The NEWLEAD Program: A New Method for the Design of Candidate Structures from Pharmacophoric Hypotheses[1]

Vincenzo Tschinke* and Nissim Claude Cohen

Research Department, Pharmaceuticals Division, CIBA-GEIGY Limited, CH-4002 Basel, Switzerland

Received March 31, 1993•

We have developed a computer program, called NEWLEAD, for the automatic generation of candidate structures conforming to the requirements of a given pharmacophore. The treatment consists in connecting the pharmacophoric pieces with spacers assembled from small chemical entities (atoms, chains, or ring moieties). We have tested the program on several sets of input fragments, each comprising selected functional groups obtained from the bioactive conformations of reference molecules. In addition to the expected solutions, the program can generate new structures that are chemically unrelated to the reference molecules. This provides an unbiased starting point for the design of new generations of lead structures. The concept used in this approach is presented and discussed. The present possibilities of the program are illustrated by some examples. The treatment is very fast, because only a few bonds are created between building blocks already having ideal geometries. The ability to generate rapidly a variety of molecules conforming to a three-dimensional pharmacophoric model makes NEWLEAD a useful tool with wide applicability in rational drug design, including the areas of molecular mimicry and peptidomimetism.

Pharmacophores are often considered in rational drug design as starting points for the conception of novel candidate structures. The pharmacophoric approach is compatible with the ideas of analogy and mimetism, the two approaches most often taken by the successive generations of drug designers.

A pharmacophore consists of the spatial arrangement of chemical groups that is recognized by a single receptor. The concept is general: it can be derived from the inspection of X-ray structures of macromolecule–ligand complexes or from known lead compounds. It is also creative: it can give access to the identification of novel structures having the desired biological activity, but which are chemically unrelated to the known reference compounds.

Designing molecules that assemble a set of disconnected functional groups in three dimensions is not a trivial task and remains very subjective. A computerized approach may provide a diverse set of solutions that are free from arbitrary user bias.

A variety of modeling techniques has been developed to derive pharmacophoric hypotheses by comparing series of bioactive molecules[2–7] or from the knowledge of the three-dimensional structure of a receptor.[8–10]

The next step consists of exploiting the adopted pharmacophoric hypothesis and designing new molecules conforming to the desired requirements. Current efforts are focused on the search[11–21] in three-dimensional databases for molecules conforming to specific pharmacophoric patterns. These methods have been recently reviewed.[22] The databases currently used by these programs contain either experimental X-ray structures, e.g. the Cambridge Structural Database,[12] or three-dimensional structures derived from modeling programs.[23]

The database search methods have been remarkably successful. Recently, more direct and perhaps more creative approaches have been conceived that no longer rely on a given database[9,10,24–32] (for a review see ref 32). Along these lines we developed the NEWLEAD program, described here.

Method

The computational approach of NEWLEAD can be simply described as follows. Given a set of at least two disconnected fragments, the problem consists of finding appropriate spacers to assemble these moieties in a single molecule. Multiple solutions are likely, because many connections with different combinations of spacers are possible. The elementary units used for constructing the spacers are library spacers (Figure 1), single-atom spacers (Figure 2), and fuse-ring spacers (Figure 3). The procedure adopted by NEWLEAD executes the following sequence of commands: (1) connect two isolated moieties, (2) retain the intermediate solutions for further processing, (3) repeat the above steps for each of the intermediate solutions until no disconnected units are found, and (4) output the final solutions, each of which is a single molecule.

The method is outlined in Figure 4. The reader is referred to the experimental Section (program implementation) for a detailed description of the code. The input of the program, the spacers, the algorithms, and the output obtained are described in the next paragraphs.

Structure of the Input. The input consists of the pharmacophore, given as a set of disconnected fragments in three dimensions. A typical input can be built by identifying key fragments of a reference molecule in its assumed bioactive conformation, eliminating the remaining part of the structure, and adding H-atoms to the chosen fragments (by convention, the program will attach spacers only on atoms explicitly carrying hydrogen atoms in the input data).

If the structure of the receptor is known, sterically forbidden regions can be mimicked by introducing additional dummy fragments (no hydrogens). An input can also be created by putting trial fragments in positions complementary to the binding sites of a known receptor.

Before starting a calculation, the user can also define a set of tolerance factors for the geometrical and steric

• Abstract published in *Advance ACS Abstracts,* October 15, 1993.

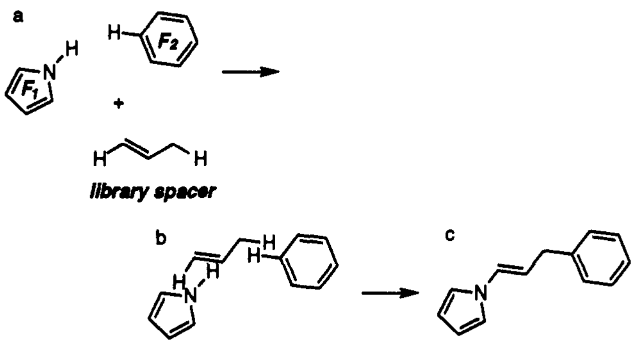

Figure 1. Elementary operations in the NEWLEAD program: library spacers. Two fragments, F_1 and F_2, are connected by a spacer. The connection is made by replacing a hydrogen atom on each fragment with a heavy atom of the spacer. (a) A spacer is selected from the library on the basis of geometrical criteria. (b) The spacer is superimposed on the two fragments. (c) The bonds are created.

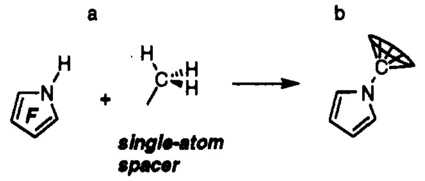

Figure 2. Elementary operations in the NEWLEAD program: single-atom spacers. (a) A fragment F and a single-atom spacer (in this case, CH_3). (b) The single-atom spacer is attached to the fragment and rotated several times around the connection bond (see text).

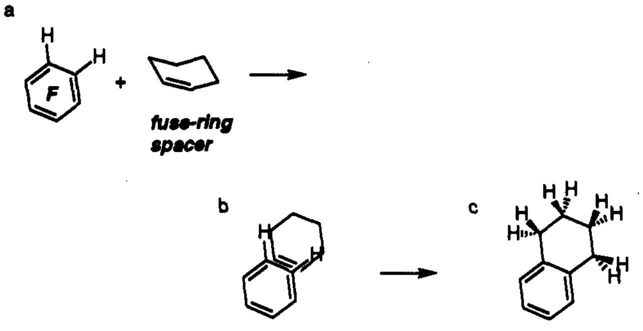

Figure 3. Elementary operations in the NEWLEAD program: fuse-ring spacers. (a) A fragment with two C–H bonds available for connections and a matching fuse-ring spacer. (b) Alignment of the two moieties. (c) The bonds are created.

requirements of the spacers, as described in the following sections. The values of these parameters are stored in a special file that can be modified before the treatment to obtain either a broad range of solutions or a smaller set.

Spacers. The program uses three types of spacer: library spacers, single-atom spacers, and fuse-ring spacers (see Figures 1–3). The combination of these elementary building blocks generates an essentially infinite number of spacers in the geometrically correct conformations to connect any two fragments. All the spacers are treated as rigid, and the proper conformations are given by the free selections of the dihedral angles around the bonds used for the connections.

