Proc. of the 1996 ACM Workshop on Applied Computational Geometry
To appear in LNCS series, Springer-Verlag, D. Manocha and M. Lin Eds.

Geometric Manipulation of Flexible Ligands*

Paul W. Finn[1], Dan Halperin[2]**, Lydia E. Kavraki[2], Jean-Claude Latombe[2], Rajeev Motwani[2], Christian Shelton[2], Suresh Venkatasubramanian[2]

[1] Pfizer Central Research, Sandwich, Kent CT13 9NJ, UK
[2] Department of Computer Science, Stanford University, Stanford, CA 94305, USA

Abstract. In recent years an effort has been made to supplement traditional methods for drug discovery by computer-assisted "structure-based design." The structure-based approach involves (among other issues) reasoning about the geometry of drug molecules (or *ligands*) and about the different spatial conformations that these molecules can attain. This is a preliminary report on a set of tools that we are devising to assist the chemist in the drug design process. We describe our work on the following three topics: (i) geometric data structures for representing and manipulating molecules; (ii) conformational analysis—searching for low-energy conformations; and (iii) pharmacophore identification—searching for common features among different ligands that exhibit similar activity.

1 Introduction

Most existing pharmaceutical drugs were found either by chance observation or by screening a large number of natural and synthetic substances [7]. In recent years there has been a growing tendency to supplement the traditional methods of drug discovery by *structure-based design*. The structure-based approach builds on the improved understanding of the molecular interaction underlying diseases, and attempts to predict the structure of a potentially active compound. The prediction can then be used either to synthetically construct such a compound, or to narrow down the screening process of existing substances. Refer to [3] for a comprehensive survey and bibliography on the subject.

The fundamental assumption of structure-based drug design is that at the molecular level, the key event leading to the desired effect of the drug is the recognition and binding of a small molecule (the *ligand*) to a specific site on a target macromolecule (the *receptor*) [3]. It is further assumed that at the

* Work on this paper has been supported by a grant from Pfizer Central Research. Work on this paper by Rajeev Motwani has been supported by an Alfred P. Sloan Research Fellowship, an IBM Faculty Partnership Award, an ARO MURI Grant DAAH04-96-1-0007, and NSF Young Investigator Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Work reported in Section 2.2 has been supported by ARO MURI Grant DAAH04-96-1-0007.

** Author's current address: Department of Computer Science, Tel Aviv University, Tel Aviv 69978, ISRAEL binding site, the ligand must present steric and electrostatic complementarity to the receiving pocket.

Depending on whether the structure of the receptor is known or not, two types of structure-based approaches can be considered. If the structure of the binding site is known, the process then centers on finding (or devising) a ligand that will complement the binding site. Our work focuses on the case when the structure of the receptor is not known. In this case, a possible scheme is to start with a set of compounds whose structure is known, and which have been observed to exhibit some level of activity with the target receptor molecule. The goal then is to extract the common features of the active compounds. These common features constitute a *pharmacophore*. With a pharmacophore at hand, the chemist can look for other compounds having similar features, but which may be more potent than any of the given compounds, or have other desirable properties such as non-toxicity.

A major source of difficulty in structure-based drug design is the flexibility of molecules to attain various *conformations*, namely different spatial configurations of their atoms. Properly handling flexible ligands has been identified as a major challenge in this field [22].

We have chosen to concentrate on the following topics where algorithmic tools are needed to support the design process: (i) data structures for representing molecule geometry and molecular surfaces; (ii) conformational analysis — searching for low-energy conformations; and, (iii) pharmacophore identification. The first two components can be viewed as support tools for the pharmacophore identification part. Although our current effort focuses on tools to support the drug design process when the receptor structure is not known, some of the tools that we are developing can also be used as building blocks to support the design when the receptor structure is known.

