Journal of Computer-Aided Molecular Design, 11 (1997) 175–192
ESCOM

J-CAMD 385

Evaluation of a method for controlling molecular scaffold diversity in de novo ligand design

N.P. Todorov* and P.M. Dean

Department of Pharmacology, University of Cambridge, Tennis Court Road, Cambridge CB2 1QJ, U.K.

Received 6 September 1996
Accepted 8 November 1996

Keywords: Simulated annealing; Drug design; Penalty function optimization; Molecular diversity

Summary

We describe an algorithm for the automated generation of molecular structures subject to geometric and connectivity constraints. The method relies on simulated annealing and simplex optimization of a penalty function that contains a variety of conditions and can be useful in structure-based drug design projects. The procedure controls the diversity and complexity of the generated molecules. Structure selection filters are an integral part and drive the algorithm. Several procedures have been developed to achieve reliable control. A number of template sets can be defined and combined to control the range of molecules which are searched. Ring systems are predefined. Normally, the ring-system complexity is one of the most elusive and difficult factors to control when fusion-, bridge- and spiro-structures are built by joining templates. Here this is not an issue; the decision about which systems are acceptable, and which are not, is made before the run is initiated. Queries for inclusion and exclusion spheres are incorporated into the objective function, and, by using a flexible notation, the structure generation can be directed and more focused. Simulated annealing is a reliable optimizer and converges asymptotically to the global minimum. The objective functions used here are degenerate, so it is likely that each run will produce a different set of good solutions.

Introduction

Computational methods for de novo ligand design are becoming increasingly important in the process of suggesting molecular structures for target therapeutic sites. The key problems in automated de novo structure generation are validation of the algorithm by considering reproducibility, enrichment of the algorithm for creating molecular diversity and, lastly, constraining the algorithm for reducing the complexity for future chemical synthesis. The origin of these problems lies in the combinatoric nature of structural assembly from a database of component molecular fragments. Scant attention has been paid to the combinatorial number of steps involved in structure assembly [1]. Even on a basic geometric skeleton, the number of possible atom assignments can be very large [2]. In practice, this combinatorial process is magnified by the amount of tolerance allowed in fitting the generated trial structures into the defined site. In this paper, we describe an algorithm for the automated generation of molecular structures satisfying both geometric and connectivity constraints. Extensive tests are performed for the validation of generated structures with a known ligand/site co-crystal complex. The classification of generated structures enables the diversity of possible solutions to be compared for different constraints imposed on the algorithm.

Two concepts for the design of ligand structures have been widely used: docking of molecules from a database and de novo design. Reviews on molecular docking have been published by Martin [3] and Blaney and Dixon [4]. A major advantage of docking algorithms is that the structures are guaranteed to be synthetically accessible. De novo design algorithms could produce completely novel molecules which have not been studied before; searches are not restricted to a limited set of molecules. Recent reviews by Slater and Timms [5], Verlinde and Hol [6] and Lewis and Leach [7] highlight the major developments in this area. Two approaches to the de

*To whom correspondence should be addressed.

novo design of molecules have been attempted. Firstly, chemical fragments are initially placed in the site to try to ensure favourable interaction and later some of them are connected into a single molecule either directly or through other bridge fragments [8–16]. Secondly, the structure is kept connected from the beginning to the end of the construction [17–24].

The structures of many proteins are solved by X-ray crystallography, NMR spectroscopy, or approximated by molecular modelling. Methods are available for the automatic determination of binding sites on the protein surface [25–27]. Receptors are usually viewed as rigid bodies and design procedures focus on the places around the receptor where a ligand is likely to bind. Binding sites could also be inferred indirectly, using molecular similarity methods to superpose active compounds optimally [28]. An envelope defined by the superposed structures restricts the space that new molecules could occupy. Pharmacophore points within the envelope determine the local properties needed. After the envelope and pharmacophore groups are derived from the molecular matching analysis, the problems of molecular design with known and unknown 3D receptor structure are closely related and can be treated similarly.