The library spacers and the fuse-ring spacers are simple molecules stored in two libraries of Cartesian-coordinate files. Additions to the libraries can be made without the need to change the program code. As the name implies, the single-atom spacers are short connections. They are generated by simple geometric operations on the fragments. The library spacers are used to connect two fragments. The single-atom spacers and the fuse-ring spacers are attached to a single fragment and used as starting points for a connection to another fragment with a library spacer.

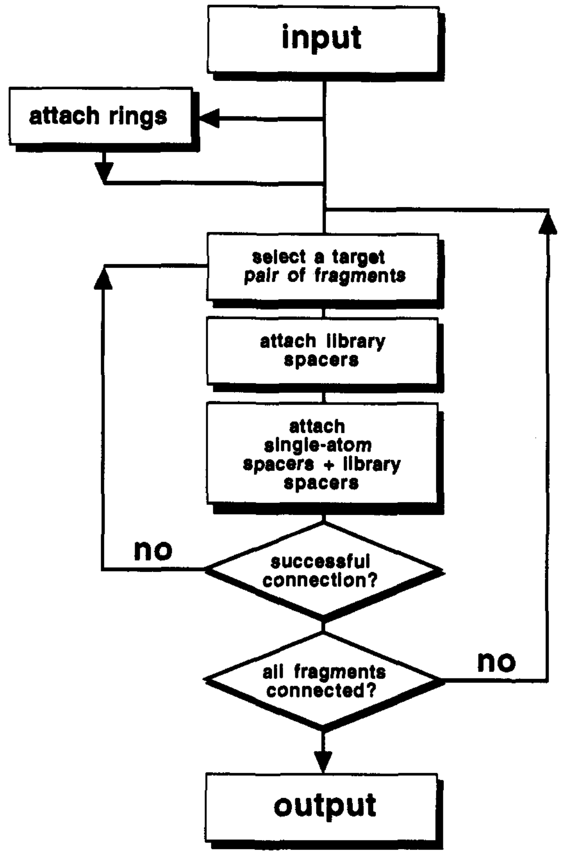

Figure 4. The basic algorithm of the NEWLEAD program.

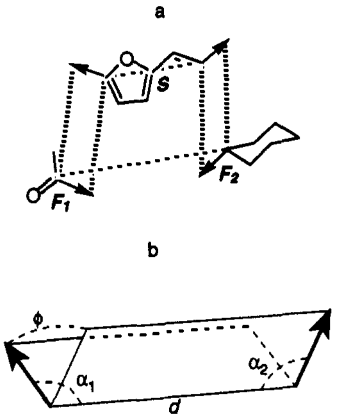

Figure 5. Representation of spacer connections as vector pairs. (a) A library spacer S, two fragments F_1, F_2, and the two vector pairs indicating their connection sites. (b) The four geometrical parameters of a vector pair: d, α_1, α_2, and ϕ.

Algorithm and Library for Spacers. The library spacers are optimized structures of small molecules such as ethylene, benzene, and methylamide. Details are given in the Experimental Section.

To attach a spacer, the program uses two bonds on the fragments and two bonds on the spacer (see Figure 1). For the sake of code implementation, these bonds are represented as vectors, following the method of Bartlett[17] (see Figure 5a). A vector tail is given by a heavy atom and a vector head by an H-atom. As depicted in Figure 5b, a bond pair is described in terms of (1) the distance d between the two heavy atoms (vector tails), (2) the two angles α_1 and α_2, and (3) the dihedral angle ϕ.

Most spacers can give more than one connection. Ethylene, for instance, can give two, corresponding to the *cis* and *trans* conformations, and benzene the three connections resulting from *ortho*, *meta*, or *para* disubstitution of the ring. The geometrical parameters of all the connections of the spacers have been calculated and stored in a key file. Currently, the library contains 68 spacers (see the Experimental Section), and the corresponding key file contains 517 entries, with values of the parameter d ranging from 1.31 to 7.67 Å.

Connection points on the fragments are also defined in terms of pairs of vectors (see Figure 5a). The algorithm to connect two fragments with a spacer is as follows. The program builds a list of vector pairs representing connection points on two fragments. For each vector pair on two fragments, the program can rapidly scan the library and find one or more spacers with matching vector pairs. The user can define tolerance values for the parameters d, α_1, α_2, and ϕ (smaller tolerance values give fewer results of higher quality; larger tolerance values give more results, but a number of them may have distorted geometries). We usually take the tolerance for the distance d in the range 0.2–0.4 Å. No simple relationship exists between this value and the actual errors on bond distances, which are usually much smaller (typically less than 0.05 Å). The error is reduced because it is distributed between two bonds and also because these two bonds are usually bent away from the segment d. The tolerances on the angles are usually set around 10°. The actual errors on valence and torsion angles are smaller (typically less than 5°), again due to the distributio.ı of the error over two bonds.

The spacers selected from the library are superimposed by the program on the fragments (Figure 1b). A spacer is accepted after an overall van der Waals radii check. The allowed distances are given by the van der Waals radii sums for any pair of atoms multiplied by a user-defined scaling factor, typically chosen between 0.7 and 0.9. Spacer connections that would involve undesired bonds such as O–O and N–N are also rejected. Finally, each spacer that satisfies the geometrical, steric, and chemical requirements is permanently connected to the fragments, leading to a new molecular moiety (see Figure 1c).

No constraints are set on the dihedral angles of the new bonds between the fragments and a spacer. For instance, suppose that the two fragments RH and R′H are connected by an ethylene spacer to form the molecule RCH═CHR′. The torsion angle of the double bond can be either 0° or 180° (library spacer), but the torsion angles of the bonds R–C and R′–C can assume any value, depending on the relative orientation of R and R′.

Algorithm for Single-Atom Spacers. The single-atom spacers are represented by the groups CH_3, NH_2, and OH. The program uses these units in conjunction with the library spacers. The corresponding algorithm is described as follows. Given a pair of fragments, the program replaces one H-atom on each fragment in turn with the groups CH_3, NH_2, and OH (see Figure 2). The program finds the vector pairs involving the single-atom spacer and the other fragment and attempts the connections with library spacers. The library search, the superimposition, and the van der Waals radii check are carried out as above. The attempt is repeated for several orientations of the single-atom spacers (we usually define between 4 and 12 orientations for each group). The same operations are repeated for all the remaining hydrogens on the fragments.

Algorithm and Library for Fuse-Ring Spacers. As an option, the program can fuse rings on suitable input fragments (see Figure 3). For this purpose we use the "fuse-ring" spacers. They consist of optimized structures of five- to seven-membered rings stored in a library of Cartesian-coordinate files. At present, rings are fused only on double bonds. For example, a benzene fragment can be transformed into a benzocycloalkane, naphthalene, indole, etc. Once fused, the rings are used for further connections involving library spacers with or without single-atom spacers. In summary, the most complex spacer created by the program involves fuse-ring, single-atom, and library spacers.

Currently, the fuse-ring library contains only six molecules (see Experimental Section). As with the library spacers, we use vector pairs to describe the bonds to be used in the connections. The values for the corresponding parameters d, α_1, α_2, and ϕ are stored in a special key file. The fuse-ring spacers are attached to a fragment by using the same algorithm and program code as for the library spacers. For each fragment the program builds a list of pairs of bonds that can be used to accommodate a fuse-ring spacer (see Figure 3a). Then the fuse-ring library is searched for matching vector pairs. Each fuse-ring spacer found is superimposed on the fragment and, after being checked for van der Waals radii violations, permanently connected (see Figure 3b,c). For these operations the program applies the same user-defined tolerance values (for the library search and the van der Waals scaling factor) as those set for the library spacers.

