Journal of Computer-Aided Molecular Design, 7 (1993) 623–647
ESCOM

J-CAMD 220

SPLICE: A program to assemble partial query solutions from three-dimensional database searches into novel ligands

Chris M.W. Ho and Garland R. Marshall*

Center for Molecular Design, Washington University, St Louis, MO 63130, U.S.A.

Received 24 March 1993
Accepted 19 June 1993

Key words. Automated drug design; Ligand design, Database searching; Molecular graphics; Foundation; Splice

SUMMARY

SPLICE is a program that processes partial query solutions retrieved from 3D, structural databases to generate novel, aggregate ligands. It is designed to interface with the database searching program FOUNDATION, which retrieves fragments containing any combination of a user-specified minimum number of matching query elements. SPLICE eliminates aspects of structures that are physically incapable of binding within the active site. Then, a systematic rule-based procedure is performed upon the remaining fragments to ensure receptor complementarity. All modifications are automated and remain transparent to the user. Ligands are then assembled by linking components into composite structures through overlapping bonds. As a control experiment, FOUNDATION and SPLICE were used to reconstruct a known HIV-1 protease inhibitor after it had been fragmented, reoriented, and added to a sham database of fifty different small molecules. To illustrate the capabilities of this program, a 3D search query containing the pharmacophoric elements of an aspartic proteinase-inhibitor crystal complex was searched using FOUNDATION against a subset of the Cambridge Structural Database. One hundred thirty-one compounds were retrieved, each containing any combination of at least four query elements. Compounds were automatically screened and edited for receptor complementarity. Numerous combinations of fragments were discovered that could be linked to form novel structures, containing a greater number of pharmacophoric elements than any single retrieved fragment.

INTRODUCTION

The majority of drugs are ligands that interact with specific enzymes and/or receptors to modulate their biological activity. The challenge of the medicinal chemist is to develop such compounds rationally, given a thorough understanding of the underlying biochemistry. Even when the crystal structure of the pharmacologic target and computer-aided molecular design software are available, however, chemical modifications necessary to improve binding are often

*To whom correspondence should be addressed.

not immediately obvious. In fact, due to competitive, proprietary pressures, a requirement for nonobvious alterations to allow patent protection is often imposed.

To aid in this process, the utility of searching 3D chemical databases has become recognized in molecular drug design [1,2]. Database screening permits the retrieval of structures that match a given pharmacophoric pattern. The assumption in using this technique is that the conformations stored in the database or generated using algorithms that employ conformational flexibility [3] are reproducible in solution, given the appropriate steric and electrostatic environment. If so, complementary ligands possessing unique geometries can be recovered, providing insight as well as a foundation for further refinement.

Receptor sites are, however, complex both in their geometric features as well as their potential energy fields [4,5]. Such receptors can contain numerous interactive sites, consisting of hydrogen-bonding loci and hydrophobic subpockets. Probability alone dictates that the odds of finding a 'magic bullet' compound that matches all pharmacophoric loci are extremely small. These odds are further diminished as the majority of database-retrieval systems maintain only a single, static conformer per structure. Clearly, other strategies are necessary in order to develop optimal ligands efficiently.

Recently, methods have been developed which employ a 'divide-and-conquer' approach to ligand design. The active site is partitioned into subsites, each containing several pharmacophoric elements. Chemical fragments or 'building blocks' complementary to each subsite are then designed or retrieved from databases. Finally, fragments are linked to form aggregate ligands. The advantage of this approach is that ligand diversity can be tremendously augmented through the *combinatorial assembly* of numerous subcomponents.

DesJarlais et al. [6] were, perhaps, the first to employ this philosophy in a novel application of the program DOCK. This well-known program searches 3D databases of ligands and determines potential binding modes of any that will fit within a target receptor [7]. Only a single, static conformation of each database structure is maintained, however, eliminating ligand flexibility from consideration. Conformational flexibility was introduced by dividing individual ligands into fragments overlapping at rotatable bonds. Each fragment was first docked separately into various receptor regions. Attempts were then made to reassemble the component parts into a legitimate structure.

A current example of a similar approach is the program LUDI [8,9], written by Böhm. In this program, a receptor volume of interest is scanned to determine subsites where potential hydrogen bonding, or hydrophobic contact, can occur. Small complementary molecules are then chosen from a database and positioned within these subsites to optimize binding energy. The process concludes with the selection of various bridging fragments intended to link subsets of small molecules together.

Chau and Dean published a series of articles [10–12] addressing whether small molecular fragments, with transferable properties, could be generated for further use in automated site-directed drug design. A program was developed to generate all 3-, 4- and 5-atom fragments containing any geometrically allowed combination of H, C, N, O, F and Cl. Aromatic fragments were produced as well. Searches of the Cambridge Structural Database (CSD) [13] were performed to determine the most frequently occurring fragments. In order to utilize these fragments as components for ligand assembly, more data were necessary to characterize them. Therefore, they were analyzed to statistically ascertain bond lengths from the CSD in order to provide some

geometrical constraints for structure assembly. Lastly, the transferability of atomic residual charges was studied by comparing charges generated for the atoms in each fragment with charges calculated for whole molecules containing the fragment.

Miranker and Karplus have described a technique for determining energetically favorable positions and orientations for functional groups in the binding site of proteins with known 3D structure. Their method is called the multiple copy simultaneous search (MCSS) [14]. It is implemented by placing several thousand copies of a functional group randomly within the active site. Effective binding positions are then determined by subjecting the groups to simultaneous energy minimization and/or quenched molecular dynamics. This method was validated by studies of the sialic acid binding site of the influenza coat protein, hemagglutinin. Functional-group minima determined through MCSS corresponded with those of the ligand in a cocrystal structure. MCSS can aid the design of ligands that incorporate such functional groups. Once the positions of complementary fragments have been established, other techniques can be used to link them into valid ligands.

In previous work, we described the 3D database search and retrieval program FOUNDATION [15]. This program contains search functionality common to many such programs. Thus, the user can seek structures containing specific 3D configurations of atoms and/or bonds with chemical requirements. However, FOUNDATION is unique in that it uses clique detection algorithms to retrieve partial query solutions. This permits the recovery of structures containing any combination of a user-specified minimum number of query elements. With regard to the divide-and-conquer approach, FOUNDATION can provide a wealth of building blocks to construct novel ligands. However, the challenge of the medicinal chemist remains to utilize the results effectively. Few strategies have been devised to address the transformation of components into distinct structures.

The difficulty arises from the volume of data one must process. Consider a scenario where one hundred potential ligand fragments are recovered. Given that all are chemically valid, each structure must first be screened to ensure steric and electrostatic complementarity with the active site. Structures that conflict sterically with the receptor must be withdrawn, although a fair percentage can be recovered by pruning appropriate atoms. What remains is the difficult task of scrutinizing fragments to find combinations that produce suitable ligands. In just one hundred components, there are nearly five thousand unique pairs of structures, along with triplets, quadruplets, and more to consider. In fact, situations can arise where segments of several different fragments may be necessary to piece together a legitimate ligand.

A medicinal chemist could not realistically review and retrieve all the useful combinations of fragments. To accomplish this, all structures must be visualized simultaneously. In essence, this would entail a 3D 'bond histogram', providing an inventory of all bond loci in space. Such a tool could reveal sites where bonds from two or more different fragments were appropriately positioned for linking. In this manner, conglomerate ligands could be systematically constructed.

Lewis and co-workers have developed one means of generating ligands from substructures [16–18]. In previous work [16,17], Lewis and Dean introduced the use of *spacer skeletons* as a means of generating site-specific ligand structures. These spacer skeletons were assemblies of molecular substructures that were fitted within a receptor's binding site. Using the accessible surface of the receptor atoms as a constraint, structures of legitimate binding ligands were extracted with an automated, molecular-editing procedure.

To develop more complex ligands, Lewis et al. recently developed an integrated drug-design package called BUILDER [18], which employs database searching techniques, structure generation algorithms, and an interactive graphics modeling environment. BUILDER utilizes the well-known DOCK program [7] to search 3D databases and retrieve candidate structures that possess steric complementarity with a receptor binding site. Regions critical for ligand binding (i.e. the pharmacophore) are designated by the user as *molecular zones*. The program allows a user to view, prioritize and interactively edit all structures that contain atoms within a zone. Fragments that sterically clash are flagged. Iterative refinement continues until satisfactory structures result, which manifest the desired functionality. These structures are then transformed into entities termed *molecular lattices*, which are essentially large supramolecular chemical graphs whose nodes (atoms) and edges (bonds) span the binding site of a receptor of interest. A molecular lattice is formed by fusing selected structures, using real and virtual bonds according to geometric and chemical criteria [19]. Search algorithms are then used to link compounds in the various zones into valid binding ligands.

In providing the medicinal chemist with such a tool, we must be cautious that the amount of information is not overwhelming. With a large number of structures, one can expect both a significant amount of noise and a high level of redundancy. The role of the computer should be to *distill* this information to its essence, while eliminating mundane and distracting routines. As such, the investigator is given the proper environment where creative efforts can be focused upon the task of ligand design and synthesis. Using these guiding principles, we have developed a program called SPLICE, which stores, edits, groups and joins retrieved structures from databases in an automated fashion to form unique aggregate ligands.