As intimated earlier, de novo ligand design is a combinatorial process related directly to the product of the number of molecular fragments that can be combined within various constraints. A variety of strategies have been developed to tackle general optimization problems. For example, branch-and-bound and A* algorithms [29] have been successful for conformational optimization, genetic algorithms [30] have been tailored to a number of drug design problems, and simulated annealing [31] has been used for molecular similarity and atom assignment problems. Our philosophy in de novo ligand design has been to divide the combinatorial problems into two parts. Firstly, our aim is to generate 3D molecular graphs to fill defined surfaces such as that of a binding site, or a supersurface obtained from molecular similarity studies. Secondly, atom assignments can then be made onto the graphs according to the dictates of molecular fields found in the site or on the defining supersurface. The latter problem has been commented on extensively [2] and we do not wish to add to it here. There are two main reasons for this division: (i) a long-term goal of our work is to analyse the production of molecular scaffolds for any constraining site or surface; and (ii) in tandem we wish to study the molecular diversity associated with atom assignment to the scaffolds. In the present paper, we use simulated annealing to study geometric problems of structure generation within a set of constraints applied to the objective function. By a careful manipulation of these constraints, it is possible to evaluate their effect on the reproducibility of structure generation and on the creation of molecular scaffold diversity.

Methods

This paper describes a structure generation algorithm that uses an optimization method (simulated annealing) to minimize a penalty function reflecting the suitability of a chemical structure as a solution to a particular problem. Structures are built by joining together predefined 3D molecular templates. The algorithm navigates through the space of connectivity classes, conformations and rigid-body transformations of various structures, guided by the function, until a skeleton is encountered that satisfies all set criteria.

Simulated annealing

The method of simulated annealing [31] is often used in practical problems for the global optimization of functions with multiple minima. An objective function, E, is defined over some configurational space. Transitions from one state s_1 to another state s_2 are generated randomly. The respective function values E_1 and E_2 are calculated and compared. If E_2 is less than E_1, the transition is accepted. Otherwise s_2 can become a successor of s_1 with a certain probability, p, that depends on a control parameter, T, analogous to temperature. The form of p is known as the Metropolis condition:

$$p(s_2|s_1) = \exp((E_1 - E_2)/T) \quad (1)$$

Initially, T is set high and many configurations for which the functional value deteriorates are accepted. The algorithm is able to escape from local minima and explore global features of the landscape. As T decreases, fewer unfavourable transitions are accepted; the algorithm gradually switches from a more random search, covering large regions of the configurational landscape, to a more local one, until eventually the global minimum state is found.

Templates

Structures are built from small 3D molecular templates. Template libraries can be easily configured by simply listing the names of the template files. In addition, one can specify the frequency of template use, the minimum and maximum number of copies required in the structures and whether the template is rigid or flexible. By default, all templates are used with the same frequency, with no restrictions on the number of copies; rigid conformations are assumed. Usually, we use a set of hydrocarbon generic templates which have been regularized by molecular mechanics minimization with the COSMIC force field [32]. The terminal atoms in each template are used to define seed positions where other fragments can be joined. By default all such atoms are used, but optionally some can be omitted.

Bond orders can also be included in the template description (otherwise all bonds are assumed single) and it

can be specified that the orders of the bonds connecting different fragments in the structures should match. Usually, we do not elaborate on bond orders and find it more efficient to divide the process into primary and secondary structure generation [8,9], although the algorithm allows the use of a single-stage procedure as well.

Configurations and initial state

An initial structure is built by linking together several fragments in a stepwise manner. For this purpose, the algorithm uses a fragment addition procedure which is described later. Essentially, an arbitrary initial structure is generated at the end of this stage, with the number of incorporated fragments restricted between specified minimum and maximum values. Some of the atoms in that structure may be positioned in forbidden regions of the receptor site; close intramolecular contacts may also be present. These and other possible violations are considered in the objective function and later resolved.

Transitions

Several modifications can be applied to generate transitions from one structure to another: rigid-body displacements, bond rotations, fragment additions, fragment removals and fragment replacements. Unless explicitly stated otherwise, random numbers are used to make various selections described in the text.

Fragment additions

A bond in the structure is selected together with a new fragment from the library. If the bond is terminal, the fragment is added to the structure using one of its seed atoms. If the bond is not terminal, the new fragment is inserted between the fragments linked by the selected bond using two of its seed atoms. The new structure is then refitted onto the old one using the correspondences between the atoms common to both structures. By default, only bonds with both constituent atoms lying inside the accessible volume are used in the procedure. If no such bonds exist, then a random bond is selected.

The new fragment should contain a minimum number of seed atoms (either one or two) to keep the new structure fully connected. For this purpose, appropriate lists of fragments, containing a specified minimum number of seed atoms, are compiled when the program is initialized and used for this selection. The frequency of use of each fragment (specified in the template library description) is also included in the list and taken into account.