There is an abundance of software tools for drug design [3, 4]. Many algorithms with a geometric flavor have been proposed and implemented in this domain. We believe that our work is innovative in the following aspects. In our study of data structures for representing molecule geometry, we aim to develop techniques that are efficient under conformation change, viz., techniques that will allow for efficient dynamic update of the structure as the molecule conformation changes. We have devised and analyzed several models for efficient dynamic maintenance of such structures, as discussed in Section 2. The goal of conformational search, which is discussed in Section 3, is to produce low-energy conformations of ligand molecules. These conformations constitute the input to the pharmacophore identification procedure. Conformational search is computationally expensive and may take hours on high speed workstations [21]. We aim for speed and efficiency of calculation, and are willing to find low-energy conformations that are not necessarily energy minima. Finally, in the pharmacophore identification component, described in Section 4, we consider a set of active molecules, each of which may be present in many low-energy conformations, and use inter-atom distances and molecular surface information to identify common structural elements of these molecules. Our search for pharmacophores is guided by user-specified size and accuracy requirements, and our software allows for an interactive refinement of the solutions obtained.

2 Geometric Data Structures

A prevailing approach to modeling the geometry of static molecules is to represent each atom as a ball of fixed radius in a fixed placement relative to the other atoms [25]. The radius assigned to each atom depends on the type of the atom. There are various sets of recommended values for atom radii, and a prevailing set is known as the *van der Waals* radii. In spite of its limitations, this model, which we will refer to as the *hard sphere model* of a molecule, is widely used.

Various techniques in drug design use *molecular surfaces*. One type of molecular surface is simply the outer boundary of the union of the balls (or spheres) in the hard sphere model above. This type is often referred to as the van der Waals surface. There are two closely related types of surfaces: the solvent accessible surface [23] and Richard's smooth analytical surface [28]. See also [9, 10, 11] and the survey by Mezey [25] for an extensive discussion on molecular surfaces.

A basic question in the geometric manipulation of molecules is the following: Given a hard sphere model of a molecule M and a query atom q, report the atoms of M intersected by q. It has been shown [16] that for a molecule with n atoms, an efficient data structure requiring $O(n)$ storage space can be constructed such that queries of the above type can be answered in $O(1)$ time each, after $O(n)$ expected preprocessing time. This data structure was used [16] to obtain efficient algorithms for constructing molecular surfaces and for computing the visibility map of molecules.

In the next subsection we describe an implementation of the above algorithm for computing the molecular surface. In Section 2.2 we extend the intersection data structure to the dynamic case.

2.1 Molecular Surfaces — Construction and Visualization

We construct an analytic representation of van der Waals molecular surfaces (the same procedure applies to solvent accessible surfaces as well). The representation consists of the patches that each atom sphere contributes to the outer surface, as well as the adjacency relations between patches on neighboring atoms that share a common arc. The basic procedure in our implementation of the computation of molecular surfaces is the construction of the subdivision (or *arrangement*) on each atom sphere s induced by the circles of intersection of other atom spheres with s. The arrangement is constructed incrementally, by adding one circle at a time, and maintaining the *trapezoidal decomposition*[3] of the current arrangement. See Figure 1 for an illustration.

[3] The trapezoidal decomposition is a refinement of the arrangement of the circles by adding certain arcs of great circles through the poles. For more details on trapezoidal decomposition, see e.g [26]. In our implementation, we extend such arcs only from points of tangency of original circles with great circles through the poles.

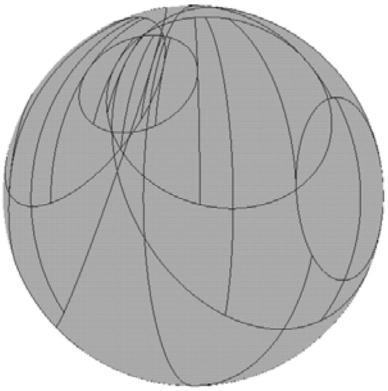


Fig. 1. Partial trapezoidal decomposition of an arrangement of small circles on an atom sphere.

We represent the arrangements of the circles using a quad-edge structure [13]. Once we obtain the arrangements for all atom spheres, we go on to construct the (van der Waals, or solvent accessible) molecular surface. This is achieved by adjusting the individual atom arrangements so as to retain only the faces in those arrangements that contribute to the outer surface, and by updating the edge information of edges that now border faces belonging to two different atom spheres (see [16] for details).