Analysis of the Output. The output produced by NEWLEAD is a set of molecules containing the original fragments now connected by spacers. The atoms belonging to the input fragments maintain their original orientations in space. The molecules are chemically plausible because of the makeup of the spacers and energetically acceptable because of the rejection of solutions with van der Waals radii violations.

The results are ordered by increasing van der Waals radii violations. Because several hundred solutions may be generated in a typical run, we found it useful to cluster them into groups having similar topological features.

In our laboratory, NEWLEAD is used by both molecular modelists and synthetic chemists. Usually the input for the program is defined during a common session. Any interesting molecular frames identified among the output are either used as queries for in-house database searches or considered directly for synthesis. Minor modifications to the selected prototypes may, of course, be considered for synthetic reasons. In some cases, additional modeling analyses are made (energy minimization, docking analysis, etc.) before the molecule is considered an interesting enough candidate to be prepared and evaluated.

Results

To illustrate the possibilities afforded by the NEWLEAD program, we have selected the following molecules: methotrexate, indomethacin, and the HIV-1 protease inhibitor A74704. The program was able to reproduce the reference structures. As expected, it also generated other structures that were chemically unrelated to the references. The tolerance factors used for each test are reported in the Experimental Section.

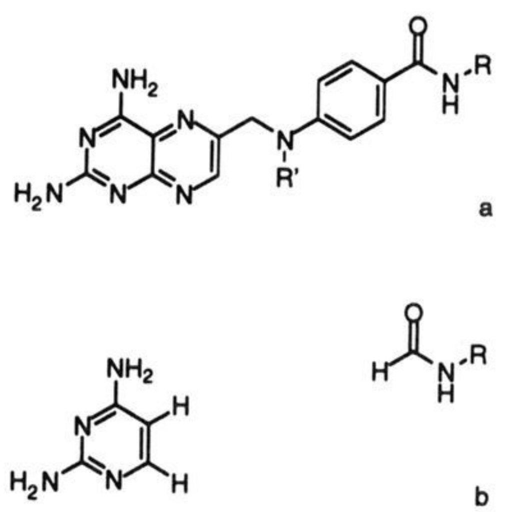

Figure 6. (a) Methotrexate molecule (R′ = CH_3); R = $CH(CO_2H)CH_2CH_2CO_2H$. (b) Input for the NEWLEAD program. In part a, the molecule with R′ = H corresponds to one of the solutions generated by the program.

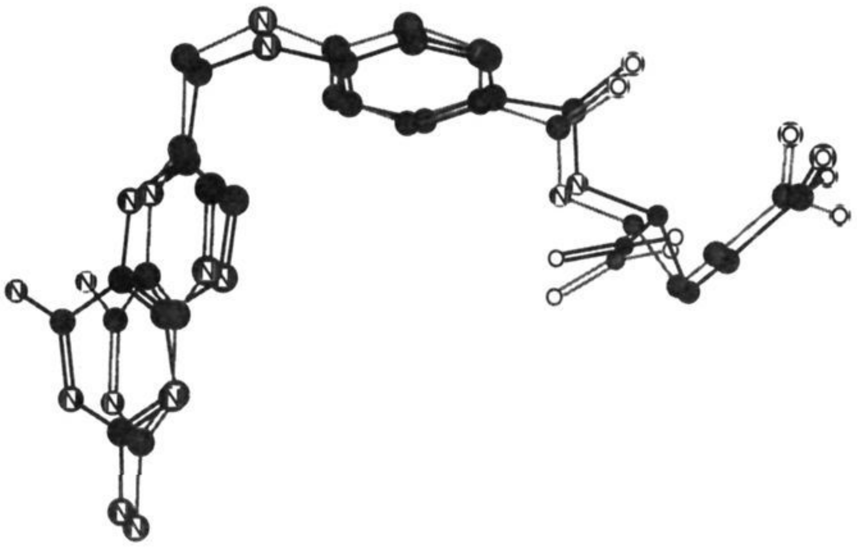

Figure 7. Comparison between a NEWLEAD result (dark grey) and the same structure after minimization (light grey). The molecule shown is the NEWLEAD result of Figure 6a (R′ = H) for the methotrexate run.

Methotrexate Test. Starting from an X-ray structure[33] of the methotrexate molecule (see Figure 6a), we removed the fused pyrazine ring, the phenyl group, and the atoms connecting these two aromatic rings. The input therefore consisted of two fragments (Figure 6b). The NEWLEAD program was run with the fuse-ring option on. The reference structure was reproduced (with one minor difference, NH instead of NCH_3; see Figure 6a). An analysis of the calculation showed that for this solution the program first attached the pyrazine ring taken from the fuse-ring library. A methyl single-atom spacer was then attached to the pyrazine, and the aniline spacer completed the connection. A comparison between this result and the same structure after minimization is shown in Figure 7. The total number of structures generated in this run was 21. Some of them are depicted in Figure 8. The results vary from structures closely related to the reference compound (see Figure 8a, where the toluene spacer replaces aniline) to rather different structures (see Figure 8b–h). In Figure 9 we depict several of these structures in three dimensions, as they were obtained from the output. The CPU time for this run was 47 s on a VAX 6320, and 5 s on a Silicon Graphics Iris 4D 310 GTXB workstation.

Indomethacin Test. For the second example we built a three-fragment input from the X-ray data of the indomethacin structure[34] (see Figure 10). It consisted of the CH_3OH, HCO_2H, and HCl fragments, set in the positions that the corresponding functional groups occupy in the reference molecule. NEWLEAD reproduced the reference structure (molecule with R = H of Figure 10a) almost identically. For this solution, a methyl single-atom spacer was first attached to the C-atom of the formic acid fragment. The indole spacer was used for the connection to the methanol fragment. In the second and last pass, the benzaldehyde spacer was added. Altogether 68 new structures were obtained in this run. Some of them are depicted in Figure 11. While a few structures are related to the reference (see, for instance, Figure 11a), most of them are quite different (see Figure 11b–d). This run took 2 min of CPU time on the VAX 6320, versus 12 s on the Silicon Graphics Iris 4D 310 GTXB workstation.

HIV-1 Protease Inhibitor Test. The third example involved the HIV-1 protease inhibitor A74704.[35] From the X-ray structure of this compound (inhibitor bound to the HIV-1 protease enzyme), we built an input for NEWLEAD consisting of five fragments: two benzene molecules, two moieties each containing a peptide bond, and one water molecule (see Figure 12b). Among other results, the program exactly reproduced the reference structure. An analysis of the calculation showed that this result was generated as follows. First, a methyl single-atom spacer was attached to the water fragment, then two ethane spacers were used to join the two benzene fragments. Finally, two methylamide spacers completed the connection. Eleven more structures were generated by the program in this run. They are all rather closely related to the reference, but that is not surprising, because the short distances left between the chosen fragments limits the number of possibilities for connection with spacers. One of the solutions includes two ethene spacers instead of the amide spacers, resulting in double bonds instead of peptide bonds. The CPU time for this run was 19 min and 30 s on the VAX 6320 and 1 min and 48 s on the Silicon Graphics Iris 4D 310 GTXB workstation.