DISCUSSION OF THE PROGRAM

Overview of ligand design

Figure 1 details our ligand development process, demonstrating how SPLICE is utilized. Given the crystal structure of a ligand–receptor system, the pharmacophoric elements are first isolated. These are the functional groups of the ligand that are crucial to receptor recognition and binding. Our goal is to use the divide-and-conquer approach to produce novel compounds that maintain the pharmacophoric pattern. Thus, our first task is to generate complementary structural fragments. The isolated pharmacophoric elements are transformed into a 3D search query, which specifies both the position of each element relative to one another, as well as acceptable atom types. We then employ FOUNDATION to search and retrieve all chemical components from our databases that contain a specified fraction of the pharmacophoric elements. By accepting structures that match various portions of the pharmacophore, we can retrieve a large number of diverse building blocks. FOUNDATION aligns each hit with the pharmacophore; thus, structures are docked in the active site with the appropriate orientation.

FOUNDATION approximates the fit of each component within the active site, and will discard structures that are clearly contacting the receptor. However, a more rigorous screening is required at this stage to assure steric compatibility. This is performed by the EDIT module of SPLICE. Structures that require subtle modifications are pruned using a standard procedure. What results is a mass of components residing within the active site that link various elements to one another.

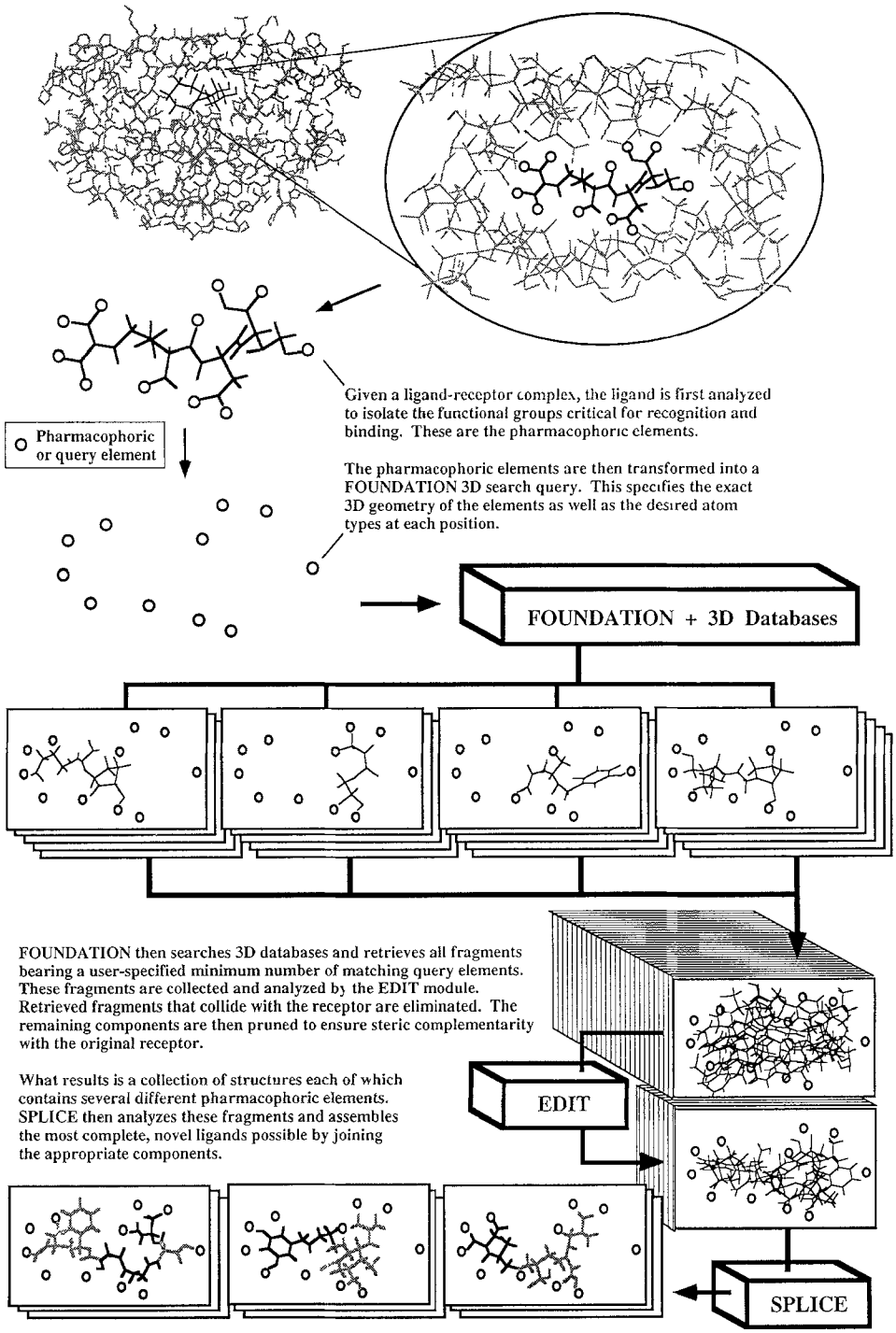


Fig. 1. Overview of FOUNDATION-SPLICE ligand design system.

SPLICE takes these structures and forms new ligands that contain a greater number of pharmacophoric elements than any single component. This is accomplished by linking ('splicing') two fragments that contain different portions of the pharmacophore with a chemical bond. Through iterative processing, SPLICE determines the largest, most complete ligands through the assembly of appropriate components.

With this chain of events in mind, we now describe each step in detail.

Generation of fragments

The core of this approach is the generation of molecular fragments. Although structures from any source may be satisfactory, those which more effectively complement the target receptor are preferred. By finding fragments that each contain several pharmacophoric elements, we maximize complementarity with minimal structural mass. This reduces the total number of atoms to be processed, and produces better ligands. Although DOCK is designed to find structures that sterically fit within an active site, it does not specifically target pharmacophoric atoms or bonds. As stated above, we utilize the 3D database search program FOUNDATION [15]. This program is suited to the 'divide and conquer' strategy since it retrieves all fragments containing a user-specified minimum number of matching query elements. Unlike DOCK, FOUNDATION, along with the majority of similar programs, retrieves structures that match a particular configuration of atoms in space. This guarantees that the fragment will place the desired functionality in the correct orientation. However, these structures may contain atoms that collide with the receptor.

Screening procedure performed on each component to determine active-site viability:

Determine atoms connecting query elements = **PATH**
IF **PATH** intersects **RECEPTOR**
  THEN reject structure!!

Determine ring atoms = **RING**
IF **RING** intersects **RECEPTOR** and contains **PATH**
  THEN reject structure!!

Structure is acceptable!! Editing procedure performed to ensure active-site complementarity:

Determine ligand atoms contacting receptor = **CONTACT**

For each **CONTACT atom:**
  Remove atom and neighbors from structure
  IF atom and/or neighbors is part of **RING**
    THEN IF **RING** is solitary
      THEN remove entire **RING**
      ELSE remove non-anchored portion(s) of **RING**

Fig. 2. Fragment screening and editing procedure used by SPLICE to ensure steric complementarity of chemical components.

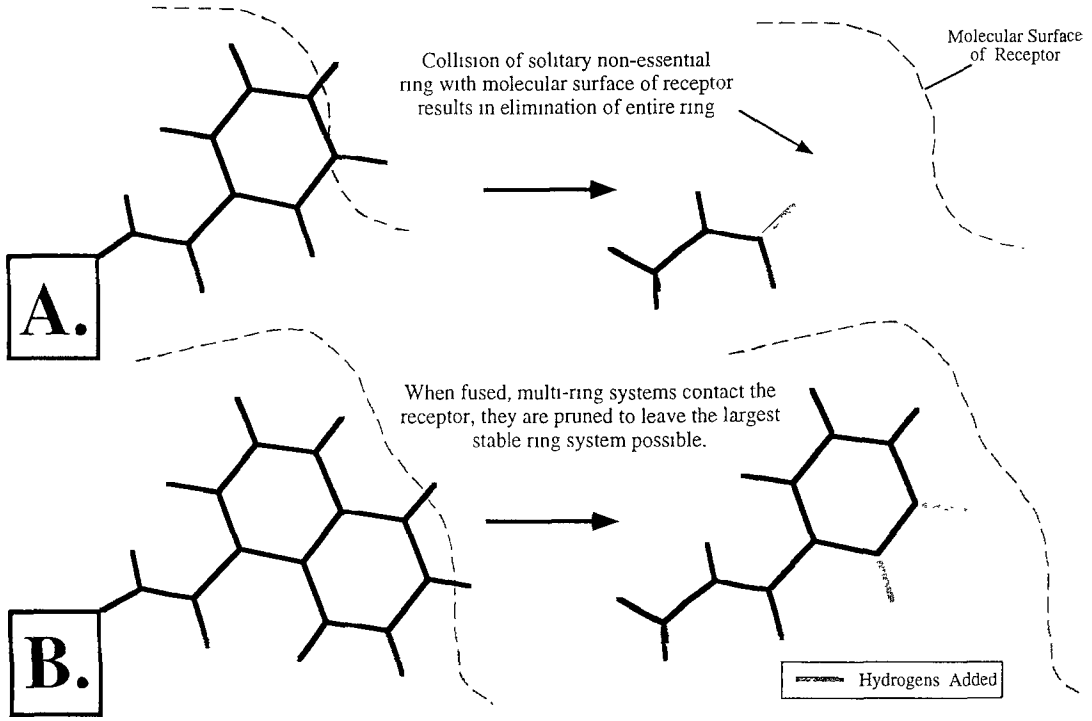


Fig. 3. Selective pruning of ring systems which collide with the receptor by the EDIT module of SPLICE.