The arrangements on the spheres are used to compute the molecular surface area and the (possibly null) area contributed by each atom to the outer surface, as well as for graphic display of the intersection pattern of each atom with its neighbors (Figure 1). The information about area contribution of individual atoms is used in the pharmacophore identification module described below. Besides computing the outer surface, the program computes internal boundary components which bound *voids* (see, e.g., [11]) and outputs their surface area as well.

The implementation provides extensive facilities for interaction. It displays the molecule and the individual atom arrangements graphically, and also allows for re-coloring of the atoms according to several parameters such as the level of contribution to the outer surface. The edges of the arrangement on each atom sphere are colored to distinguish between the contribution of the atom to the outer molecular surface and to each void (if any). The program also provides various statistics on the data structures it uses and allows for experimenting with and fine-tuning certain parameters that control these structures.

2.2 Dynamic Maintenance

We can model the flexibility of a molecule to attain different conformations as if the atoms were rigid links of a robot linkage, and the bonds between some of the atoms are rotational (or rotatable) joints (see Section 3 for more details on this simplified model). We describe these kinematic constraints by a graph where each node corresponds to an atom sphere, and an edge between nodes describes a constraint. An edge can be either rigid — when there is a fixed relative displacement between the atoms that it connects, or rotatable — when there is a degree of freedom of rotation around a fixed line between the two atoms.

Suppose that we are given a sequence of update requests that are aimed at changing the conformation of the molecule. This is done by giving a sequence of joint angles to which we need to update the rotatable bonds. The sequence of updates is interleaved with intersection queries of the form: does a given query sphere intersect any of the atom spheres in the molecule at its current conformation? We require a strategy for maintaining a data structure which processes a sequence of updates and queries in optimal time, where the processing algorithm has the freedom to break or merge substructures.

In [15] we study several models of dynamic maintenance of such kinematic structures, and devise maintenance algorithms for them. We give a worst-case optimal strategy for the case where the molecule graph is a tree. The key idea behind this is that of a *balanced decomposition* of the tree into subtrees such that the number of subtrees (which corresponds to the cost of a query) is roughly the same as the size of each subtree (which corresponds to the worst case cost of an arbitrary update). We describe an efficient algorithm for constructing such a balanced decomposition. We also show that obtaining the optimal solution for a given sequence of updates and queries, even when the molecule graph is a path, is an NP-hard problem and we present approximation algorithms for this case.

3 Conformational Search

Searching the conformational space of small ligands is an important operation in the process of pharmaceutical drug design [22]. Given a function that computes the energy of the molecule, the goal of the search is to produce low-energy conformations that are geometrically distinct. These conformations can provide the input to a pharmacophore generation procedure, or can be used to screen large databases of protein molecules for possible docking receptors.

When conformational search is conducted, the prevailing practice is to consider only the torsional degrees of freedom of the molecule (see Figure 2 for an example). Other degrees of freedom, such as bond lengths between atoms or bond angles between consecutive bonds, are often ignored since their variation does not drastically change the molecular conformation. Another widely used approximation is to consider that the molecule is in vacuum. In this case, the energy of the molecule can be computed by empirical force-fields which consider only intramolecular quantities. The definition of such fields has been the result of intensive research [21]. Typically they have terms that involve pairs, triplets, and quadruples of bonded atoms, as well as pairs of distant non-bonded atoms. In reality, however, the conformation of a molecule is influenced both by intramolecular and intermolecular forces. Our techniques require as input a pro-

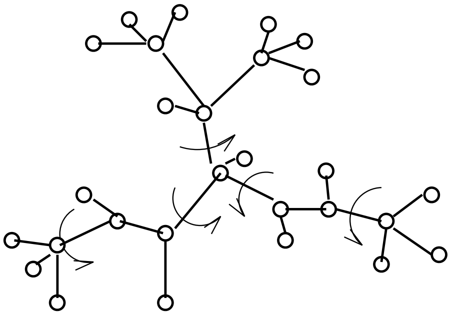


Fig. 2. The molecule of valine with its torsional degrees of freedom. A ball indicates an atom center, and a line segment indicates a bond between two atoms.

cedure that computes the energy of a molecule and will work if this procedure models the molecule in solution, or takes into account an external force field.