Fragment removals

A fragment from the structure with one or two connections to other fragments is selected and removed. When two disconnected substructures are produced (in the case of two connections), the substructures are joined using the bonds previously linking them to the deleted fragment. The new structure is refitted onto the old one using the correspondences between the atoms common to both structures.

Fragment replacements

A fragment in the structure is replaced by a fragment selected from the library which contains a sufficient number of seed atoms to keep the structure fully connected. The seed-atom lists described above are used to select an appropriate fragment. Substituents of the old fragment are reconnected at seed-atom positions around the new fragment. The new structure is refitted onto the old one using the correspondences between the atoms common to both structures.

Rigid-body translations and rotations

The structure is translated by a random amount within a specified range. It is also rotated around a random axis passing through one of its atoms. One translation and one rotation are used in a single move. By default, the maximum translation distance is 0.2 Å and the maximum rotation angle is 5°.

Bond rotations

An acyclic bond is selected and the structure is rotated around it. Both parts of the structure that are separated by the bond can be selected for the rotation. Bonds which belong to templates specified as rigid are excluded. By default, the maximum rotation angle is 10°.

Frequency of different transition types

The frequency of generation of the moves is subject to adjustment. One peculiarity is that fragment addition and fragment removal type transitions are combined into a single transition type. When this new transition type is selected, a further decision is made about whether to add, remove or keep the same number of fragments in the structure. Equal probabilities are used to select among these three alternatives, with the restriction of keeping the number of fragments within the defined minimum and maximum bounds throughout the run. By default, equal selection frequencies are used for all four types of transition: rigid-body translation/rotation, bond rotation, fragment addition/removal and fragment replacement.

Ring closures

Before the score is calculated, an optional check is performed to determine whether additional rings can be formed. Pairs of terminal and non-terminal atoms closer than a certain distance are collected into a list. By default, no ring-closure checks are performed; this is specified by setting this distance threshold value to –1.0 Å. The entries in the list are matched pairwise against each other. If the terminal atoms from each entry are connected to the non-terminal atoms of the opposite entry,

the terminal atoms are removed and a new (ring-closure) bond is formed between the non-terminal atoms.

Bond orders

Bond order information can also be taken into account by two mechanisms: either as a penalty term in the objective function or while the structure is built and modified. In the second procedure the transitions are carefully checked to ensure bond order consistency. When a fragment is added or replaced, not only fragments with a minimal number of seed atoms are used to keep the structure connected, but also their bond orders should match. When a fragment with two neighbours is deleted, the bonds by which the two resulting substructures are joined should be of the same order. Bond orders are also checked when additional rings are formed.

Definition of several template sets and joining rules

It is useful to have a way to specify that certain fragments should be present in all solutions. Two mechanisms have been developed to achieve this. The first one uses a penalty term in the objective function which counts how many copies of each template are incorporated into the structure. It works well in many cases, but may not be very reliable if there are a large number of templates in the library and strict constraints. In the second strategy, the required templates are selected in the beginning and are incorporated into the ligand all the way through. It is also possible to replace one of these templates by another one from a predefined set. Thus, several template sets are defined and each template from one set can replace any other template from the same set. One can think of the sets as labelled by integer numbers 1,2,... and that templates with the same label are interchangeable.

There is also a default (0th) set which is used to incorporate specified templates into the structures in addition to the templates from the other sets. Templates from the default set are used when the number of fragments in the structure is changed; one of them is selected when a fragment is added to, or deleted from, the evolving ligand. This allows one to change the number of fragments in the structure and then sample structures with all possible values between the specified minimum and maximum number of fragments. Fragments from the other sets are always present, but can be mutated to other templates from the same set, or their connections to fragments from the other sets may be changed.

For finer control, one can specify that some of the atoms in the templates should not be used as seeds where other templates can be joined. This option will limit the search over a more restricted structural space. Bond-order labels can also be used effectively to restrict molecular diversity. For example, it is possible to define several different labels for single bonds, say 1, 1', 1", etc., in the template files and require consistent bond-order labelling.

These mechanisms allow one to define the range of interesting molecules as wide, or as narrow, as desired by varying the number and the content of the template sets. There is a product relationship of the number of templates in each library and the number of molecules which will be sampled. In addition, different seed atoms and bond-order labels can be used to interconnect the templates and further increase or reduce structural diversity.