Discussion

For the three test cases, the NEWLEAD program was able to generate results that were either similar to the reference molecules or different in structure. This demonstrates the validity of the program as well as its predictive value.

The geometrical quality of the structures is under the control of the user and depends on the tolerance factors allowed for the spacer search. Small tolerance values ensure structures with bond distances and angles close to ideal values. The user can control the energetics of the solutions by choosing the van der Waals radii scaling factor. As previously mentioned, we do not constrain the torsion angles at the junctions between fragments and spacers. Although some structures may have values of these angles different from the known equilibrium positions, we feel that, as long as the van der Waals criterion is satisfied, the energy of the proposed molecules should remain acceptable.

Approaches for Structure Generation in Rational Drug Design. The existing computer programs for structure generation in drug research[9,10,17,24–32] use two main types of treatment to build candidate structures. The first involves the use of single atoms as building blocks[24–27] and the second the use of larger chemical elements, or templates.[9,10,17,28–32]

The programs[24–27] in the first category generate structures by attaching one atom after the other, starting from a "seed" atom. Several positions of each new atom are considered, leading to multiple solutions.

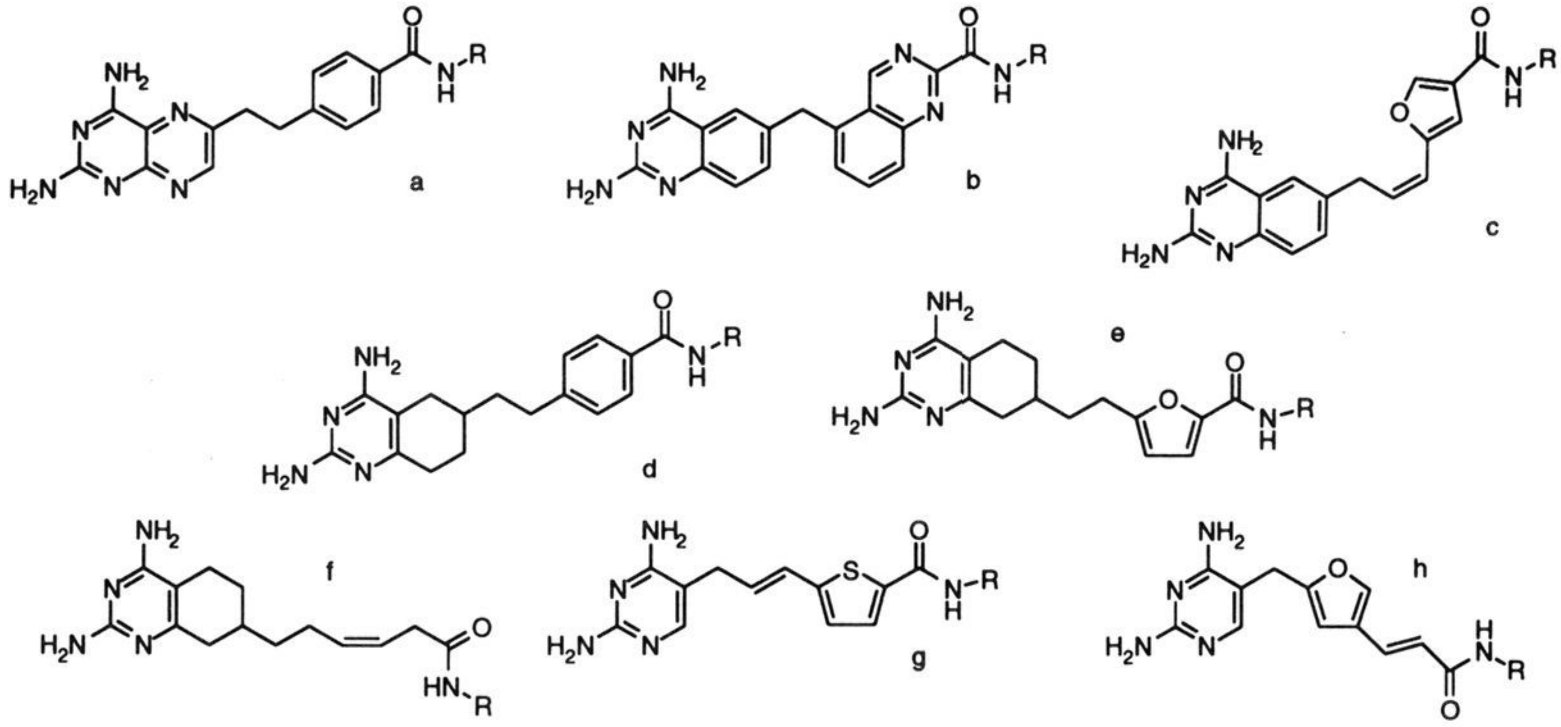

Figure 8. (a–h) Selected results generated by the NEWLEAD program from the methotrexate input of Figure 6b (R = $CH(CO_2H)CH_2CH_2CO_2H$).