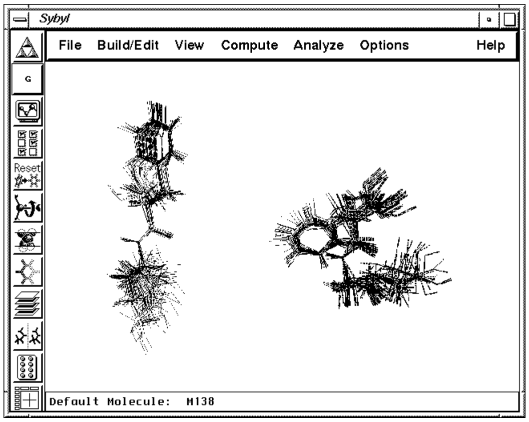
Conformational analysis is a very hard search problem when the molecule under consideration has more than 5 degrees of freedom. For a comprehensive survey of previous methods see [8, 21, 30]. During search, emphasis is placed on generating geometrically different conformations of a molecule within a user-defined energy interval. The underlying assumption is that one of these conformations will be adopted inside a receptor cavity. Selective generation of conformations with certain properties may also be desired. For example, if a pharmacophore is known, one may want to find low-energy conformations that retain the features of the pharmacophore at their relative positions. We describe below a method for generating low-energy conformations of a molecule, and our preliminary efforts to efficiently organize the resulting conformations in clusters and provide input for our pharmacophore generation procedure.

3.1 General Framework

Inspired by our success with probabilistic techniques for robot motion planning [17], we have implemented a search procedure which randomly samples the conformational space of small molecules. The energy of the molecules subjected to conformational search is computed by the Tripos Force Field [32]. Our method is divided into three steps: generation of random conformations, minimization of these conformations, and grouping or clustering of the minimized conformations.

Generation of conformations. During this step, a large number of conformations, frequently tens of thousands, are generated at random over the conformational space of the molecule. In contrast with previous search methods that discretize each torsional angle, we obtain a random conformation by selecting each torsional degree of freedom uniformly from its allowed range. However, if information is available for the preferred values of a particular torsion angle (such information has been collected from crystallographic databases [20]), then it is easy to select the value of this torsion angle according to a distribution that reflects the above information. The resulting structure is stored only if it avoids self-collisions which may result from intersections of the *hard spheres* of non-bonded atoms.

Minimization. An efficient minimizer [5, 29] is used to obtain conformations that are at local energy minima. It is important to note that minimization is the most time-consuming step during conformational search, and that its efficiency is crucial for the performance of the search. A large part of the running time is spent towards the end of the procedure, when the process tries to meet certain user-defined stopping criteria. We relax these criteria to achieve significant reductions in the running time at the expense of accepting conformations that are not at local energy minima. However, experiments show that conformations very close to a local minimum usually have similar geometries. These conformations are later grouped by our implementation in clusters as explained in the next paragraph. For the purposes of most applications (e.g., pharmacophore generation) only one conformation per cluster is retained. Hence, no relevant information is likely to be lost by avoiding the full minimization of conformations that belong to the same cluster. In the framework of conformation generation and minimization, the work described in Section 2.2 may prove beneficial in reducing the time spent on collision checking and on the calculation of various energy function terms.



Default Molecule: M138

Fig. 3. Two clusters of an actual drug molecule.