Thus, each fragment must be processed to ensure steric complementarity. This task, detailed below, is automated and remains transparent to the user.

Fragment editing

Although FOUNDATION approximates whether atoms matching the query are within the active site, a more rigorous screening is necessary, especially if data from other sources are used. Stringent criteria described below are first used to eliminate those structures that are incapable of maintaining their pharmacophoric pattern within the active site. Then, with the emphasis on minimizing user burden as well as preserving the recovered structures, we employ a rule-based, automated protocol that identifies and processes those structures that require pruning to attain steric complementarity.

SPLICE's screening and editing procedure is outlined in Fig. 2. Each structure is first examined using a depth-first search [20] to determine the atoms comprising the shortest paths between all pharmacophoric elements. These atoms are termed *path atoms*. If any path atom collides with the receptor, the structure is rejected since maintaining the pharmacophoric pattern would be impossible. Of course, receptors are not static entities. Receptor atoms that are not sterically constrained could be displaced upon binding, allowing structures to bind that otherwise would contact these atoms. Therefore, SPLICE decreases the van der Waals radii of user-designated receptor atoms to compensate for atomic motion. Processing is then continued to locate all ligand

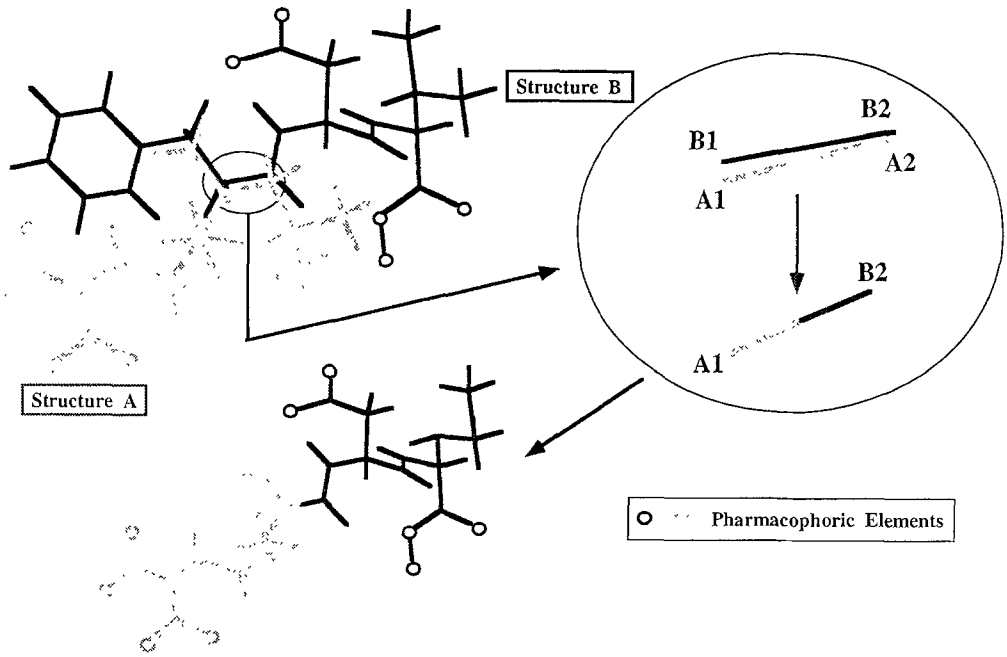


Fig. 4. Formation of a hybrid structure by splicing two components at an overlapping bond.

atoms present in any ring system. If a ring clashes with the receptor and contains any path atom, the parent structure is also rejected.

Structures that pass the above criteria are edited with an automated procedure to ensure steric fit. Starting from the most distal atoms and proceeding towards the interior, the structure is systematically checked for receptor contact. As described in Fig. 2, fragments are clipped off until the structure is satisfactory. Atoms that contact the receptor are removed along with all neighbors. Should any part of a nonessential ring collide with the receptor, the entire ring structure is deleted. In the case of receptor contact with fused multiring systems, only the atoms that are not stably locked in any ring are deleted. These ring editing scenarios are depicted in Fig. 3. In this figure, fragment A contains a solitary ring that is nonessential, meaning that it is not involved with any path or query element. Since a portion of the ring contacts the receptor, we eliminate it completely. On the other hand, fragment B contains a nonessential, fused, multiring assembly. Since one of the rings contacts the receptor, we eliminate it as well. However, we limit our pruning to leave the largest stable ring system possible. When completed, each structure should reside within the active site, yet retain the necessary elements that maintain its conformation.

Component assembly

Given a large set of overlapping components, each containing a subset of the pharmacophore, our goal is to form novel ligands by assembling appropriate structures. To do so, we must first determine which pairs of structures can be joined. The union of two components requires a precise alignment of atoms to allow the formation of a linking bond. One must confirm that bond angles and lengths are within reason to ensure legal geometry. We accomplish this by isolating pairs of structures that both contain a mutual bond whose atoms overlap nearly perfectly. This

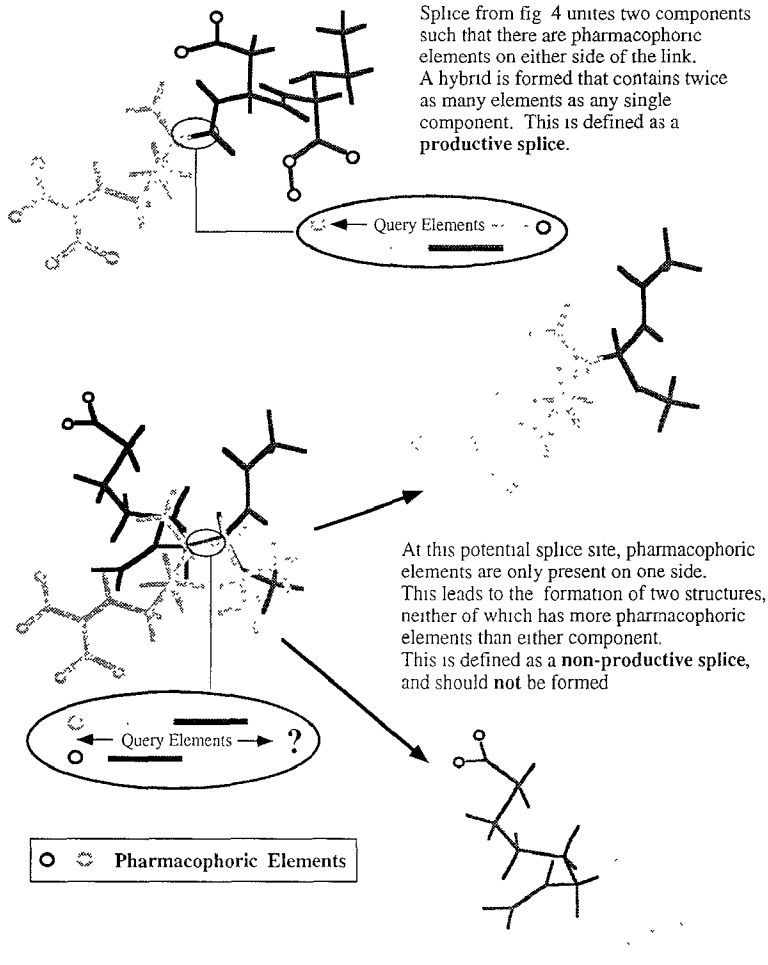


Fig. 5. The distinction between productive and nonproductive splices.

is demonstrated in Fig. 4. In this example, structure A contains a bond A1-A2 that overlaps nearly perfectly with bond B1-B2 in structure B. If the distances between atoms, A1 $\leftrightarrow$ B1 and atoms A2 $\leftrightarrow$ B2 are within a minute user-specified range, then structures A and B can be joined by creating the new bond A1-B2. We define this procedure as *splicing* A and B together at bond A1-B2 to form a new hybrid structure.

However, it is important to realize that not all pairs of structures that contain superimposed bonds should be spliced. In fact, the majority of these potential assemblies may be non-productive or chemically impossible. At least three examples exist:

The first concerns the splicing of bonds of different order. Atoms from different bond orders cannot be spliced together, since those at each end of the spliced bond would have an incompatible hybridization. Furthermore, an atom's hybridization cannot simply be altered as this disrupts its geometry. In practice, we limit splicing to single bonds. This permits a full range of motion about the mutual bond, allowing the conformation of both segments to be attained.