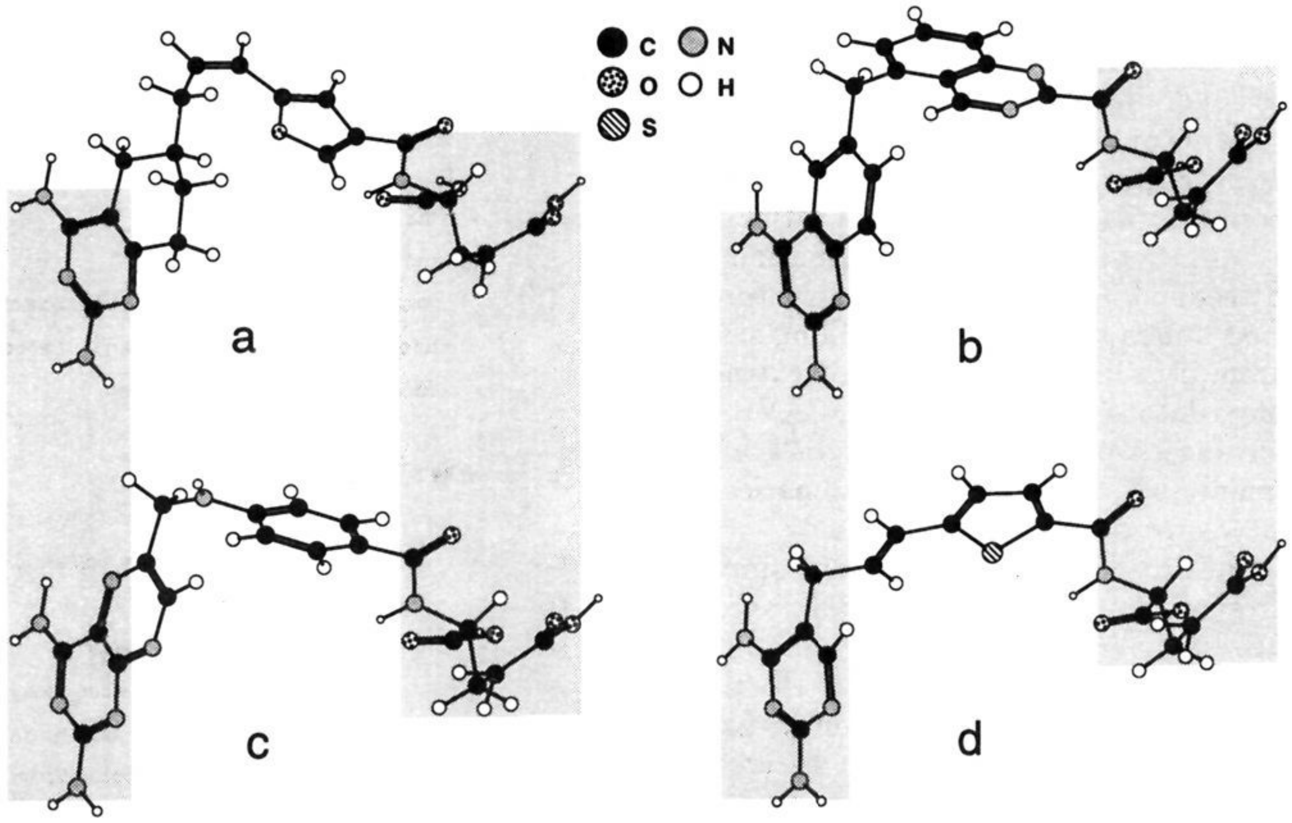

Figure 9. Several NEWLEAD results from the input of Figure 6b. Shaded areas indicate schematically the fragments given in the input, which are present in the same orientations in all NEWLEAD results. The corresponding 2-D representations of the above molecule are shown in Figures 8g (a), 8b (b), 6a (c, R′ = H), and 8c (d).

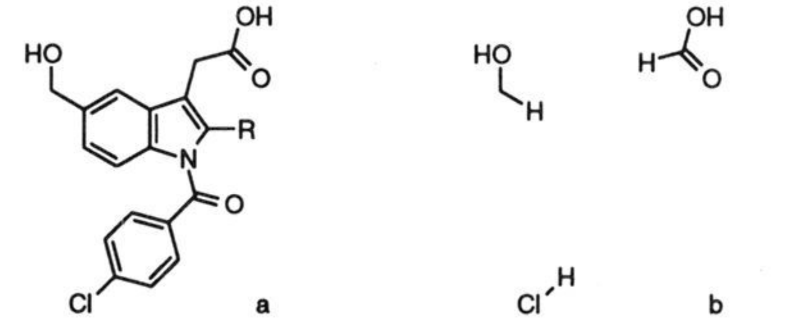

Figure 10. (a) Indomethacin molecule (R = CH_3). (b) Input for the NEWLEAD program. In part a, the molecule with R = H corresponds to one of the solutions generated by the program.

CAVEAT[17] and LUDI[10] are examples of programs that build structures by using chemical templates.[9,10,17,28–32] CAVEAT attempts to find spacers that simultaneously satisfy multiple connections on isolated fragments. The bonds chosen for connections on the fragments are represented as vectors. The program rapidly finds suitable spacers from a subset of the Cambridge Structural Database[12] containing thousands of polycyclic structures, also represented as sets of vectors. The results of a CAVEAT run are new molecules that combine the fragments and the spacers and therefore differ from the original database entries. This feature distinguishes CAVEAT from other 3D database search programs used for design.[13–16,18,20–21] The LUDI program builds ligands inside a receptor cavity. It places fragments from a library of 800 small molecules at favorable interaction sites inside the cavity. A second library of 1100 molecules contains linkers, similar to the spacers of the present approach. The linkers can be connected to one, two, or three fragments already placed in the cavity to create more complex ligands.

The present program is more similar to the chemical template methods than to the one-atom-at-a-time building approach, because it relies on predefined libraries of spacers and fused rings. Nevertheless, our use of the single-atom spacers is similar in spirit to the latter methods. NEWLEAD attaches single atoms in many orientations

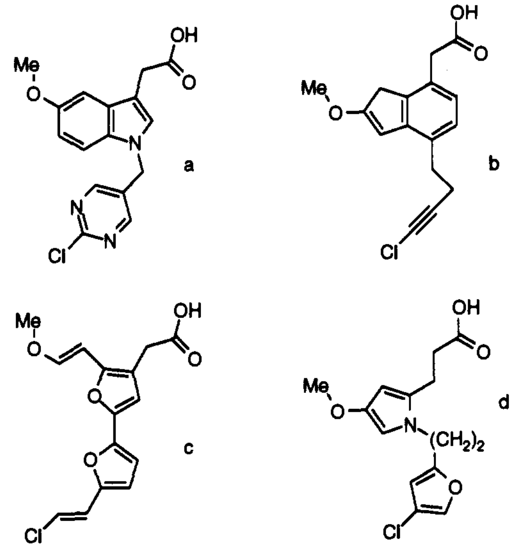

Figure 11. (a–d) Selected results given by the NEWLEAD program from the indomethacin input (Figure 10b).

on a given fragment. The next step taken is the connection of a spacer from the library rather than of another single atom.

An important point of contact between NEWLEAD and CAVEAT is that both programs rely entirely on the careful choice of pharmacophoric pieces in precise orientations in space (vector representation) to carry the information on biological activity needed to create candidate structures. The present method builds spacers by assembling small chemical blocks from libraries, while CAVEAT extracts the spacers from large databases. The input for CAVEAT consists of the coordinates of vectors defining anchoring points on the fragments, while NEWLEAD automatically considers different vector pairs on the fragments.

The linking functionality of the LUDI program is similar to the approach presented here, especially in the creation of short links between two fragments. The programs differ somewhat in the creation of long connections. Because the input fragments for NEWLEAD are given in fixed positions in space, our program is designed to create complex spacers composed of several elements for long connections (e.g. in the methotrexate test, above). In LUDI, the difference between fragments and linkers is more subtle. To create a ligand spanning a long distance inside a receptor cavity, LUDI attaches linkers on single existing fragments and simultaneously places new fragments inside the cavity. When these growing structures become close enough, they can be linked by the program into a unit.

Enhancements to NEWLEAD. Future enhancements to NEWLEAD will include extensions of the fuse-ring treatments, hydrogen-bond anchoring, and possible motions of the input fragments.

Conclusions

The NEWLEAD computer program for the automatic generation of lead structures is able to assemble disconnected pharmacophoric fragments in three dimensions and to generate multiple solutions that are chemically unrelated to each other. The solutions are likely mimic compounds of the active molecules used to define the pharmacophore, because they have the functional groups considered responsible for the biological activity in the same relative orientation in three dimensions. The program is expected to be a useful tool with broad applicability in rational drug design, including the field of peptidomimetism.

Experimental Section

Molecular Modeling. Structure Optimization. The structure of Figure 7 was energy-minimized using MACROMODEL[36] and the NCC force field.[37] The superposition of the two structures of Figure 7 was carried out using the all-atom rigid superposition of MACROMODEL. The library spacers and the fuse-ring spacers (see below) were also energy-minimized using MACROMODEL and the NCC force field.

Tolerance factors for methotrexate: 0.15 Å for the distance d, 15.0° for the angles α_1, α_2, and ϕ, and 0.75 for the van der Waals radii scaling factor.

Tolerance factors for indomethacin: 0.25 Å for the distance d, 10.0° for the angles α_1, α_2, and ϕ, and 0.45 for the van der Waals radii scaling factor.

Tolerance factors for the HIV-1 protease inhibitor A74704: 0.4 Å for the distance d, 20.0° for the angles α_1, α_2, and ϕ, and 0.7 for the van der Waals radii scaling factor.