Distinct geometric conformations. Interpreting a set of a few thousands of conformations is challenging and can benefit from extensive research in the area of clustering [18]. Currently, our system partitions conformations into clusters by using an easily-computable measure of distance between conformations. In particular, three distinct atoms a_1, a_2, a_3 are first selected in the molecule. The distance of two conformations is defined as the sum of the Euclidean distances of their corresponding atoms, after identifying atom a_1, direction $a_1 a_2$, and plane $a_1 a_2 a_3$ in the two conformations. Clustering is performed by placing a given conformation in an existing cluster if its distance from the "center" of that cluster is less than a predefined value. If no such cluster is found, a new cluster is initiated. The center of a cluster can be the conformation with the lowest energy in the cluster. Despite its simplicity, this procedure seems to perform well in practice when a few thousands of conformations are involved. Examples of clusters for a molecule with 11 degrees of freedom are shown in Figure 3. We plan to investigate further the relationship between cluster generation and the extent to which minimization should be performed. Additionally, randomization and hashing techniques may reduce the complexity of clustering and permit its efficient use with very large numbers of conformations.

4 Pharmacophore Identification

In the literature, pharmacophore identification techniques model a molecule either as a graph (where atoms are vertices and bonds are edges) [31], or as a set of points in 3-dimensional space [6]. The second approach is often justified by arguing that pharmacophores tend to be non-local — bond information seems to have less of an influence on ligand-receptor binding than other properties such as hydrophobicity, hydrogen bonding, and electrostatic interactions (e.g., see [12]). Therefore, we model a molecule as a set of *labeled* points in three dimensions, where each point represents the center of an atom. In fact we expand each point into a small sphere of radius ϵ centered at that point, where ϵ represents uncertainty in locating atoms. Labeling allows us to generalize the notion of compatibility between atoms, so that we can take into consideration other structural/chemical properties of the atoms without having to change our algorithms. One type of labeling that we use is the surface area contributed by an atom to the outer molecular surface (see Section 2.1).

Since each molecule may exist in one of many low-energy conformations, the representation of a molecule consists of a collection of one or more point sets, each corresponding to one such conformation. The problem now is to determine one or more point sets that are congruent to a subset of some conformation in most of the molecules, where the only transformations permitted are translations and rotations. Note that allowing molecules to have multiple conformations makes the problem much more difficult, because now for any pair of molecules, we may potentially have to compare each possible pair of conformations in these molecules to determine even one pharmacophore.

It is known that if we restrict each molecule to one conformation, finding the largest such subset is NP-hard; from the point of approximate solutions, it is known to be hard to approximate [1] and only weak positive results are known [19]. There exist polynomial-time algorithms for the problem if we consider only two point sets [2], and these could be used to build algorithms over a larger number of point sets; however, these solutions are inefficient. In the next subsection, we describe an implementation based on a fast heuristic to determine pharmacophores between two point sets. Subsequently, we outline a scheme whereby such pharmacophores could be used to reconstruct a solution for many point sets efficiently.

4.1 The Two-Set Problem

Our implementation uses the following observation: if we can identify three pairs of atoms (one from each of the two conformations) that belong to a pharmacophore, then we can simply compute a transformation from the first triplet to the second. This transformation can be used to transform the first conformation, and a nearest-neighbor search will then yield the rest of the pharmacophore. We are not interested in very small pharmacophores, so we assume that any common substructure is not too small: specifically, for some constant $\alpha \leq 1$, there exists a common substructure P in conformations C_1, C_2 s.t.

$$|P| \geq \alpha \min(|C_1|, |C_2|).$$

This is a reasonable assumption because conformation sizes are typically of the order of 30-50 atoms, and non-trivial pharmacophores are of the order of 5-15 atoms. Therefore, α can vary between 0.1 and 0.3.

Now, we can randomly sample triplets of points from the smaller conformation. If we store distances between pairs of conformation atoms in a hash table, we can probe this table with the inter-point distances of the given triplet to determine triplets in the second conformation congruent to it. Then, as noted above, computing the transformation from one triplet to the other (if one exists) will yield a common substructure. We do this procedure repeatedly, until we obtain a substructure that satisfies the minimum size requirement.

It is easy to see that the probability of a random triplet belonging to a pharmacophore is at least α^3. Therefore, $\tilde{O}(\frac{1}{\alpha^3})$ iterations will yield the pharmacophore with high probability. In practice, we run the program in Monte Carlo mode – if a large invariant is not detected within these number of iterations, the program halts, reporting failure.

