the rigid group contains only one atom. For example, this could be the case of a *Y-X-Y arrangement where the rigid group in the middle contains one atom only (X) --- the dashes correspond to rotatable bonds. The method we have outlined calls for the association of this single-atom group with a magic pair.

If the two rotatable bonds connecting X to its two neighbors are not collinear, then the rigid group can be augmented to also include its two neighbors: the magic pair could then be defined with the help of the two non-collinear rotatable bonds. The augmented group has now three atoms and can produce lookup table entries during the preprocessing and matching/derivation phases.

If on the other hand, the two bonds are collinear, the situation needs to be treated differently: the single atom X is then assigned to either of its two neighboring groups and is not in a group by itself anymore.

Appendix 3

Assigning Weights To Hypothesis Contributions

Throughout the discussion we implicitly assumed that each of the contributions to a given hypothesis was assigned the same (constant) weight. However, deviations from this framework may be necessary.

According to the description of the algorithm, every query triplet is associated, with the help of the lookup table, with a triplet of a database molecule. For each such association a hypothesis is generated. When all of the produced hypotheses become available, the method will cluster together those hypotheses which are consistent when placing a rigid group's magic pair (and thus the rigid group). Our suggestion is to assign a *weight* to each generated hypothesis; the weight is carried along through the clustering phase where it is used to compute the total support for each of the distinct hypotheses.

The manner in which weights are assigned is a direct consequence of our treating individual rigid groups as molecules in their own right. If each component hypothesis was given the same weight (e.g. 1), then it is clear that hypotheses corresponding to large rigid groups will have a large final support. Indeed, a large rigid group will give rise to a large number of generated triplets and stored lookup table entries; all these entries will generate consistent placement hypotheses and thus the final hypothesis they corroborate will receive a lot of support. The opposite will be true for a small group.

The following suggestion assigns weights to each lookup table entry that are to be used to tag the generated hypotheses. As already explained, the scheme is designed so that it gives more credit to entries corresponding to small rigid groups and less credit to entries corresponding to large rigid groups. Let e_{ij} be the number of lookup table entries that the i-th rigid group of the j-th database molecule will produce. The j-th molecule is assumed to consist of n_j rigid groups. Let also m_{ij} be the number of magic pairs that are associated with this rigid group. Let w_{ij} denote the sought value for the weight to be assigned to each of the e_{ij} entries. We first compute a constant B as

$$B = \max_{i,j} \{e_{ij} \cdot n_j \cdot \max \{m_{ij} - 1, 1\}$$

If we select a value M that satisfies the inequality $M \geq B$ then we can define the weights as

$$w_{ij} = M / \{e_{ij} \cdot \max \{m_{ij} - 1, 1\}\}$$

With the above scheme, if a query molecule contains a substructure matching a database molecule's rigid group in its entirety, then the final hypothesis for the position and orientation of this group's magic pair will be corroborated by e_{ij} associations with total support equal to $M / \max\{m_{ij} - 1, 1\}$.

As was evidenced by our implementation of the method, the above suggested scheme performs very well with a very large and diverse database such as the public component of the National Cancer Institute's database of carcinogens [34]. A number of variations on the above weighting scheme are also possible and are discussed in work in preparation.

Appendix 4

Hypothesis Threading / Conformer Derivation

Let us recall again the situation that exists immediately after the histograming of the entries in the voting table has completed. For the purpose of describing the method, in what follows we will concentrate only on those of the entries that belong to molecule m of the database.

We will assume that there is a total of H entries listing positions and orientations for the magic pairs of the various rigid groups of molecule m. If molecule m comprises G rigid groups and no group in particular is any more *important* than any other, we will assume without loss of generality that there is H/G many entries corresponding to the i-th rigid group, ($i=1,...,G$) of the molecule.

We next divide the H entries into G many categories. The i-th category contains only those entries that pertain to the i-th rigid group (see also Figure 4). Each entry within a category corresponds to a placement of the rigid group in the three-dimensional space which will *reveal* the substructure that is common between the rigid group under consideration and the query molecule. At the same time, this same entry can be viewed as a hypothesis for the placement of the rigid group, in a potential conformation of the entire molecule m, the conformation exhibiting the maximum possible overlap with the query molecule.