Program Implementation. The implementation of NEWLEAD can be described with reference to the pseudocode given below and to the flow chart in Figure 4. For the sake of discussion, two major sections can be defined, the main program and the "attach_spacers" function. We shall describe the main program first and then, in more detail, the attach_spacers function. The main program follows:

```
1   main {
2       read the original input stream
3       if the ring option is selected {
4           create a list list_1 of bond pairs that can accommodate fuse-rings
5           attach_spacers(list_1, ring_library)   /* attach fuse-ring spacers */
6           add the solutions to the input stream
7       }
6       while the input stream is not empty {
9           for each set of fragments in the input stream {
10              build a list of fragment pairs, by increasing distance
11              for each fragment pair {
12                  build a list list_2 of bond pairs
13                  attach_spacers(list_2, spacer_library)
14                  attach single-atom spacers, build a new list list_3 of bond pairs
15                  attach_spacers(list_3, spacer_library)
16                  if one or more spacers are found, exit loop
17              }
18              write the intermediate solutions to the input stream
19              write the final solutions to the output stream
20          }
21      }
22      sort the solutions by increasing VdW radii overlaps
23      order the solutions with similar spacers in groups; write report
24  }
```

Description of the Main Program. On line 2 of the main program, the input as supplied by the user is read into the input stream. On lines 3–7, we have a section that performs the fusion of rings on suitable fragments (see Figures 3 and 4). All the pairs of bonds suitable for fusing a ring are added to a list, *list_1*. *list_1* is passed to the attach_spacers routine, with an indication that the library of fuse rings, *ring_library*, is to be used. The solutions given by attach_spacers are new sets of fragments with fused rings; they have the same number of fragments as the original input sets from which they are derived. These solutions are added to the input stream for further processing.

The loop of lines 8–21 comprises the body of the NEWLEAD code. The loop processes the current input stream. If needed, it creates a new input stream with the intermediate solutions to be processed in the next pass. The loop is terminated when no

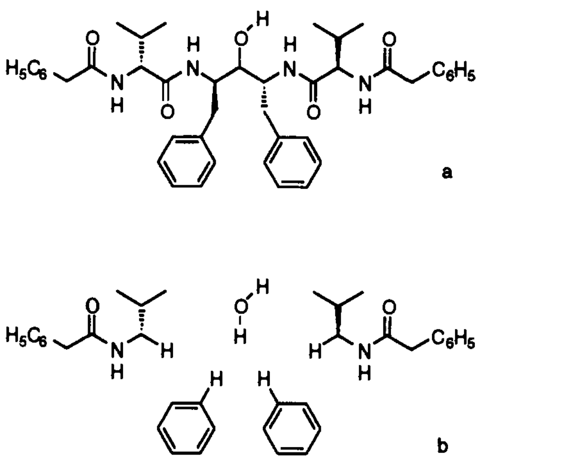

Figure 12. (a) A74704 molecule. (b) Input for the NEWLEAD program.

more structures are found in the current input stream. The input stream may contain only one set of fragments at the beginning on a program run, or even several hundreds of sets in an intermediate pass.

The program processes each set of fragments in the input stream, lines 9–20. The first operation on a set of fragments, line 10, is to build a list of all the fragment pairs (one or more). This list is ordered, by increasing distances between the fragments, to favor shorter connections. A loop, lines 11–17, processes the fragment pairs. In the case of more than two fragments, the loop is terminated, line 16, as soon as one pair of fragments can be connected by one or more spacers (Figure 4). This is to avoid the occurrence of duplicate results when the intermediate solutions are processed in the next pass.

For a given pair of fragments, the program attempts to find spacers, lines 12–15. On line 12, the program builds a list, *list_2*, of the bond pairs shared by the two fragments. On the next line, *list_2* is passed to the attach_spacers routine, with an indication to use the default spacer library, *spacer_library*. This operation may yield the first solutions. To get more solution, lines 13 and 14 are used. Single-atom spacers are attached to the fragments (Figures 2 and 4). A new list, *list_3*, contains pairs involving one bond on a single-atom spacer and one bond on the other fragment. A call to attach_spacers with arguments *list_3* and *spacer_library* attempts to find more connections between the two fragments, line 15.

The program handles the solutions found for a given set of fragments on lines 18 and 19. On line 18, the intermediate solutions, which need more spacer connections, are entered in the new input stream to be processed in the next pass. The final solutions are written to the output stream, line 19.

The program reaches line 22 after the input stream is fully processed. The final solutions are ordered by increasing van der Waals radii solution. On line 23, the program attempts to group together the solutions that have similar spacers. A report is written giving the number of solutions and describing the makeup of each of them in terms of spacers.

Below, we give the pseudocode for the attach_spacers function:

```
1   attach_spacers(list, library) {
2     for each bond pair in list {
3       from library, find spacers with matching bond-pairs, within tolerance
4       for each hit {
5         attempt the atom superposition of the hit on the fragment(s)
6         check the Van der Waals radii errors
7         if the violation is higher than the threshold, skip to the next entry  /* reject */
8         make permanent bond connections to the fragment(s)
9       }
10    }
11  }
```

Description of the attach_spacers Function. The attach_spacers function is at the heart of the NEWLEAD program. It is flexible enough to be used to fuse rings to a fragment and to join two fragments with a library spacer. As shown on line 1, the input for attach_spacers is a list of bond pairs, *list*, and a library of spacers, *library*.

At lines 2–10, the program processes each pair in *list*. At line 3, several hits are found in *library*, based on the geometrical matching of the bonds on the fragment(s) and on the spacers. Next, the program attempts the fitting in place of each hit (lines 4–9). A standard superposition algorithm is used to align the bonds of the spacer to the bonds of the fragment(s) (Figures 1b and 3b). If the spacer can be superimposed successfully, the van der Waals radii errors are calculated, line 6. The calculation involves only the atoms that would be part of the final structure connected to the fragment(s). On line 7, a spacer is accepted if no two atoms are closer than the sum of the van der Waals radii times a user-defined factor.

A spacer that passes the van der Waals radii test can then be connected to the fragment(s), line 8. Several atoms are deleted from the spacer and the fragment(s), and two new bonds are created (see Figures 1c and 3c).

Library spacers: ethane, ethanol, ethylamine, ethanethiol, acetaldehyde, propane, propanol, butane, pentane, hexane, heptane, ethylene, methylamide, propene, ethanoic acid, ethylamide, methylethylester, *trans*-2-butene, *N*-methylethylamide, benzene, pyridine, pyrimidine, pyrazine, pyridazine, toluene, benzaldehyde, 2-methylpyrimidine, 4-methylpyridine, 5-methylpyrimidine, aniline, styrene, benzamide, ethynylbenzene, cyclohexane, methylcyclohexane (equatorial), pyrrole, pyrazole, indazole, furan, thiophene, *N*-methylpyrrole, 2-methylpyrrole, 2-methylfuran, 2-methylthiophene, 2-ethenylpyrrole, 2-ethenylfuran, 2-ethenylthiophene, 3-ethenylpyrrole, 3-ethenylfuran, 3-ethenylthiophene, naphthalene, quinoline, isoquinoline, phthalazine, cinnoline, quinoxaline, quinazoline, pteridine, 1-8-naphtyridine, 1-methylnaphthalene, 2-methylnaphthalene, 2-naphthaldehyde, indole, indazole, benzimidazole, purine, benzofuran, benzothiophene.