Our implementation has a graphic interface to display the results of the search. Various tolerances, such as errors in point locations, errors in distance measurements, and minimum subset sizes can be specified. Multiple solutions can be displayed simultaneously, and the user can change parameters and view the results in real time. It should be noted that we use actual sets of drug molecules with known pharmacophoric subsets, and the above procedure is quite effective in determining the *correct* solution (as detected by chemists).

4.2 Extension to Multiple Molecules

Previous attacks on the problem often use a clique-finding approach similar to the above to determine common subsets. This is then extended to multiple molecules in a natural way, where one molecule is compared with each of the rest simultaneously, and then the procedure is iterated [24]. This approach is simple, but computationally expensive, and cannot handle situations where a pharmacophore may exist in most, but not all, molecules. It should also be noted that these methods do not address the case where molecules can be in one of many conformations.

In general, it is easier to determine whether a given substructure exists in a conformation than to search for some common substructure that exists in two conformations. Therefore, our approach in the case of multiple molecules uses two primitives:

P1 that takes two sets of conformations and produces a set of common substructures from them, and

P2 that takes such a set of substructures, and another molecule, and *filters* the invariants through this molecule, retaining only those that are contained in it.

One simple strategy would choose two molecules arbitrarily, apply P1 on them, and use P2 repeatedly on the result and the rest of the molecules. Another possible strategy would be to view common substructures as sets of *pseudo-conformations*, and extend P1 to process such pseudo-conformations as well. Currently, we are investigating the efficacy of such strategies on various actual sets of drug molecules.

It is important to represent molecules and invariant groups compactly, to make P1 and P2 more efficient. To do this, we use techniques similar in spirit to geometric hashing [33], a technique used for pattern recognition in computer vision, and recently also in computational biology [27]. We hash all the conformations of all the molecules into a table, using a hash function of a geometric nature defined on pairs or triplets of atoms, and taking all such subsets. With every such table entry, we associate the conformation which contributed to a hit on it. Now, if we hash the pharmacophores in a similar fashion, we can identify which conformations are hit frequently, and so can deduce which molecules are likely to possess at least one conformation containing the pharmacophore.

References

1. T. Akutsu, M. Halldorsson, On the approximation of largest common point sets, *Proc. of the International Symposium on Algorithms and Computations, Springer-Verlag Lecture Notes on Computer Science* **834** (1987), pp. 405–413.
2. H. Alt, K. Melhorn K., H. Wagener and E. Welzl, Congruence, Similarity, and Symmetries of Geometric objects, *Discrete Comput. Geom.* **3** (1988), pp. 237–256.
3. L.M. Balbes, S.W. Mascarella, D.B. Boyd, A perspective of modern methods for computer aided drug design, *Reviews in Computational Chemistry* **5** (1994), VCH Publishers Inc., pp. 337–379.
4. D.B. Boyd, Compendium of molecular modeling software, *Reviews in Computational Chemistry* **4** (1993), VCH Publishers Inc., pp. 229–257.
5. R.P. Brent, *Algorithms for finding zeros and extrema of functions without calculating derivatives*, Ph.D. Thesis, Stanford University, 1971.