Let $<i,\boldsymbol{p}_{1k},\boldsymbol{v}_{1k},\boldsymbol{p}_{1k},\boldsymbol{v}_{2k},w_{ik}>$ denote an entry from the i-th category. The vectors $\boldsymbol{p}_{1k}$ and $\boldsymbol{v}_{1k}$ represent the hypothesized position and orientation of the *first* magic vector attached to the i-th rigid group; similarly for the vectors $\boldsymbol{p}_{2k}$ and $\boldsymbol{v}_{2k}$. Since we have already sub-selected only the entries for molecule m, there is no need for a molecule identifier. Optionally, a weight w_{ik} may be attached to the hypothesis, in essence indicating the amount of support for this hypothesis (see also relevant Appendix discussion). Note that we can build two lists of pointers, one for each magic vector, that will maintain each category's entries sorted according to the properties of either magic vector; the cost of this operation will be no more than O (H/G log(H/G)), per category.

Finally, in addition to the above information we also have access to the topology for model m and thus know which rigid groups are connected with one another. Equivalently, given a hypothesis $<i,\boldsymbol{p}_{1k},\boldsymbol{v}_{1k},\boldsymbol{p}_{1k},\boldsymbol{v}_{2k},w_{ik}>$ from category i the topology dictates which category we are allowed to search for matching placement hypotheses.

The task at hand is to combine consistent hypotheses from each rigid group in a manner that is also consistent with the molecule's topology. The end result corresponds to a valid conformation of the entire molecule m that will place it in the best possible overlap with the query molecule and ideally would include one hypothesis from each of the G groups. The end result will have a total support that will be the sum of the support for each of the component hypotheses. Combinations of hypotheses that will lead to the largest possible value for the total support are most preferable.

We begin by selecting one hypothesis from say the i-th category. Let the topology indicate that the i-th rigid group is connected via a rotatable bond to the j-th rigid group. Without loss of generality let us assume that it is the second magic vector from $<i,\boldsymbol{p}_{1k},\boldsymbol{v}_{1k},\boldsymbol{p}_{1k},\boldsymbol{v}_{2k},w_{ik}>$ that corresponds to the rotatable bond connecting the i and j rigid group. A conformation initiation can occur if and only if we can find a hypothesis $<j,\boldsymbol{p}_{1m},\boldsymbol{v}_{1m},\boldsymbol{p}_{1m},\boldsymbol{v}_{2m},w_{jm}>$ in the j-th category that agrees with the one under consideration on the position and orientation of the common magic vector. If the first magic vector from the second entry corresponds to the common rotatable bond, it must hold that $\boldsymbol{p}_{2k}=\boldsymbol{p}_{1m}$ and $\boldsymbol{v}_{2k}=\boldsymbol{v}_{1m}$. Joining these two hypotheses into an {i,j} conformation will "consume" this common magic vector. Those of the remaining magic vectors which hypothesize placement of remaining rotatable bonds are considered available for a new join operation. The {i,j} conformation has a total support equal to $w_{ik}+w_{jm}$.

An {i,j} conformation can subsequently be augmented by attaching one of its available magic vectors to a hypothesis $<l,\boldsymbol{p}_{1n},\boldsymbol{v}_{1n},\boldsymbol{p}_{1n},\boldsymbol{v}_{2n},w_{ln}>$ from the l-th category if and only if (a) the group to which the available magic vector corresponds and l-th rigid group share a rotatable bond (this is resolved by the available topological information) and, (b) the selected available magic vector and the hypothesis for the l-th group agree on the position and orientation of the common rotatable bond. The common magic vector is subsequently consumed and the conformation is augmented to an {i,j,l} conformation with total support $w_{ik}+w_{jm}+w_{ln}$. Conformation augmentation proceeds in a recursive manner until it includes one hypothesis from each category, or no further augmentation can take place.

The final {i,j,l,...,p} conformation is built using a collection of hypotheses that correspond to placements of the respective groups that are consistent with one another and with the topology of molecule *m.* The total support for the final conformation will be equal to the sum of the weights that are attached to each of the hypotheses included in the conformation. When each of the named rigid groups is placed in three-dimensional space as the respective hypothesis dictates, the entire molecule will be in the conformation that will place it in the best possible overlap with the query molecule.

Last, we will examine the complexity of this last operation. For this we need to introduce the concept of the *hypothesis fan-out*: this is the number of hypotheses which can be found in a category to be considered for a join operation and are consistent with the magic vector to be consumed. The fan-out in essence is a measure of how many hypotheses in another group can be expected to match the position and orientation of a given available magic vector. Again, and without loss of generality we assume that this number is constant and equal to *d.*