Fuse-ring spacers: pyrrole, cyclopentene, cyclohexene, benzene, pyrazine, cycloheptene.

References

(1) Preliminary results of this work were presented at the XIIth International Symposium on Medicinal Chemistry, Basel, Switzerland, September 13–17, 1992; Abstract OC-08.3.

(2) (a) Marshall, G. R.; Barry, C. D.; Bosshard, H. E.; Dammkoehler, R. A.; Dunn, D. A. The Conformational Parameter in Drug Design: The Active Analog Approach. In *Computer-Assisted Drug Design*; Olson, E. C.; Christoffersen, R. E., Eds.; ACS Symposium Series 112; American Chemical Society: Washington, DC, 1979; pp 205–226. (b) Mayer, D.; Naylor, C. B.; Motoc, I.; Marshall, G. R. A Unique Geometry of the Active Site of Angiotensin-Converting Enzyme Consistent with Structure-Activity Studies. *J. Comput.-Aided Mol. Design* **1987**, *1*, 3–16.

(3) Crippen, G. M. Distance Geometry Approach to Rationalizing Binding Data. *J. Med. Chem.* **1979**, *22*, 988–997.

(4) Sheridan, R. P.; Nilakantan, R.; Dixon, J. S.; Venkataraghavan, R. The Ensemble Approach to Distance Geometry: Application to the Nicotinic Pharmacophore. *J. Med. Chem.* **1986**, *29*, 899–906.

(5) Lloyd, E. J.; Andrews, P. R. A Common Structural Mode for Central Nervous System Drugs and Their Receptors. *J. Med. Chem.* **1986**, *29*, 453–462.

(6) *Catalyst.* BioCAD Corporation: 1091 N. Shoreline Blvd., Mountain View, CA, 1991.

(7) *Apex-3D* Biosym Technologies, Inc.: 9685 Scranton Road, San Diego, CA, 1993.

(8) Goodford, P. J. A Computational Procedure for Determining Energetically Favourable Binding Sites on Biologically Important Macromolecules. *J. Med. Chem.* **1985**, *28*, 849–857.

(9) Lewis, R. A.; Roe, D. C.; Kuntz, I. D. Automated Site-Directed Drug Design Using Molecular Lattices. *J. Mol. Graphics* **1992**, *10*, 55–56.

(10) (a) Böhm, H. J. The Computer Program LUDI: A New Method for the de Novo Design of Enzyme Inhibitors. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 61–78. (b) Böhm, H. J. LUDI: Rule-based Automatic Design of New Substituents for Enzyme Inhibitor Leads. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 593–606.

(11) (a) Gund, P. Three-Dimensional Pharmacophore Pattern Searching. *Prog. Mol. Sub-Cell. Biol.* **1977**, *5*, 117–143. (b) Gund, P.; Wipke, W. T.; Langridge, R. Computer Searching of a Molecular Structure File for Pharmacophoric Patterns. In *Proceedings of the International Conference on Computers in Chemical Research and Education*; Elsevier: Amsterdam, 1974; Vol. 3, pp 33–38.

(12) (a) Allen, F. H.; Bellard, S.; Brice, M. D.; Cartwright, B. A.; Doubleday, A.; Higgs, H.; Hummelink, T.; Hummelink-Peters, B. G.; Kennard, O.; Motherwell, W. D. S.; Rogers, J. R.; Watson, D. G. The Cambridge Crystallographic Data Centre: Computer-Based Search, Retrieval, Analysis and Display of Information. *Acta Crystallogr.* **1979**, *B35*, 2331–2339. (b) Allen, F. H.; Kennard, O.; Taylor, R. Systematic Analysis of Structural Data as a Research Technique in Organic Chemistry. *Acc. Chem. Res.* **1983**, *16*, 146–161. (c) Allen, F. H.; Davies, J. E.; Galloy, J. J.; Johnson, O.; Kennard, O.; Macrae, C. F.; Mitchell, E. M.; Mitchell, G. F.; Smith, J. M.; Watson, D. G. The Development Of Versions 3 and 4 of the Cambridge Structural Database System. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 187–204.

(13) (a) Jakes, S. E.; Willet, P. Pharmacophoric Pattern Matching in Files of 3-D Chemical Structures: Selection of Interatomic Distance Screens. *J. Mol. Graphics* **1986**, *4*, 12–20. (b) Jakes, S. E.; Watts, N. J.; Willet, P.; Bawden, D.; Fisher, J. D. Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Evaluation of Search Performance. *J. Mol. Graphics* **1987**, *5*, 41–48. (c) Brint, A. T.; Willet, P. Pharmacophore Pattern Matching In Files of 3D Chemical Structures: Comparison of Geometric Searching Algorithms. *J. Mol. Graphics* **1987**, *5*, 49–56. (d) Brint, A. T.; Willet, P. Identifying 3D Maximal Common Substructures Using Transputer Networks. *J. Mol. Graphics* **1987**, *5*, 200–207. (e) Brint, A. T.; Willet, P. Algorithm for the Identification of Three-Dimensional Maximal Common Substructures. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 152–158. (f) Poirette, A. R.; Willet, P.; Allen, H. F. Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Characterization and Use of Generalized Valence Angle Screens. *J. Mol. Graphics* **1991**, *9*, 203–217.

(14) Martin, Y. C.; Danaher, E. B.; May, C. S.; Weininger, D. MENTHOR, A Database System for the Storage and Retrieval of Three-Dimensional Molecular Structures in Associated Data Searchable by Substructural, Biologic, Physical, or Geometric Properties. *J. Comput.-Aided Mol. Design* **1988**, *2*, 15–29.

(15) (a) Van Drie, J. H.; Weiniger, D.; Martin, Y. C. ALADDIN: An Integrated Tool for Computer-Assisted Molecular Design and Pharmacophore Recognition from Geometric, Steric, and Substructure Search of 3D Molecular Structures. *J. Comput.-Aided Mol. Des.* **1989**, *3*, 225–251. (b) Martin, Y. C. Computer Design of Potentially Bioactive Molecules by Geometric Searching with ALADDIN. *Tetrahedron Comput. Methods* **1990**, *3*, 15–25.

(16) (a) Sheridan, R. P.; Nilakantan, R.; Rusinko, A.; Bauman, N.; Haraki, K. S.; Venkataraghavan, R. 3DSEARCH: A System for Three-Dimensional Substructure Searching. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 255–260. (b) Sheridan, R. P.; Rusinko, A.; Nilakantan, R.; Venkataraghavan, R. Searching for Pharmacophores in Large Coordinate Data Bases and Its Use in Drug Design. *Proc. Natl. Acad. Sci. U.S.A.* **1989**, *86*, 8165–8169.

(17) Bartlett, P. A.; Shea, J. T.; Telfer, S. J.; Waterman, S. CAVEAT: A Program to Facilitate the Structure-Derived Design of Biologically Active Molecules. *Spec. Publ.—R. Soc. Chem.* **1989**, *78* (Molecular Recognition in Chemical and Biological Problems), 182–196.

(18) (a) Moock, T. E.; Christie, B.; Henry, D. MACCS3D: A New Database System for Three-Dimensional Molecular Models. In *Chemical Information Systems: Beyond the Structure Diagram*; Bawden, D.; Mitchell, E. M., Eds.; Ellis Horwood: Chichester, 1990; pp 42–49. (b) Christie, B. D.; Henry, D. R.; Guner, O. F.; Moock, T. E. MACCS-3D: A Tool for Three-Dimensional Drug Design. In *Proceedings of the 14th International On-Line Information Meeting*; Raitt, D., Ed.; Learned Information: London, 1990; pp 137–161.