The second example is depicted in Fig. 5. This illustrates the difference between a productive

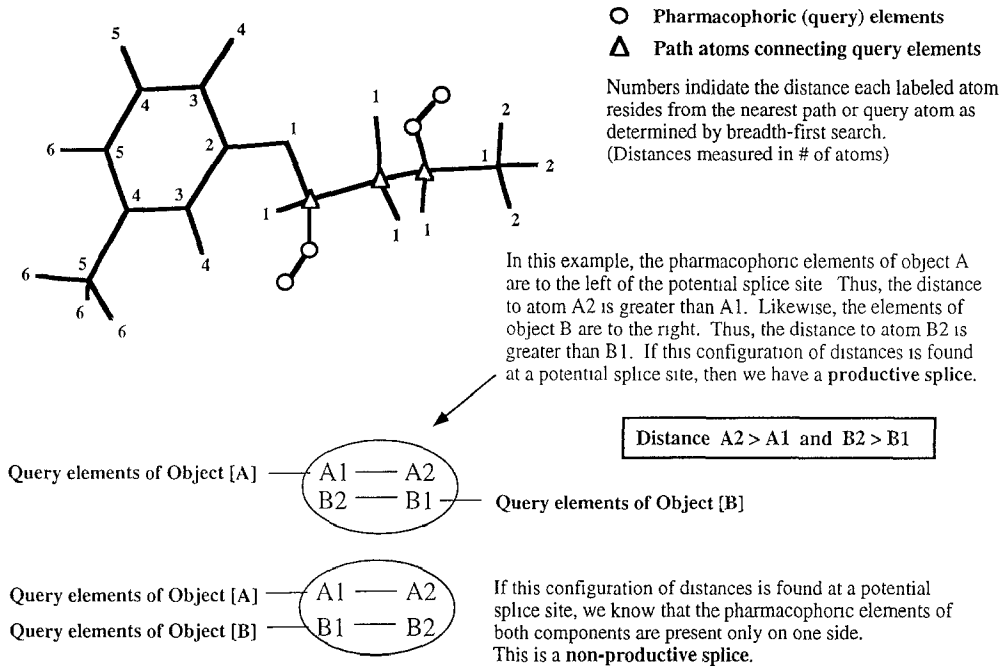


Fig. 6. Result of breadth-first search performed on each chemical component to reveal distances that atoms reside from query and path atoms and their use in screening for potential productive splices.

and a nonproductive splice. The *productive splice* is defined as the joining of two components such that the pharmacophoric elements of one segment are linked to the elements of the other. In essence, pharmacophoric elements must be present on both 'sides' of the link. This ensures that a useful, if valid, compound will result from the joining of the fragments. Such a compound is desirable, since it should contain a greater number of pharmacophoric elements than any individual component. Conversely, a *nonproductive splice* occurs when the linking of two components fails to increase the number of pharmacophoric elements in the hybrid ligand. These compounds should be rejected.

To validate a productive splice, a search for query elements is initiated at the linker bond, proceeding in both directions for each structure. The link is accepted if two conditions are met: (1) At least one pharmacophoric element must be found on either side of the link from different structures. (2) The sum total of all the pharmacophoric elements found is equal to or greater than a user-defined minimum. Furthermore, the path atoms on either side of the link must not bump one another. SPLICE allows the user to specify a threshold number of combined elements that must be found for an acceptable link.

Although graph-searching algorithms are efficient, there is considerable CPU 'overhead' for retrieving and manipulating atomic coordinate information and connectivity data. Considering that thousands of potential splices may need evaluation, a quick method is required to determine when a link might be productive, so that these calculations are performed only when necessary. This is accomplished by performing a breadth-first search [20] on all structures to determine the distances each atom resides from the query element paths. These distances are stored with the

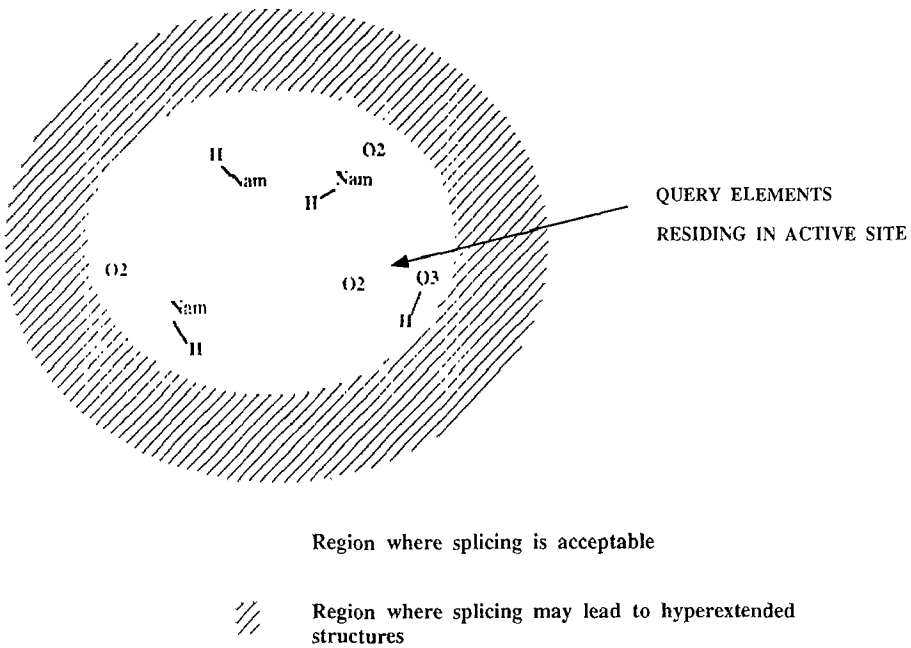


Fig. 7. Regions where spliced bonds are most desired.

coordinate information of each atom. As shown in Fig. 6, these distances can be used to quickly determine whether pharmacophoric elements are present on either side of a potential splice. Consider a potential splice between bond A1-A2 of component A and bond B1-B2 of component B. Since the distance of atom A2 is greater than A1, we know that there are query elements to the left of the link. Similarly, since the distance of atom B2 is greater than B1, we know that there are

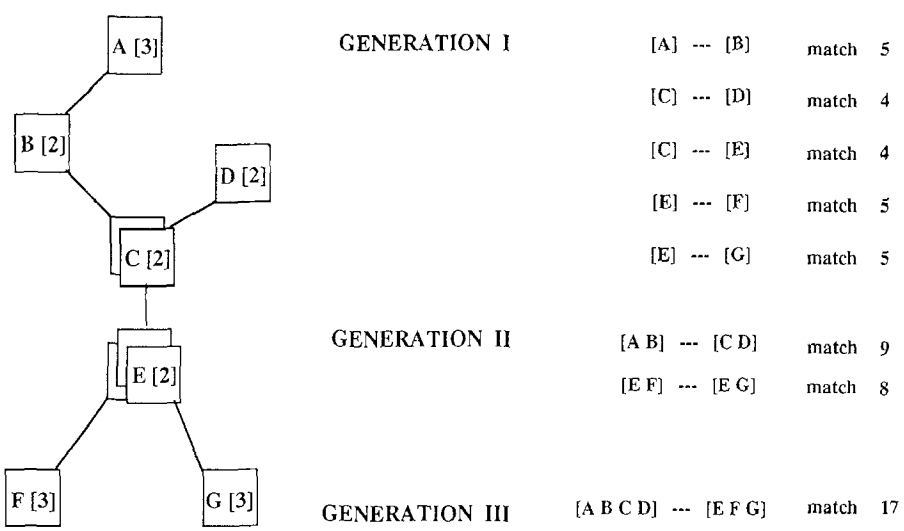


Fig. 8. Iterative generation of largest possible ligands.

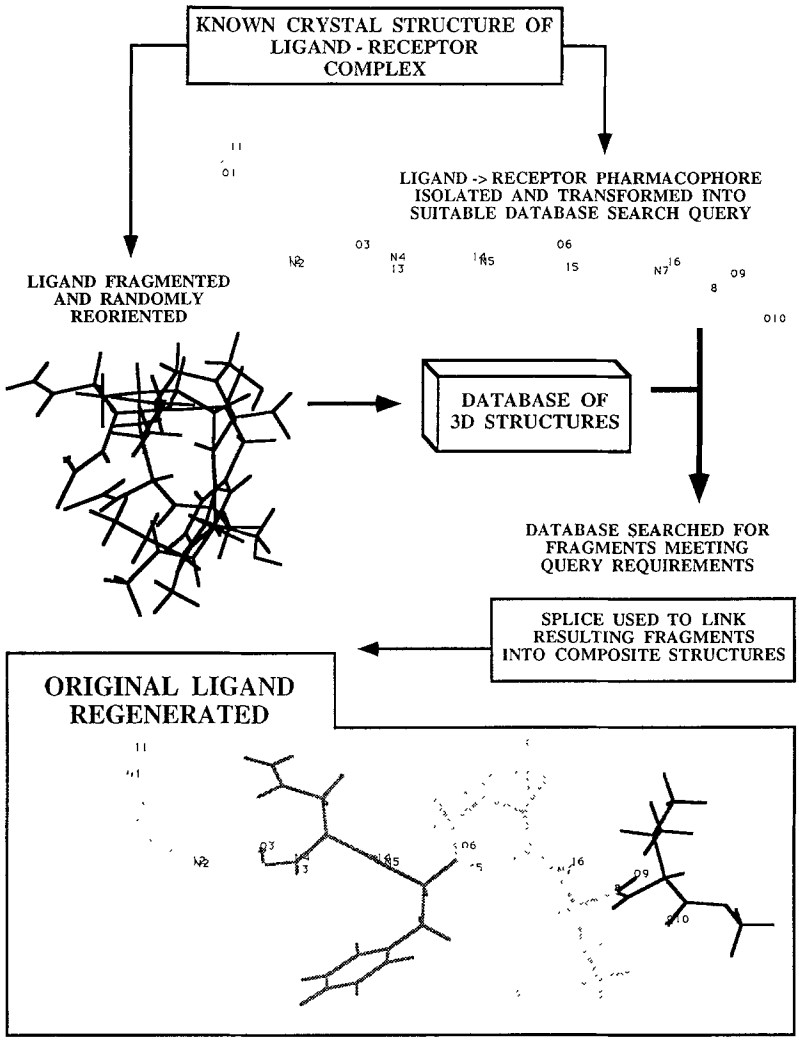


Fig. 9. Schematic of control experiment.

elements to the right of the link as well. In the second case, the distances indicate that there are only elements found on one side of the link (in both compounds A and B). Thus, we need not perform the atom-by-atom search to verify a productive splice.