6. Brint, A. T. and Willet P, Algorithms for the identification of three-dimensional maximal common substructures. *J. Chem. Inf. Comput. Sci.* **27** (1987), pp. 152–158.
7. C.E. Bugg, W.M. Carson, and J.A. Montgomery, Drugs by design, *Scientific American*, December 1993, pp. 92–98.
8. D.E. Clark, G. Jones, P. Willet, P.W. Kenny, and R.C. Glen, Pharmacophoric pattern matching in files of three-dimensional chemical structures: Comparison of conformational searching algorithms for flexible searching, *J. Chem. Inf. Comput. Sci.* **34** (1994), pp. 197–206.
9. M.L. Connolly, Solvent-accessible surfaces of proteins and nucleic acids, *Science* **221** (1983), pp. 709–713.
10. M.L. Connolly, Analytical molecular surface calculation, *J. of Applied Crystallography* **16** (1983), pp. 548–558.
11. H. Edelsbrunner, M. Facello, P. Fu, and J. Liang, Measuring proteins and voids in proteins, *Technical Report*, HKUST-CS94-19, Department of Computer Science, Hong Kong University of Science and Technology, 1994.
12. R.C. Glen, G.R. Martin, A.P. Hill, R.M. Hyde, P.M. Woollard, J. Salmon, J. Buckingham and A. Robertson, Computer-aided design and synthesis of 5-substituted Tryptamins and their pharmacology at the 5-HT Receptor: discovery of compounds with potential anti-migraine properties, *J. Med. Chem.*, **38** (1995), pp. 3566-3580.
13. L.J. Guibas and J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics*, **4** (1985), pp. 74–123.
14. D. Gusfield and R. W. Irving, *The stable marriage problem: structure and algorithms.* MIT Press, Cambridge, 1989.
15. D. Halperin, J.-C. Latombe and R. Motwani, Dynamic maintenance of kinematic structures, Manuscript, 1996.
16. D. Halperin and M.H. Overmars, Spheres, molecules, and hidden surface removal, *Proc. 10th ACM Symposium on Computational Geometry*, Stony Brook, 1994, pp. 113–122.
17. L.E. Kavraki, *Random networks in configuration space for fast path planning*, Ph.D. Thesis, Stanford, 1995.
18. L. Kaufman and P.J. Rousseeuw, *Finding groups in data an introduction to cluster analysis*, Wiley, NY, 1990.
19. S. Khanna, R. Motwani, and Frances F. Yao, Approximation algorithms for the largest common subtree problem, Report No. STAN-CS-95-1545, Department of Computer Science, Stanford University (1995).
20. G. Klebe and T. Mietzner, A fast and efficient method to generate biologically relevant conformations, *J. of Computer Aided Molecular Design* **8** (1994), pp. 583-606.
21. A.R. Leach, A survey of methods for searching the conformational space of small and medium size molecules, *Reviews in Computational Chemistry* **2** (1991), VCH Publishers Inc., pp. 1–55.
22. T. Lengauer, Algorithmic research problems in molecular bioinformatics, *IEEE Proc. of the 2nd Israeli Symposium on the Theory of Computing and Systems*, 1993, pp. 177–192.
23. B. Lee and F.M. Richards, The interpretation of protein structure: Estimation of static accessibility, *J. of Molecular Biology* **55** (1971), pp. 379–400.

24. Y.C. Martin, M.G. Bures, E.A. Danaher, J. DeLazzer, I. Lico, and P.A. Pavlik, A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists, *J. of Computer-Aided Molecular Design* **7** (1993), pp. 83–102.
25. P.G. Mezey, Molecular surfaces, in *Reviews in Computational Chemistry*, Vol. I, K.B. Lipkowitz and D.B. Boyd, Eds., VCH Publishers, 1990, pp. 265–294.
26. K. Mulmuley, *Computational Geometry: An Introduction Through Randomized Algorithms*, Prentice Hall, New York, 1993.
27. R. Norel, D. Fischer, H.J. Wolfson, and R. Nussinov, Molecular surface recognition by a computer vision-based technique, *Protein Engineering* **7** (1994), pp. 39–46.
28. F.M. Richards, Areas, volumes, packing, and protein structure, in *Annual Reviews of Biophysics and Bioengineering* **6** (1977), pp. 151–176.
29. D.A. Pierre, *Optimization theory with applications*, Dover, NY, 1986.
30. A. Smellie, S.D. Kahn, and S.L. Tieg, Analysis of conformational coverage: 1. Validation and estimation of coverage, *J. Chem. Inf. Comput. Sci*, 35(1995), pp. 285-294.
31. Y. Takahashi, Y. Satoh and S. Sasaki, Recognition of largest common structural fragment among a variety of chemical structures, *Analytical Sciences* **3** (1987), pp. 23–28.
32. Tripos Associates Inc., Sybyl Manual, St. Louis, MO.
33. H. J. Wolfson, Model-based object recognition by geometric hashing, *Proc. of the 1st European Conference on Computer Vision* (1990), pp. 526-536.

This article was processed using the LaTeX macro package with LLNCS style