(19) Murral, N. W.; Davies, E. K. Conformational Freedom in 3D Databases. 1. Techniques. *J. Chem. Inf. Comp. Sci.* **1990**, *30*, 312–316.

(20) (a) DesJarlais, R. L.; Sheridan, R. P.; Dixon, J. S.; Kuntz, I. D.; Venkataraghavan, R. Docking Flexible Ligands to Macromolecular Receptors by Molecular Shape. *J. Med. Chem.* **1986**, *29*, 2149–2153. (b) DesJarlais, R. L.; Sheridan, R. P.; Seibel, G. L.; Dixon, J. S.; Kuntz, I. D.; Venkataraghavan, R. Using Shape Complementarity as an Initial Screen in Designing Ligands for a Receptor Binding Site of Known Three-Dimensional Structure. *J. Med. Chem.* **1988**, *31*, 722–729. (c) DesJarlais, R. L.; Seibel, G. L.; Kuntz, I. D.; Furth, P. S.; Alvarez, J. C.; Ortiz de Montellano, P. R.; DeCamp, D. L.; Craik, C. S. Structure-Based Design of Non-Peptidic Inhibitors Specific for the Human Immunodeficiency Virus 1 Protease. *Proc. Natl. Acad. Sci. U.S.A.* **1990**, *87*, 6644–6648. (d) Shoichet, B. K.; Bodian, D. L.; Kuntz, I. D. Molecular Docking Using Shape Descriptors. *J. Comput. Chem.* **1992**, *13*, 380–397. (e) Meng, E. C.; Shoichet, B. K.; Kuntz, I. D. Automated Docking with Grid-Based Energy Evaluation. *J. Comput. Chem.* **1992**, *13*, 505–524.

(21) Lawrence, M. C.; Davis, P. C. CLIX: A Search Algorithm for Finding Novel Ligands Capable of Binding Proteins of Known Three-Dimensional Structure. *Proteins* **1992**, *12*, 31–41.

(22) (a) Martin, Y. C.; Bures, M. G.; Willet, P. Searching Databases of 3D Structures. In *Review in Computational Chemistry*; Lipkowitz, K. B., Boyd, D. B., Eds.; VCH: New York, 1990; pp 213–263. (b) Willet, P. *Three-Dimensional Chemical Structure Handling*; Research Studies Press: Taunton, 1991. (c) Martin, Y. C. 3D Database Searching in Drug Design. *J. Med. Chem.* **1992**, *35*, 2145–2154.

(23) (a) Rusinko, A.; Sheridan, R. P.; Nilakantan, R.; Haraki, K. S.; Bauman, N.; Venkataraghavan, R. Using CONCORD to Construct a Large Database of Three-Dimensional Coordinates from Connection Tables. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 251–255. (b) Leach, A. R.; Prout, K.; Dolata, D. P. The Application of Artificial Intelligence to the Conformational Analysis of Strained Molecules. *J. Comput. Chem.* **1990**, *11*, 680–693. (c) Bartlett, P. A., Lauri, G., Weiss, G. CAVEAT, ILIAD & TRIAD: 3D Searching for Molecule Design. 12th Annual Conference of the Molecular Graphics Society, Interlaken, Switzerland, June 7–11, 1993; Abstract O32.

(24) Nishibata, Y.; Itai, A. Automatic Creation of Drug Candidate Structures Based on Receptor Structure. Starting Point for Artificial Lead Generation. *Tetrahedron* **1991**, *43*, 8985–8990.

(25) Nishibata, Y.; Itai, A. A Starting Point for Rational Lead Generation: Automatic Creation of Drug Candidate Structures Based on 3D-Structure Of Receptor. XIIth International Symposium on Medicinal Chemistry, Basel, Switzerland, September 13–17, 1992; Abstract OC-08.5.

(26) Bohacek, R. S. GROW: A New Method for Generating Large Numbers of Novel Guest Molecules that are Sterically and Chemically Compatible to a Host Molecule Binding Site: Application To An Enzyme/Inhibitor System. XIIth International Symposium On Medicinal Chemistry, Basel, Switzerland, September 13–17, 1992; Abstract OC-08.4.

(27) Rotstein, S. H.; Murcko, M. A. GenStr 1.0: A Method for *De Novo* Drug Design. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 23–43.

(28) (a) Danziger, D. J.; Dean, P. M. Automated Site-Directed Drug Design: A General Algorithm for Knowledge Acquisition about Hydrogen-Bonding Regions at Protein Surfaces. *Proc. R. Soc. London* **1989**, *B 236*, 101–113. (b) Danziger, D. J.; Dean, P. M. Automated Site-Directed Drug Design: The Prediction and Observation of Ligand Point Positions at Hydrogen Bonding Regions on Protein Surfaces. *Proc. R. Soc. London* **1989**, *B 236*, 115–124. (c) Lewis, R. A.; Dean, P. M. Automated Site-directed Drug Design: The Concept of Spacer Skeletons for Primary Structure Generation. *Proc. R. Soc. London* **1989**, *B 236*, 125–140. (d) Lewis, R. A.; Dean, P. M. Automated Site-directed Drug Design: The Formation of Molecular Templates in Primary Structure Generation. *Proc. R. Soc. London* **1989**, *B 236*, 141–162.

(29) Lewis, R. A. Automated Site-Directed Drug Design: Approaches to the Formation of Three-Dimensional Molecular Graphs. *J. Comput.-Aided Mol. Des.* **1990**, *4*, 205–210.

(30) Moon, J. B.; Howe, W. J. Computer Design of Bioactive Molecules: A Method for Receptor-Based de Novo Ligand Design. *Proteins* **1991**, *11*, 314–328.

(31) Gillet, V.; Johnson, A. P.; Mata, P.; Sike, S.; William, P. SPROUT: A Program for Structure Generation. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 127–153.

(32) Rotstein, S. H.; Murcko, M. A. GroupBuild: A Fragment-Based Method for *De Novo* Drug Design. *J. Med. Chem.* **1993**, *36*, 1700–1710.

(33) Structure of methotrexate complexed to the enzyme, Protein Data Bank, entry code 4DFR. For a description of the Protein Data Bank, see: Bernstein, F. C.; Koetzle, T. F.; Williams, G. J. B.; Meyer, E. F.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures. *J. Mol. Biol.* **1977**, *112*, 535–542.

(34) X-ray structure, Cambridge Structural Database (see ref 12), entry code INDMET.

(35) X-ray structure of the inhibitor complexed to the enzyme, Protein Data Bank, entry code 9HVP.

(36) Mohamadi, F.; Richards, N. G. J.; Guida, W. C.; Liskamp, R.; Lipton, M.; Caufield, C.; Chang, G.; Hendrickson, T.; Still, W. C. MacroModel—An Integrated Software System for Modeling Organic and Bioorganic Molecules using Molecular Mechanics. *J. Comput. Chem.* **1990**, *11*, 440–467.

(37) For the NCC force field, see: Cohen, N. C. In *Advances in Drug Research*; Testa, B., Ed.; Academic Press: New York, 1985; Vol. 14, p 50 and references therein.