The third example of forbidden interactions concerns ring systems. A splice may be rejected if the mutual bond resides within a ring. The ring is crucial to maintaining the conformation of that portion of the fragment. Splicing into a ring system may disrupt its stabilizing influence. However, the user is allowed the choice of whether such assemblies are permitted.

Bond management

Since hundreds of molecular structures may be needed to construct novel ligands, tens of thousands of bonds may have to be processed. With such a large volume of data, it is imperative

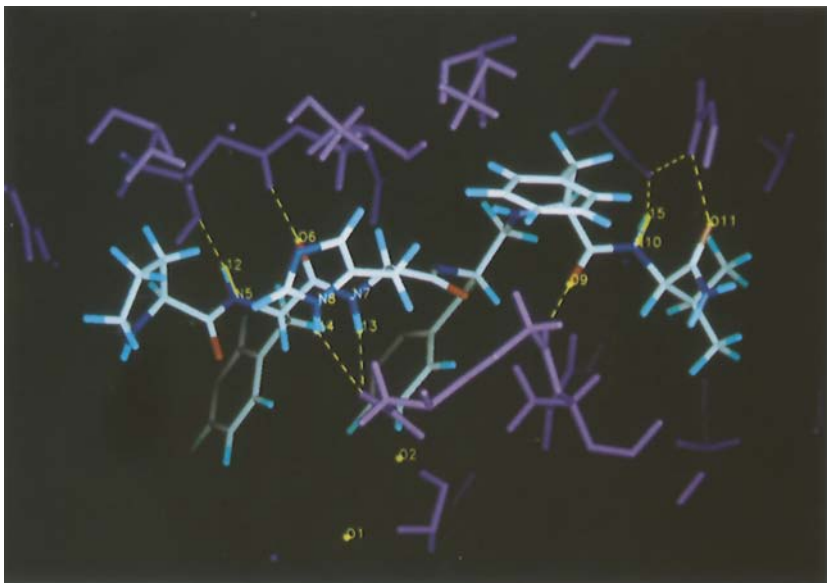


Fig. 10. X-ray crystal structure of proteinase–inhibitor complex.

to filter out the noise from the signal. As discussed above, compounds are individually screened to remove sterically unfavorable regions, leaving the atoms necessary to link query elements and maintain a stable conformation. This eliminates a large percentage of the bonds, leaving a much more manageable quantity. We also impose another filter to restrict the region where splicing can occur to a volume that resides within a user-specified distance from any query element. As shown in Fig. 7, splices that occur within the acceptable shaded zone produce short, interconnecting links that are desired between query elements. However, splices occurring in the striped regions form unusually large 'looped' structures. In testing our program, we found that these structures were nearly always rejected due to intrastructural contacts as the looped regions folded back on themselves to connect the query elements. Moreover, the entropic cost of binding a ligand containing lengthy chains, separating pharmacophoric elements, is much greater than for a smaller, more compact structure. Furthermore, the elimination of bonds residing in these regions from the lattice greatly improves program speed.

When all useful bonds have been isolated, they are sorted according to their coordinates along the x-axis. As a result, potentially overlapping bonds are ordered serially, thereby grouping candidates to be screened for precise overlap. We then progress along the x-axis, systematically comparing bonds that are within a user-specified distance of one another for precise overlaps. Once a match is found, the link is validated as described above to determine if appropriate atom types are involved, whether productive splices are created, and whether ring structures are disrupted.

In any structural database, one often finds groups of compounds that are structurally similar. Such structures can be stereoisomers, conformers or series of compound derivatives. The problem is that if one such compound is retrieved from a 3D search process, its siblings will likely be found as well. If these compounds are analyzed, numerous overlaps will be detected due to structural

Database searched: cho_dbase
Threshold number of query atoms = 4
Number of structures skipped = 0
Hit limit = 500

ATOMS

1	1.72820	6.81030	4.84730	+/- 0.150
2	1.13860	3.34980	4.31920	+/- 0.150
3	-5.25710	2.24610	5.41650	+/- 0.150
4	-4.54040	2.85070	5.56950	+/- 0.150
5	-3.64160	4.85040	-0.57760	+/- 0.150
6	-2.47360	3.08440	-2.08860	+/- 0.150
7	-0.56260	2.98840	-0.78760	+/- 0.150
8	0.08040	5.38740	-2.50660	+/- 0.150
9	4.38240	-1.27060	-0.72360	+/- 0.150
10	5.21240	-3.37060	-1.09860	+/- 0.150
11	7.49440	-4.24160	-2.36260	+/- 0.150
12	-4.40510	4.34450	-0.97900	+/- 0.150
13	-0.16530	3.24430	0.09370	+/- 0.150
14	0.02960	5.46980	-1.48120	+/- 0.150
15	5.01730	-4.27160	-1.48600	+/- 0.150

BONDS

[1] query atoms: 4 -> 3
[2] query atoms: 5 -> 12
[3] query atoms: 7 -> 13
[4] query atoms: 10 -> 15
[5] query atoms: 8 -> 14

TYPE: Atom # 3 => Hydrogen only
TYPE: Atom # 12 => Hydrogen only
TYPE: Atom # 13 => Hydrogen only
TYPE: Atom # 14 => Hydrogen only
TYPE: Atom # 15 => Hydrogen only
TYPE: Atom # 5 => Any Nitrogen or Oxygen
TYPE: Atom # 7 => Any Nitrogen or Oxygen
TYPE: Atom # 8 => Any Nitrogen or Oxygen
TYPE: Atom # 10 => Any Nitrogen or Oxygen
TYPE: Atom # 1 => Oxygen only
TYPE: Atom # 2 => Oxygen only
TYPE: Atom # 6 => Oxygen only
TYPE: Atom # 9 => Oxygen only
TYPE: Atom # 11 => Oxygen only

RMS fit threshold set at 0.1500 angstroms

CAVITY file = cavity.fil Cavity inclusion dist = 1.000000
Specified fraction of path atoms in cavity = 0.750000

Ad Infinitum constraint set at 60000.000000 iterations.

Progress monitored every 200 structures

Fig. 11. Search query derived from proteinase–inhibitor crystal complex.

similarity. Unfortunately, these compounds are not very useful since they connect the same query elements and do not yield novel ligands if spliced in any way. Thus, we can tally the number of overlapping bonds between any two structures under consideration. A user-specified threshold number of overlapping bonds can be defined to designate structural similarity. If two compounds have a greater number of overlapping bonds, they can be disregarded. This directs the CPU to process more effective combinations.

SPLICEPARAM.DAT

<MISCELLANEOUS>

0.50	Bond resolution factor (angstroms)
2.0	Bond inclusion distance (angstroms)
0.50	vdW contact stringency - INTER structure
0.80	vdw contact stringency - INTRA structure
10	Similarity index = # of matching bonds
5	Mininum number of combined query elements

TRIPOS ATOM TYPES AND VDW RADII ----------

31 Different atom types defined:

C.3	1.520	C.2	1.530	C.1	1.540	C.ar	1.530
N.3	1.450	N.2	1.480	N.1	1.500	N.ar	1.480
N.am	1.450	N.pl3	1.500	N.4	1.450	O.3	1.360
O.2	1.360	S.3	1.700	S.2	1.720	S.O	1.700
S.O2	1.700	P.3	1.750	H	1.080	F	1.300
Cl	1.650	Br	1.800	I	2.050	Si	2.100
LP	0.8600	Du	0.000	Na	2.300	K	2.800
Ca	2.750	Li	1.800	Al	2.050		

Fig. 12. User-defined parameters for SPLICE.

Iterative postprocessing

When processing is completed, SPLICE automatically joins each matched pair of fragments to form a novel structure. Molecular information is stored in the form of SYBYL mol2 files. Depending upon the minimum number of elements specified for an acceptable combination, resulting structures may not satisfy the entire pharmacophore. As shown in Fig. 8, several fragments may exist that must be combined to form the most complete ligand. To find these combinations, SPLICE is repeatedly run using the structures produced in the previous generation. With each iteration, the number of query elements required is increased. Usually, the most complete ligand is constructed within three iterations.

COMPUTATIONAL METHODS AND RESULTS

Implementation

SPLICE is written in C and presently runs on the Silicon Graphics IRIS, SUN, and E&S ESV machines. The program is compatible with SYBYL [21] molecular modeling software, using mol2 files as input and output, but can easily be modified to accept other molecular coordinate formats.