Working in the context of the example for molecule *m*, we can derive an estimate of the cost for producing the answer with the largest value of total support. The worst case scenario is the one where the molecule has no star units, and this is what we will assume for *m* here. Given a starting hypothesis, and since the entries of the various categories are assumed sorted, we can identify the *d* matching hypotheses that can allow conformation initiation at a cost of no more than $(d + \log(H/G))$ steps. For each one of these *d* matching hypotheses there is a fan-out of *d* and it will cost us no more than $(d(d + \log(H/G)))$ steps to consider all of them. At the next level, we will have d^2 hypotheses to consider each with a fan-out of *d* again costing us $(d^2(d + \log(H/G)))$. At the level prior to the last we will have d^{G-2} hypotheses to consider each with a fan-out of *d.* All in all in, and for the starting hypothesis, we can determine and build the list of consistent hypotheses with the maximum possible support in approximately $T = \frac{d^{G-1}}{d-1}\left(\log\left(\frac{H}{G}\right) + d\right)$ steps. Finally, we can reasonably consider the fan-out to be equal to a fraction of the number of entries in a given category, say $d = \left(\frac{H}{kG}\right)$ with $k > 1$.

Setting H=300, G=6, and k=10 in this last formula we have that the order of magnitude for T is 10^4, a result that reflects a substantially smaller computational load than what a fixed tesselation of a molecule's conformational space would generate. In reality, the situation is a little better than what the above result indicates and this is further corroborated by our experimental results with NCI's large-size database.

Concluding this discussion, we wish to point out that if the rigid group under consideration has a symmetry (e.g. a planar six-member ring) then additional computational burden is imposed. We are currently extending our approach to handle this situation in a computationally cheaper manner and our analysis and results will be presented in a future discussion.

References

[1] A. Aho, R. Sethi et al., *"Compilers: Principles, Techniques And Tools."* Addison-Wesley, Reading, MA. 1986.

[2] P. A. Bath, A. R. Poirette et al., "Similarity Searching In Files Of Three-Dimensional Chemical Structures: Comparison Of Fragment-Based Measures Of Shape Similarity." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 1241-1247, 1994.

[3] G. W. Bemis and I. Kuntz, "A Fast And Efficient Method For 2D and 3D Molecular Shape Description." In *Journal Of Computer-Aided Drug Molecular Design*, Vol. 6, pp. 607-628, 1992.

[4] R. D. Brown, G. M. Downs et al., "Hyperstructure Model For Chemical Structure Handling: Techniques For Substructure Searching." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 47-53, 1994.

[5] A. Califano and R. Mohan, "Multidimensional Indexing For Recognizing Visual Shapes." In *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 16, No. 4, pp. 373-392, 1994.

[6] A. Califano, and I. Rigoutsos, "FLASH: A Fast Lookup Algorithm For String Homology." In Proceedings Of Intelligent Systems For Molecular Biology, Bethesda, MD, pp. 56-64, 1993.

[7] Chem-X Chemical Design, 200 Rte. 17 South, Suite 120, Mahwah, NJ 07430.

[8] B. D. Christie, D. R. Henry et al. Database Structure And Searching In MACCS-3D." In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 653-664, 1990.

[9] D. E. Clark, G. Jones, et al., "Pharmacophoric Pattern Matching In Files Of Three-Dimensional Chemical Structures: Comparison Of Conformational-Searching Algorithms For Flexible Searching." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 197-206, 1994.

[10] Concord-3D, Tripos Associates, 1699 Hanley Road, St. Louis, Missouri, 63144.

[11] R. L. DesJarlais, R. P. Sheridan et al., "Docking Flexible Ligands To Macromolecular Receptors By Molecular Shape." In *Journal Of Medicinal Chemistry*, Vol. 29, 1986.

[12] J. E. Dubois, G. Carrier, et al., "DARC Topological Descriptors For Pattern Recognition In Molecular Database Management Systems And Design." In *Journal Of Chemical Information And Computer Sciences*, Vol. 31, No. 4, pp. 574-578, 1991.

[13] W. Fisanick, K. P. Cross et al., "Similarity Searching On CAS Registry Substances. 1: Global Molecular Property And Generic Atom Triangle Geometric Searching." In *Journal Of Chemical Information And Computer Sciences*, Vol. 32, No. 6, pp. 664-674, 1992.

[14] W. Fisanick, A. H. Lipkus et al., "Similarity Searching On CAS Registry Substances. 2: 2D Structural Similarity." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 130-140, 1994.

[15] D. Fischer, O. Bachar et al., "An Efficient Automated Computer Vision Based Technique For Detection Of Three-Dimensional Structural Motifs In Proteins." In *Journal Of Biomolecular Structure & Dynamics*, Vol. 9, No. 4, pp. 769-789, 1992.

[16] D. Fischer, R. Nussinov et al., "3D Substructure Matching In Protein Molecules." In Proceedings 3rd Symposium On Combinatorial Pattern Matching, pp. 133-147, May 1992.