Next, we shall summarize how this mechanism operates in practice. When the initial structure is built, one template is selected (randomly) from each of the non-default sets and incorporated into the structure. After that, several templates are added from the default set. Of course, the number of defined template sets should be consistent with the specified minimum and maximum limit for the number of fragments.

When transitions are generated, templates are replaced only by templates from the set to which they belong. For example, for the fragment addition/removal transition this works as follows. When the number of fragments does not change, a fragment from the structure is selected together with a new fragment from the same set; the old fragment is removed and the new one is added to the structure. When the number of fragments in the structure changes, only fragments from the default (0th) set are used.

A complication could arise during a fragment removal operation if no suitable fragment is available for deletion. This can happen if more than one template set is defined and all templates from the default set are 'buried' among three or more non-default templates connected to them (or two templates, but with connection bonds of different order). In such cases (usually very rare) a warning is issued and another type of transition is selected.

Each of the incorporated templates can also be brought near a specified position in the site by including a special term in the objective function (Appendix, Note 1). Also, an option is provided to place and keep templates from one of the sets at their specified positions.

In the simplest case, only one set of templates (the default) is defined. Additional sets are defined to incorporate desirable templates into all generated structures. This is not a complicated procedure; simply a set label is appended after the name of each template file in the library. If the label is skipped, the default set is assumed.

Scoring

The form of the score function, F, is selected to become zero when all the constraints are satisfied and to be positive otherwise. Thus, miscellaneous constraints could be set and it would be clear when all of them are satisfied. Various constraints are encoded into a penalty function, F, composed of 11 terms:

$$F = \sum_{i=1}^{11} w_i F_i \qquad (2)$$

where F_i are terms that correspond to different constraint types and w_i are weighting factors. F_i are further expressed through a function, f, defined as

$$f(x,x',x'') = \begin{cases} x - x' & \text{if } x < x' \\ 0 & \text{if } x' \le x \le x'' \\ x'' - x & \text{if } x > x'' \end{cases} \quad (3)$$

f is zero if the variable x is within a specified interval and linearly positive otherwise. F_1 is a ligand-point term, F_2 is a bond-path term, F_3 is a repulsive intermolecular steric term, F_4 is an attractive intermolecular steric term, F_5 is an intramolecular steric term, F_6 is a torsion energy term, F_7 is a number-of-atoms constraint term, F_8 is a number-of-rings constraint term, F_9 is a number-of-templates constraint term, F_{10} is a template-deviation-from-original-position term and F_{11} is a bond-order-mismatch term. See the Appendix, Note 1, for details.

Simulated annealing implementation

In this work, we use a heterogeneous annealing algorithm [33] with an exponential cooling schedule. The default initial temperature is set to 1.0, the length of the Markov chain is 500, the temperature decrement factor between successive Markov chains is 0.9 and the number of Markov chains is 20. If a structure with a sufficiently low score is encountered, the annealing procedure is terminated. Otherwise, it continues until the specified number of Markov chains is completed. In both cases, the score of the best structure found in the annealing is checked against another threshold and, if lower, the structure is written to disc. The default values for the score thresholds to terminate the annealing, and to write the structures out, were both set to 0.5. The annealing procedure is repeated either a specified number of times or until a specified number of solutions is generated.

Structure clean-up

Each solution structure is cleaned up before it is written out. There are three options here. First, the structures are written as they come out of the annealing. Second, terminal acyclic atoms which are not essential are trimmed while keeping the structure fully connected. An atom is considered essential if it matches a ligand point or (optionally) is involved in a steric interaction. Third, acyclic and ring atoms are removed if they are not essential. Rings are kept intact if they contain at least one atom which makes a contribution. The third option is the default.

Families of structures

The structure generation procedure is not systematic and when a large set of structures is generated, identical copies of the same molecule will appear. The structures are classified on the basis of structure connectivity and atom positions.

A direct connectivity comparison between two structures is not possible because the atom numbering is not necessarily the same. The correspondence between the atoms of the two molecules is established by canonicalization of the connectivity table of each molecule. The atoms are renumbered in a unique way independent of their initial order. See the Appendix, Note 2, for details.

Structure comparison

After the structures are renumbered canonically, each of them is pairwise compared with those already processed and lists are compiled of structures with the same connectivity. Optionally, for each atom in one of the structures being compared, the squared distance is found to the nearest atom from the same symmetrically equivalent class in the second structure. The sum of these distances is accumulated over the atoms of both molecules and the equivalent of an rms value is calculated. If this value is less than a specified threshold, the structures are considered equivalent. This option allows one to distinguish between different binding modes of structures with the same connectivity.