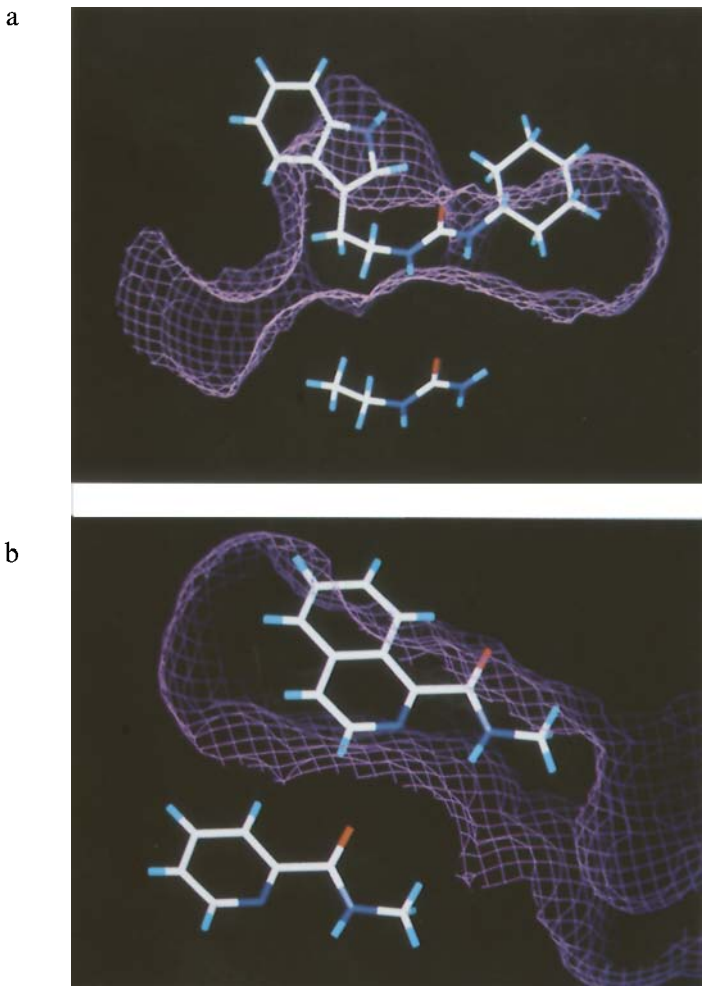


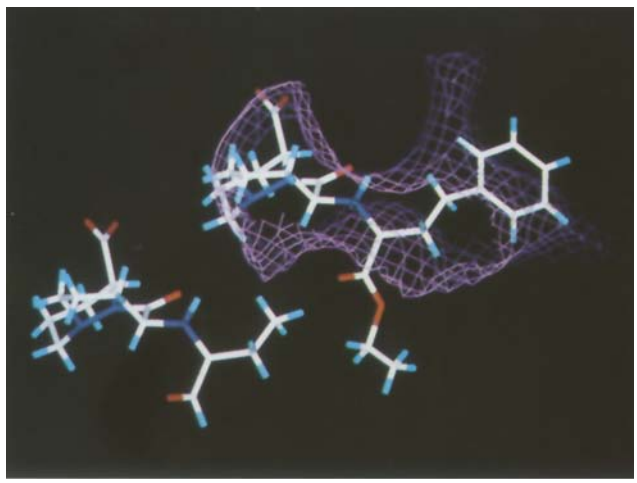
Fig. 13. Examples of structural pruning to ensure steric complementarity.

SPLICE can be licensed from Washington University (contact the Center for Molecular Design for licensing information).

Control experiment – Reconstruction of known ligand

As a control experiment, we first tested whether FOUNDATION and SPLICE could reconstruct a known ligand after it had been fragmented and 'buried' within a sham database. The crystal structure of HIV-1 protease complexed with the inhibitor JG-365 [22] was used. As detailed in Fig. 9 the hydrogen-bonding atoms critical for recognition and binding were extracted and transformed into a FOUNDATION database search query. This file specified the {x,y,z} coordinates of each element as well as acceptable atom types. The inhibitor was then extracted and fragmented into four components, each containing three or four pharmacophoric elements. These fragments were each reoriented and translated randomly. Hydrogen atoms were added to

c



d

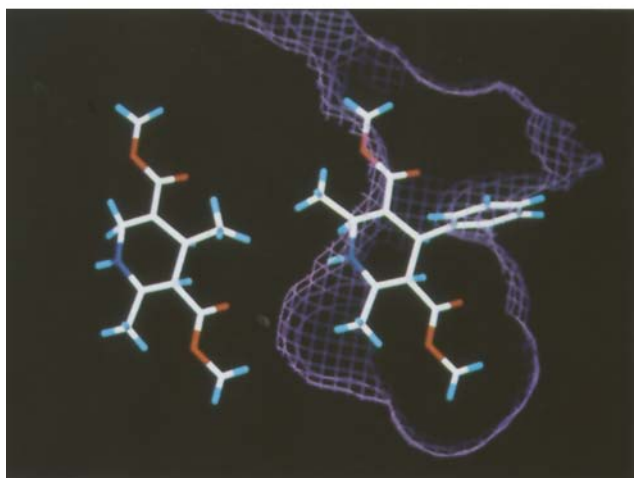


Fig. 13 (continued).

each fragment at the atom where the inhibitor was cleaved in order to give each component an appropriate 'overlap' with the next (if realigned properly).

The fragments were then added to a database containing 50 different small molecules. To eliminate proximity bias, each inhibitor component was separated by at least 10 structures. Using FOUNDATION, the database was searched to retrieve compounds containing any combination of three or more query elements. The program generated 150 fragments, reorienting each structure to match its corresponding query elements. The combination of a large query and the small number of required matching elements produced the relatively high number of hits. As such, structures could assume numerous query matching orientations. In fact, the four original components accounted for 30 unique hits. Using the fragments, SPLICE generated over 13 novel structures, containing a minimum of seven matching elements. SPLICE was then run again using these structures as input. By raising the minimum number of desired elements to 16 (the entire

query), two second-generation structures were created. One was the original ligand; the other was a derivative where a reoriented fragment, #1, was substituted in place of #4.

Development of novel acid proteinase inhibitors

To demonstrate the ability of SPLICE to generate novel structures, we developed a search query using the X-ray structure of acid proteinase (*Rhizopus chinensis*) shown in Fig. 10, cocrystallized with an inhibitor bound in the active site [23]. Again, the inhibitor was extracted from the enzyme complex and all atoms forming intermolecular hydrogen bonds were isolated. These atoms, along with several other hydrogen-bonding sites, were transformed into the query listed in Fig. 11. Each query element was assigned an error margin of 0.15 Å in addition to a global rms fit specification of 0.15 Å. Bonds between appropriate query elements (H-bond donors) were specified. Atom-type specifications were assigned depending upon the role of the element as hydrogen-bond acceptor (any oxygen) or donor (any N or O → hydrogen). Furthermore, a file containing a filler-lattice [24] delineating the extent of the enzyme active site was used to ensure that all hits would contain path atoms within this region. Using this query, a search was conducted against a subset of the Cambridge Crystallographic Database [13] to retrieve structures containing a minimum of four matching atoms.

Approximately 40 000 database compounds were scanned in about six hours of CPU time (SGI-380), generating 131 hits. The input data submitted to SPLICE consisted of a listing of hits, the search query used to retrieve the structures, a 498-atom subset of the receptor centered about the active site, and a file of user-defined parameters. These parameters are listed in Fig. 12. An rms deviation of 0.500 Å (bond resolution factor) was allowed between bonds considered for overlap. A van der Waals stringency factor of 0.500 was used to reduce the radii of all receptor atoms to allow for flexibility in determining receptor contact. A limit of 10 matching bonds was set to disregard pairs of structures containing a greater number of near-identical bonds. The region of acceptable splicing was limited with a bond-inclusion distance of 2.0 Å from any query element. Finally, a minimum number of five query elements was required of linked structures returned.

SPLICE required approximately 2 min of CPU time (SGI-380) to edit the entire list of structures for steric complementarity with the proteinase-active site. A SYBYL macro file was generated that, when executed in SYBYL (5 min execution time), pruned the sterically offending atoms of each acceptable structure and filled valences with hydrogens. Four examples of modifications are shown in Fig. 13. Sample output from the edit log is listed in Fig. 14. Of the 131 structures submitted for editing, 59 structures were accepted and processed for splicing. For each structure, the query elements, path, ring, receptor contact, and pruned atoms were determined and recorded.

SPLICE required approximately 2 min of CPU time (SGI-380) to screen all bonds in the bond lattice and determine pairs of matched structures. From the 59 structures, a total of 3129 bonds were processed. Of these, 2174 were accepted for splicing based upon the 2.0 Å bond-inclusion distance constraint. With a predetermined minimum of five matching query elements, 353 structures were generated: 248 matched five elements, 76 matched six elements, 28 matched seven, and one matched eight. Many of these structures were redundant (see Discussion below). Using these structures as input, a second run produced two structures matching 10 query elements. Several are shown in Fig. 15. Sample output from the splice log is listed in Fig. 16.