[17] M. R. Garey and D. S. Johnson, *"Computers And Intractability: A Guide To The Theory Of NP-Completeness."* W. H. Freeman And Company, San Francisco, 1979.

[18] A. C. Good, T. J. A. Ewing et al., "New Molecular Shape Descriptors: Applications In Database Screening." In *Journal Of Computer-Aided Molecular Design*, Vol. 9, pp. 1-12, 1995.

[19] J. Greene, S. Kahn et al., "Chemical Function Queries For 3D Database Search." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 6, pp. 1297-1308, 1994.

[20] O. F. Guner, D. R. Henry et al., "Flexible Queries In 3D Searching. 2:Techniques In 3D Query Formulation." In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 557-563, 1990.

[21] K. S. Haraki, R. Sheridan et al., "Looking For Pharmacophores In 3D Databases: Does Conformational Searching Improve The Yield Of Actives?" In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 565-573, 1990.

[22] M. G. Hicks and C. Jochum, "Substructure Search Systems. 1: Performance Comparison Of The MACCS, DARC, HTSS, CAS Registry MVSSS, and S4 Substructure Search Systems." In *Journal Of Chemical Information And Computer Sciences*, Vol. 30, No. 2, pp. 191-199, 1990.

[23] C. Humblet and J. B. Dunbar, "Chapter 29: 3D Database Searching And Docking Strategies." In *Annual Reports In Medicinal Chemistry-28*, M. Venuti (Ed.), Academic Press, pp. 275-284, 1993.

[24] T. Hurst, "Flexible 3D Searching: The Directed Tweak Technique." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 190-196, 1994.
[25] S. K. Kearsley, D. J. Underwood et al., "Flexibases: A Way To Enhance The Use Of Molecular Docking Methods." In *Journal Of Computer-Aided Molecular Design*, Vol. 8, 1994, pp. 565-582.
[26] I. Kuntz, "Structure-Based Strategies For Drug Design And Discovery." In *Science*, Vol. 257, pp. 1078-1082, August 21, 1992.
[27] Y. Lamdan, J. Schwartz et al., "Affine-Invariant Model-Based Object Recognition." In *IEEE Transactions On Robotics And Automation*, Vol. 6, No. 5, pp. 577-589, 1990
[28] Y. Lamdan and H. Wolfson, "Geometric Hashing: A General And Efficient Model-Based Recognition Scheme.' In Proceedings International Conference On Computer Vision, pp. 237-249, 1988.
[29] A. R. Leach and I. Kuntz, "Conformational Analysis Of Flexible Ligands In Macromolecular Receptor Sites." In *Journal Of Computational Chemistry*, Vol. 13, No. 6, pp. 730-748., 1992.
[30] M. F. Lynch and P. Willett, "Chemical Structure Retrieval: A Review Of Current Research In The Department Of Information Studies, University Of Sheffield." In *Chimica Oggi*, Vol. 63, pp. 55-63, December 1990.
[31] Y. Martin, "3D Database Searching In Drug Design." In *Journal Of Medicinal Chemistry*, Vol. 35, No. 12, pp. 2145-2154, 1992.
[32] Y. Martin, M. G. Bures et al., "Searching Databases Of Three-Dimensional Structures." In *Reviews In Computational Chemistry*, K. Lipkowitz and D. Boyd (Eds.), VCH Publishers Inc., pp. 213-263, New York 1990.
[33] P. McHale, "Access, Integration And Interoperability: The MDL Approach To In-House 2D/3D Searching, Spreadsheets, And QSAR." In Proceedings Of the Chemical Information Conference, pp. 46-50, Annecy, France, 1993.
[34] G. W. A. Milne, M. C. Nicklaus et al., "National Cancer Institute Drug Information System 3D Database." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 5, 1994.
[35] M. D. Miller, S. K. Kearsley et al., "FLOG: A System To Select 'Quasi-Flexible' Ligands Complementary To A Receptor Of Known Three-Dimensional Structure." In *Journal Of Computer-Aided Molecular Design*, Vol. 8, 1994, pp. 153-174.
[36] T. E. Moock, D. R. Henry et al., "Conformational Searching In ISIS/3D Databases." In *Journal Of Chemical Information And Computer Sciences*, Vol. 34, No. 1, pp. 184-189, 1994.
[37] J. B. Moon and W. J. Howe, "3D Database Searching And De Novo Construction Methods In Molecular Design." In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 697-711, 1990.
[38] N. W. Murrall and E. K. Davies, "Conformational Freedom In 3D Databases. 1: Techniques." In *Journal Of Chemical Information And Computer Sciences*, Vol. 30, pp. 312-316, 1990.
[39] R. Nilakantan, N. Bauman et al., "New Method For Rapid Characterization Of Molecular Shapes: Applications In Drug Design." In *Journal Of Chemical Information And Computer Sciences*, Vol. 33, No. 1, pp. 79-85, 1993.
[40] R. Nussinov and H. Wolfson, "Efficient Detection Of Three-Dimensional Motifs In Biological Macromolecules By Computer Vision Techniques." In Proceedings National Academy Of Sciences (USA), Vol. 88, pp. 10495-10499, 1991.
[41] T. Okada and W. T. Wipke, "CLUSMOL: A System For The Conceptual Clustering Of Molecules." In *Tetrahedron Computer Methodology*, Vol. 2, No. 4, pp. 249-264, 1989.
[42] C. A. Peperell, R. Taylor et al., "Implementation And Use Of An Atom-Mapping Procedure For Similarity Searching In Databases Of 3D Chemical Structures." In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 575-593, 1990.
[43] E. M. Rasmussen, P. Willett et al., "Searching Of Chemical Structure Databases With Parallel Computer Hardware." In *Analytica Chimica Acta*, Vol. 235, No. 199, pp. 77-86, 1990.
[44] E. M. Ricketts, J. Bradshaw et al., "Comparison Of Conformations Of Small Molecule Structures From The Protein Data Bank With Those Generated By Concord, Cobra, ChemDBS-3D and Converter, And Those Extracted From The Cambridge Database." In *Journal Of Chemical Information And Computer Sciences*, Vol. 33, No. 6, pp. 905-925, 1993.
[45] I. Rigoutsos, "Massively Parallel Bayesian Object Recognition." Ph.D. Thesis, New York University, 1992.