Implementation

The structure generation algorithm has been implemented in a set of computer programs SKELDIV, written in FORTRAN 77 and C. The following routines from Numerical Recipes [34] have been used: RAN1 and RAN3 for random number generation and AMOEBA and AMOTRY for the simplex optimization. The programs were compiled on Dec Alpha 600 model 5/266 at optimization level 4. Further details on parameters and usage are provided in the Appendix, Note 3.

Results

Protein test site

The methotrexate (MTX) binding site of *Lactobacillus casei* dihydrofolate reductase is used to illustrate the method. This enzyme catalyses the conversion of dihydrofolate to tetrahydrofolate, which is an important reaction in the biosynthetic pathway of nucleic acids. Dihydrofolate reductase from *L. casei* co-crystallized with NADPH and the competitive inhibitor methotrexate is entry 3DFR [35] in the Brookhaven Protein Databank. For this experiment we removed NADPH and all water molecules from the site.

Lattice maps

The lattice map of the site was constructed inside a cuboid with edges parallel to the coordinate axes and 4.0 Å away from any atom of MTX. A grid step value of 0.3 Å was used. The volume accessible for structure generation was defined using the default parameters. There were 213 864 lattice points, of which 38 819 were accessible.

Contact scores

The number of protein non-hydrogen atoms within 4.5 Å was found for each lattice point inside the accessible volume and later used to calculate the attractive steric score of the ligand. The calculation of lattice maps and contact scores takes about 3 s. The minimum required number of contacts between the protein and the ligand was set to 200.

Ligand points

A query was used composed of six spheres, with radii of 1.0 Å, with centres at the positions of the N3, N6, O27, O28, O32 and O33 atoms of MTX. The data were extracted for the 3DFR entry file and converted into a format which can be read by the program. All spheres were to be matched by terminal ligand atoms and the atoms matching spheres O27 and O28 and spheres O32 and O33 to be separated by two bonds. This query models the connectivity pattern of the MTX molecule. In some of the tests, another sphere centred at N3 with a radius of 3.0, 4.0 or 5.0 Å was added in order to be matched by a ring centre. This constraint was used to enhance the placement of rings in the pteridine pocket.

Templates

Generic hydrocarbon templates were used with all non-terminal and all terminal atoms specified respectively as C and H. The geometries of the templates were minimized by molecular mechanics using the COSMIC force field [32]. Templates are specified precisely for each run, but for a general overview see Figs. 3–5.

Test runs

Structure generation can be regarded as a process of continuous refinement which starts from some simple requirements; constraints are gradually introduced or relaxed until the most satisfactory performance is achieved. After this methodological probing a long production run is initiated for detailed analysis and, possibly, reiteration. One hundred solutions were generated in each run, except in run 7', in which the number of solutions was 10 000. The results of the tests described below are summarized in Table 1. The total number of annealing trials to generate the required number of solutions can be estimated from the sol (%) column in the table. For example, a 50% success rate means that 200 trials are necessary to generate 100 solutions. Figure 1 depicts the first 10 solutions produced in each run. This allows for a visual assessment of the results.

Test run 1

Three sets of templates were defined. The default (0th) set contained methane and formaldehyde-like templates, set number one contained naphthalene and set number two contained benzene. Thus, each solution structure would contain one naphthalene, one benzene and a number of methane and formaldehyde templates, and in this respect they would resemble the MTX ligand. Structures were requested with 5–15 fragments, 15–50 non-hydrogen atoms. At the start of the annealing from a random arrangement the score was about 100 and then gradually decreased.