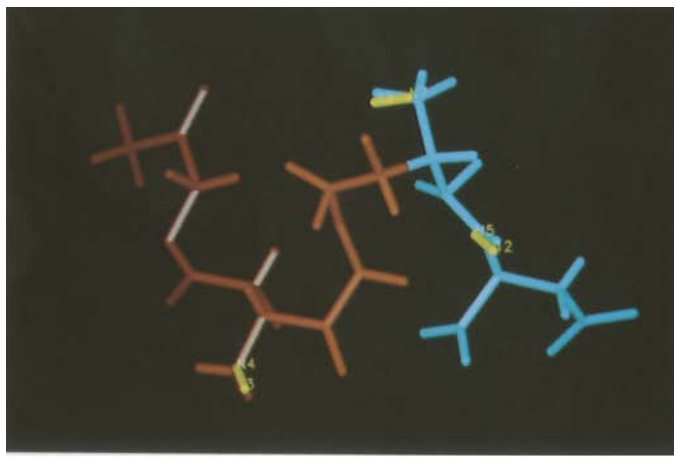
```
EDITING LOG

Query file =          query.mmm
Number of atoms:      15

Receptor file =       lock_small.mmm
Number of atoms:      187

PROCESSING OF INDIVIDUAL SUBSTRUCTURES:

acenht.mol2          Atoms: 37 Bonds: 38
  Query elements:    [ 9] 3   [ 6] 4   [ 7] 7   [13] 23
  Path atoms:        1  3  4  5  6  7  8  9 10 11 12 13 14 23
  Ring atoms:        1  5  8  9 10 16 17 18 19 20 21
  Contact atoms:
  Structure is accepted!
  Clipped atoms:

ackynu.mol2          Atoms: 66 Bonds: 66
  Query elements:    [ 2] 33   [ 1] 34   [ 4] 35   [ 3] 64
  Path atoms:        19 20 21 22 23 24 25 26 27 28 29 33 34 35 36 64
  Ring atoms:        1  2  3  4  5  6 19 20 21 22 23 24
  Contact atoms:
  Structure is accepted!
  Clipped atoms:     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 37 38
                     39 40 41 42 43 44 45 46 47 48 49 50 65

alalno.mol2          Atoms: 40 Bonds: 39
  Query elements:    [ 6] 3   [ 7] 14   [ 5] 15   [13] 28   [12] 31
  Path atoms:        3  14  15 24 25 28 31
  Ring atoms:        16 17 18 19 20 21
  Contact atoms:
  Structure is accepted!
  Clipped atoms:     1  39

atetcy10.mol2        Atoms: 59 Bonds: 60
  Query elements:    [ 7] 5   [ 4] 33   [13] 39   [ 3] 53
  Path atoms:        5  10 12 13 14 15 16 17 18 19 20 21 22 23 24
                     25 26 27 28 29 33 39 53
  Ring atoms:        12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
                     27 28 29
  Contact atoms:     19 20 21 22 23 24 25 26 31 47 56
  Structure is rejected.

etc. . .

TOTAL NUMBER OF STRUCTURES READ:      131
NUMBER OF STRUCTURES ACCEPTED:        59
```

Fig. 14. Sample listing of edit log output.

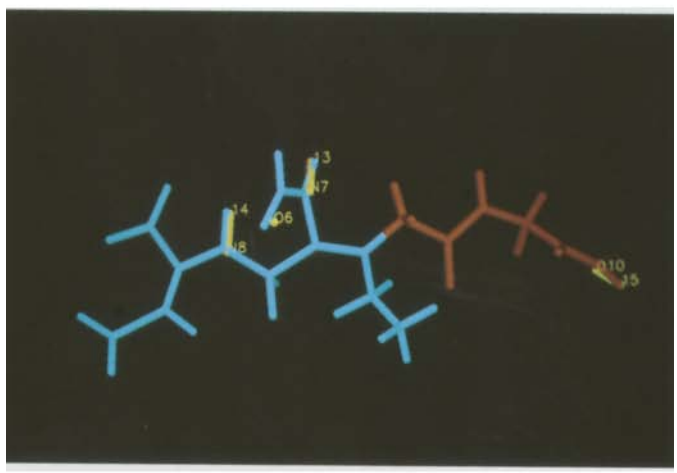
DISCUSSION

The use of database search and retrieval systems has become established in recent years. Since the inception of this technology, systems have been used to find potential lead compounds, providing inspiration as well as a foundation for further drug refinement. With improvements in technique and CPU power, searches have become more flexible, employing user-defined constraints to ensure the retrieval of the most useful structures. However, as efficient as these

a



b



c

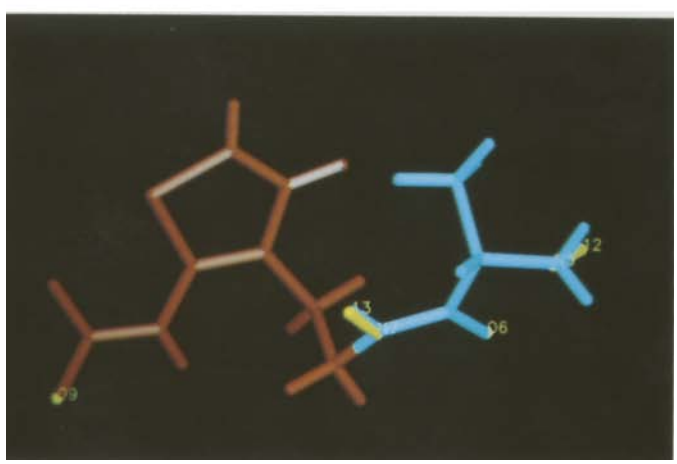
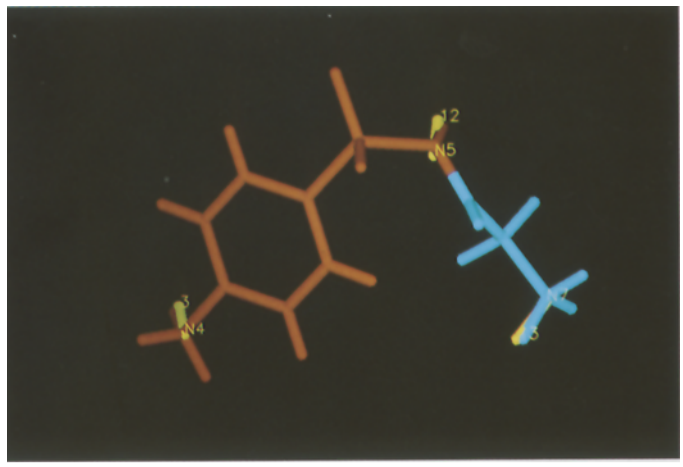
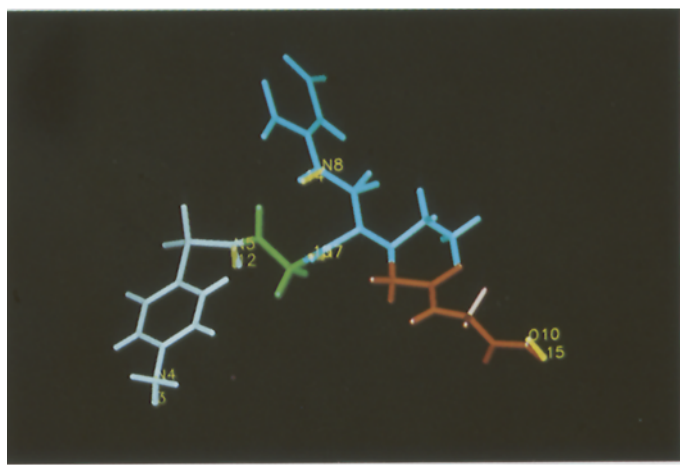


Fig. 15. Selected spliced structures.

d



e



f

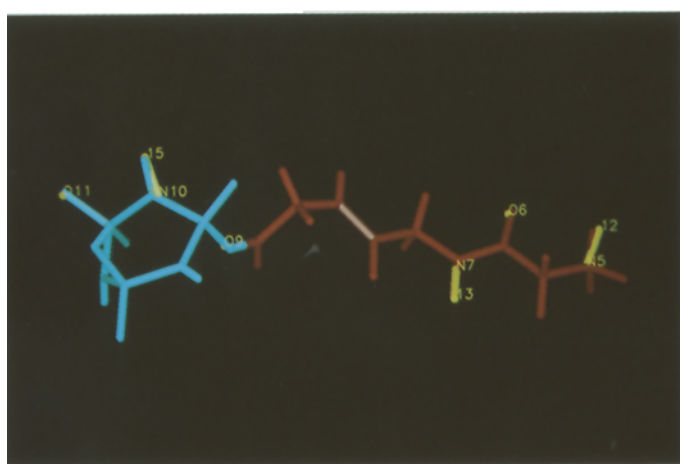


Fig. 15 (continued).

SPLICE LOG

Query file = query.mmm
Number of atoms: 15

1	1.72820	6.81030	4.84730
2	1.13860	3.34980	4.31920
3	-5.25710	2.24610	5.41650
4	-4.54040	2.85070	5.56950
5	-3.64160	4.85040	-0.57760
6	-2.47360	3.08440	-2.08860
7	0.56260	2.98840	-0.78760
8	0.08040	5.38740	-2.50660
9	4.38240	-1.27060	-0.72360
10	5.21240	-3.37060	-1.09860
11	7.49440	-4.24160	-2.36260
12	-4.40510	4.34450	-0.97900
13	-0.16530	3.24430	0.09370
14	0.02960	5.46980	-1.48120
15	5.01730	-4.27160	-1.48600

Bond inclusion lattice created
Number of points: 110

PROCESSING OF INDIVIDUAL SUBSTRUCTURES:

acenht.edt Atoms: 37 Bonds: 38
Query elements: [9] 3 [6] 4 [7] 7 [13] 23
ackynu.edt Atoms: 33 Bonds: 33
Query elements: [2] 15 [1] 16 [4] 17 [3] 32
alalno.edt Atoms: 38 Bonds: 38
Query elements: [6] 2 [7] 13 [5] 14 [13] 27 [12] 30
bcpilg.edt Atoms: 60 Bonds: 60
Query elements: [6] 3 [9] 6 [7] 8 [13] 44
bervez.edt Atoms: 56 Bonds: 55
Query elements: [7] 4 [8] 6 [6] 18 [13] 28 [14] 30
bervez2.edt Atoms: 68 Bonds: 68
Query elements: [8] 4 [7] 14 [14] 33 [13] 38
bibruz.edt Atoms: 25 Bonds: 24
Query elements: [5] 1 [6] 4 [9] 13 [12] 14
bibruz2.edt Atoms: 25 Bonds: 24
Query elements: [8] 1 [10] 13 [14] 14 [15] 25
bizsos.edt Atoms: 93 Bonds: 93
Query elements: [4] 2 [7] 34 [3] 48 [13] 82
bopwuy01.edt Atoms: 25 Bonds: 25
Query elements: [5] 2 [7] 3 [12] 12 [13] 13
bopwuy012.edt Atoms: 25 Bonds: 25
Query elements: [7] 2 [5] 3 [13] 12 [12] 13
budxut.edt Atoms: 33 Bonds: 33
Query elements: [10] 4 [7] 6 [13] 25 [15] 33
butfur.edt Atoms: 41 Bonds: 40
Query elements: [4] 6 [3] 14 [7] 27 [13] 32
calxes20.edt Atoms: 27 Bonds: 26
Query elements: [5] 1 [6] 5 [7] 6 [12] 15 [13] 22
capzic.edt Atoms: 58 Bonds: 58
Query elements: [7] 5 [8] 6 [13] 33 [14] 34
carneo.edt Atoms: 43 Bonds: 44
Query elements: [7] 11 [6] 13 [9] 18 [13] 33
zzzkvu10.edt Atoms: 22 Bonds: 21
Query elements: [7] 2 [8] 3 [13] 9 [14] 11