[46] I. Rigoutsos, and A. Califano, “Searching In Parallel For Similar Strings.” In *IEEE Computational Science And Engineering*, Vol. 1, No. 2, pp. 60-75, Summer 1994.
[47] I. Rigoutsos, “Three-Dimensional Affine Invariant Hashing Defined Over Any Three-Dimensional Convex Domain And Producing Uniformly-Distributed Hash Keys.” US Patent Application #Y0996-045, Spring 1996. In preparation.
[48] I. Rigoutsos, “System And Method For Conformationally-Flexible Molecular Recognition.” US Patent Application #Y0995-229, December 1995.
[49] B. Sandak, R. Nussinov et al., “An Automated Computer Vision And Robotics-Based Technique For 3D Flexible Biomolecular Docking And Matching.” In *CABIOS*, Vol. 11, No. 1, pp. 87-99, 1995.
[50] G. Stockman, “Object Recognition And Localization Via Pose Clustering.” In *Computer Vision, Graphics and Image Processing*, Vol. 40, pp.361-387, 1987.
[51] SYBYL Release 6.01, Tripos Associates Inc., 1699 S. Hanley Rd., St. Louis, MO.
[52] J. R. Ullman, “An Algorithm For Subgraph Isomorphism.” In *Journal Of The Association For Computing Machinery*, Vol. 23, No. 1, pp. 31-42, January 1976.
[53] V. J. Van Geerestein, N. C. Perry et al., “3D Database Searching On The Basis Of Ligand Shape Using The SPERM Prototype Method.” In *Tetrahedron Computer Methodology*, Vol. 3, No. 6C, pp. 595-613, 1990.
[54] P. Willett, “Ranking And Clustering Of Chemical Structure Databases.” In *Physical Property Prediction In Organic Chemistry*. C. Jochum and M. Hicks and J. Sunkel (Eds.), Springer-Verlag, pp. 191-207, Heidelberg, 1988.
[55] P. Willett, “Processing Of Three-Dimensional Chemical Structure Information Using Graph-Theoretic Techniques.” In Proceedings Online Information 90, pp. 115-127, London, England, 1990.
[56] P. Willett, “Algorithms For The Calculation Of Similarity In Chemical Structure Databases.” In *Concepts And Applications Of Molecular Similarity*, M. Johnson and G. M. Maggiora (Eds.), John Wiley And Sons Inc., pp. 43-63, 1990.
[57] P. Willett, “A Review Of Chemical Structure Retrieval Systems.” In *Journal Of Chemometrics*, Vol. 1, pp. 139-155, 1987.
[58] W. T. Wipke and D. Rogers, “Tree-Structured Maximal Common Subgraph Searching. An Example of Parallel Computation With a Single Sequential Processor.” In *Tetrahedron Computer Methodology*, Vol. 2, No. 3, pp. 177-202, 1989.