Fig. 16. Sample listing of splice log output.

```
TOTAL NUMBER OF STRUCTURES =          59
TOTAL NUMBER OF BONDS SCREENED =      3129
NUMBER OF BONDS  CONSIDERED FOR SPLICING =        2174

MATCH  5 :    alalno.edt & bopwuy01.edt      ATOMS: 14-24   4-2
MATCH  5 :    diyzeq.edt & alalno.edt        ATOMS: 12-8    14-24
MATCH  5 :    diyzeq.edt & calxes20.edt      ATOMS: 12-8    1-2
MATCH  6 :    glyglp.edt & jaycer.edt        ATOMS: 15-2    13-4
MATCH  6 :    glytre02.edt & jaycer.edt      ATOMS: 8-2     13-4
MATCH  6 :    bibruz.edt & bcpilg.edt        ATOMS: 5-3     19-8
MATCH  7 :    bibruz.edt & bervez.edt        ATOMS: 5-3     10-4
MATCH  7 :    bibruz.edt & bervez2.edt       ATOMS: 5-3     3-14
MATCH  7 :    bibruz.edt & fusmip.edt        ATOMS: 5-3     25-8
MATCH  8 :    mghphe20.edt & hetaur10.edt    ATOMS: 41-11   13-3
```

Fig. 16 (continued).

programs have become, the rate-determining step in any drug design process remains the user's interpretation and manipulation of retrieved data.

With the advent of programs such as LUDI [8,9] and BUILDER [18], investigators are trying a new approach of combining pharmacophore subcomponents into novel, aggregate ligands. Unfortunately, this approach greatly increases the analytical burden placed upon the medicinal chemist. In essence, all fragments must be reviewed in the context of one another to determine combinations that will produce satisfactory constructions. Without tools such as BUILDER or SPLICE, this task alone renders such an approach too difficult and time-consuming to be effective.

Regardless of the structure-generating method employed, the building blocks themselves must possess steric and electrostatic complementarity with the receptor. The task of screening and editing hundreds of structures can be distracting, as well as time consuming. The attention of the medicinal chemist should be directed towards more creative tasks, for example, visualizing the appropriate chemical modifications needed to optimize a drug's efficacy, or judging the synthetic feasibility of a particular compound. By automating rule-based procedures that take up time, but do not require adaptive thinking, a significant portion of the user's attention is freed for more productive purposes. In addition, transparently performing such tasks diminishes user bias in the editing process. Since the algorithm requires structural overlap, interfragment contacts are inconsequential. However, the removal of selected fragments can abolish opportunities for assemblies that are not immediately apparent.

The automation of editing and ligand assembly is made possible by the characterization performed initially on each fragment. By categorizing the various components of each fragment

(i.e. ring, query, path, non-path, and receptor contact), we are able to implement rule-based procedures that make processing *consistent* and *easy to direct* through user-defined parameters.

Bond management is the most vital element of the program. Several key functions are performed. First, it serves the basis for generating novel structures by determining sites of overlap. Secondly, it references components containing desired query elements with available attachment sites, enabling the program to determine rapidly the best candidates to link. Furthermore, since all atomic coordinates and atom types are maintained, self-consistent ligands devoid of steric complications are ensured. Finally, it serves as a source of statistical information. With this knowledge, SPLICE both increases its efficiency and reduces the amount of noise by limiting structural duplication.

In our experience with SPLICE, one pitfall has become apparent. To implement the 'divide-and-conquer' approach, one requires components that satisfy subsets of the pharmacophore. If acceptable subsets are relatively small, numerous combinations of elements will be valid. As such, an overabundance of structures may be retrieved. As shown in our control experiment, 150 hits were generated from a database of only 50 structures. In essence, the majority of structures contained various groups of pharmacophoric elements in different orientations.

However, the bulk of these hits was very similar. If a particular configuration of query atoms is common (i.e. the heteroatoms of a carboxylic acid or peptide bond), then thousands of similar hits are possible. If the recovery of overwhelming numbers of hits prevents the searching of a fair number of structures, then desired, novel configurations of atoms might be missed. Likewise, the splicing of fragments that are alike will combinatorially generate an even greater number of ligands that resemble one another. Database retrieval systems designed to retrieve partial hits must address this problem. One solution may be to monitor the connectivities and atom types of the retrieved structures' path atoms. As such, duplicate structures could be screened out.

One improvement of SPLICE would be to grant each chemical component limited conformational flexibility. Clearly, the bonds connecting any query elements could not be twisted, as this would disrupt the pharmacophore. However, conformational flexibility could be permitted for a select percentage of the non-path atoms. This would give the non-path atoms of each component the ability to explore conformation space and enhance their probability of overlapping with other components.

Another obvious extension to SPLICE is to include bridging fragments to enhance the probability of linking fragments with excellent interaction with other subsites. A library of acceptable fragments could easily be screened for the ability to join two ligand fragments together, whose combined interaction scores with the receptor were above a certain threshold. Recursive database searches could also provide such fragments.

Through automated, rule-based editing and linking procedures, we have tried to eliminate as much burden on the chemist as possible. The goal of our program is to distill from hundreds of component fragments the most plausible combinations capable of producing valid ligands. In doing so, we boost the signal and eliminate much of the noise as well as the bulk of fragment processing. As such, the attention of the medicinal chemist can be fully directed towards the crucial issues regarding the chemical assembly of the ligands.

ACKNOWLEDGEMENTS

This work was supported by the National Institutes of Health (grants GM-24483 and AI-27302) and the Medical Scientist Training Program in the Division of Biology and Biomedical Sciences at Washington University (NIH Training Grant GM-07200).

REFERENCES

1 Martin, Y.C., J. Med. Chem., 35 (1992) 2145.
2 Martin, Y.C., Bures, M.G. and Willett, P., In Lipkowitz, K.B. and Boyd, D.B. (Eds.) Reviews in Computational Chemistry, VCH Publishers, Inc., New York, 1990, pp. 213–256.
3 Murrall, N.W. and Davies, E.K., J. Chem. Inf. Comput. Sci., 30 (1990) 312.
4 Fersht, A., Enzyme Structure and Mechanism, Freeman, New York, 1977.
5 Beddell, C.R., The Design of Drugs to Macromolecular Targets, Wiley, New York, 1992.
6 DesJarlais, R.L., Sheridan, R.P., Dixon, J.S., Kuntz, I.D. and Venkataraghavan, R.J., J. Med. Chem., 29 (1986) 2149.
7 Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R. and Ferrin, T.E., J. Mol. Biol., 161 (1982) 269.
8 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 61.
9 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 593.
10 Chau, P.L. and Dean, P.M., J. Comput.-Aided Mol. Design, 6 (1992) 385.
11 Chau, P.L. and Dean, P.M., J. Comput.-Aided Mol. Design, 6 (1992) 397.
12 Chau, P.L. and Dean, P.M., J. Comput.-Aided Mol. Design, 6 (1992) 407.
13 Allen, F.H., Kennard, O. and Taylor, R., Acc. Chem. Res., 16 (1983) 146.
14 Miranker, A. and Karplus, M., Protein Struct. Funct. Genet., 11 (1991) 29.
15 Ho, C.M.W. and Marshall, G.R., J. Comput.-Aided Mol. Design, 7 (1993) 3.
16 Lewis, R.A. and Dean, P.M., Proc. Roy. Soc. London Ser. B., 236 (1989) 125.
17 Lewis, R.A. and Dean, P.M., Proc. Roy. Soc. London Ser. B., 236 (1989) 141.
18 Lewis, R.A., Roe, D.C., Huang, C., Ferrin, T.E., Langridge, R. and Kuntz, I.D., J. Mol. Graphics, 10 (1992) 66.
19 Lewis, R.A. and Kuntz, I.D., In Karalainen, E.J. (Ed.) Scientific Computering and Automation, Elsevier, Amsterdam, 1991, pp. 117–132.
20 Cormen, T.H., Leiserson, C.E. and Rivest, R.L., Introduction to Algorithms, McGraw-Hill Book Co., St. Louis, 1991.
21 Tripos Associates, Inc., St. Louis, MO, U.S.A.
22 Rich, D.H., Green, J., Toth, M.V., Marshall, G.R. and Kent, S.B.H., J. Med. Chem., 33 (1990) 1285.
23 Suguna, K., Padlan, E.A., Smith, C.W., Carlson, W.D. and Davies, D.R., Proc. Natl. Acad. Sci. USA, 84 (1987) 7009.
24 Ho, C.M.W. and Marshall, G.R., J. Comput.-Aided Mol. Design, 4 (1990) 337.