TABLE 1
RESULTS FROM STRUCTURE GENERATION IN THE MTX BINDING SITE OF 3DFR

Test run	$\overline{f}$	σ	sol (%)	cpu_{trial} (s)	cpu_{sol} (s)
1	0.51	0.95	77.52	9.22	11.89
2	0.37	0.37	84.75	8.71	10.27
3	0.41	0.36	78.74	9.33	11.84
4	1.43	1.58	39.37	11.00	27.95
5	0.57	1.76	75.19	8.23	10.95
6	0.65	1.66	78.13	8.18	10.47
7	1.21	2.07	59.17	8.92	15.08
7'	1.23	2.65	61.13	8.64	14.14
8	0.64	1.81	76.34	7.91	10.36
9	0.83	1.08	61.35	9.81	16.00
10	0.32	1.65	91.74	8.57	9.34
11	0.97	1.12	53.76	10.77	20.03
12	1.26	1.14	32.37	11.87	36.55
13	1.09	1.69	46.73	12.60	26.96
14	1.05	1.72	44.44	13.80	31.05
15	0.60	0.70	69.44	9.99	14.38
16	0.98	0.81	38.46	11.87	30.87
17	0.66	0.64	60.61	12.14	20.02
18	0.71	0.75	58.48	13.19	22.56
19	0.56	0.74	74.07	9.92	13.39
20	0.85	1.04	44.05	11.16	25.33
21	0.72	0.71	57.14	12.16	21.28
22	0.73	0.73	56.50	13.03	23.07

One hundred solutions were generated in each run, except in run 7' in which the number of solutions was 10 000. $\overline{f}$ is the average score over all trials, σ is the score standard deviation, sol is the percentage of solutions, cpu_{trial} is the CPU time per trial and cpu_{sol} is the CPU time per solution.

Typically, the solution structures had the ring systems placed outside the pteridine cleft, although an occasional structure with a ring positioned in the pocket was produced (Fig. 1(1)). This suggests that there are a number of solutions satisfying the imposed constraints and that those with the pteridine ring placed outside the pocket are more numerous or more easily accessible.

Test runs 2–4

We shall illustrate how the F_{10} term of the objective function can be used constructively to specify that the naphthalene template is required in the pocket. The naphthalene template was superposed onto the pteridine ring of MTX by ring atom correspondences. During the run, the corresponding template in the structure was forced to occupy a similar position. A 2.0 Å constraint was used for the average deviation between non-hydrogen atoms. Thus, we have guaranteed that a ring system similar to

the pteridine one will be placed in the pocket. The program in most cases suggests alternative placements for the pteridine ring (Fig. 1(2)).

Another method for constraining the naphthalene template position is to alter the way in which the average deviation of that template from its desired position is calculated; this can be achieved by limiting the correspondences to those from the same symmetrically equivalent class. Generally, the matches between the template and the pteridine system in MTX are closer (Fig. 1(3)), but the difference is not great.

It is also possible to specify exactly the correspondences of the atoms between which the average deviation is measured to restrict even more the class of solutions produced (Fig. 1(4)). This gives a smaller percentage of solutions and a consequent larger time per solution.

There are further ways to constrain the diversity of the solutions, although such trials will not be examined here. For example, one can disable some of the seed atoms in the naphthalene template; or apply a smaller ring-centre radius; or restrict the number of bonds between the N3 and N6 ligand points; or restrict the bounds for the number of fragments in the structures; or specify fewer of the atoms in the naphthalene template as seed atoms where other templates can be joined.

Test runs 5–7

In this set of runs, the position of the benzene ring from the second template set was also constrained in a similar manner. Tolerances of 2.0 Å (run 5), 1.5 Å (run 6) and 1.0 Å (run 7) were tried for the average standard deviations of the naphthalene and the benzene templates. The deviation was calculated between all non-hydrogen atoms (as in run 2). The influence of a tighter constraint on the generated solutions is noticeable when Figs. 1(5), 1(6) and 1(7) are compared.

In this set of runs, we also tried to determine how many solutions have to be generated to get some structures with the same connectivity repeated as well as produce structures with the connectivity of MTX.

The 100 solutions for each run were classified in about 1.3 s. There were 99 structures with unique connectivity generated in run 5, 100 in run 6 and 98 in run 7.

Next, we included the MTX structure into the list of solutions and classified them again. No MTX replicas were present in all three runs. The program cleaned up the structures at the end of each annealing run by removing some of the atoms to extract the minimal essential set of atoms. C15 and N23 were deleted from the MTX structure, which would happen if a structure with the connectivity of MTX is produced in the annealing. One structure with such connectivity was produced in run 5.

Run 7 was repeated extensively to generate 10 000 solutions, and the results are included as entry 7' in Table 1. A total of 16 359 trials were performed. The 10 000 solutions were classified in 215 s; 3437 structures had unique connectivities. Different solutions were generated with different probabilities. The most frequently generated structure was produced 124 times. There were also 26 structures with the connectivity of MTX (with C15 and N23 deleted), which can be seen in Fig. 2.

The first solution with that connectivity was solution number 1146. On average, there was one such structure with that connectivity in every 378 solutions (with a standard deviation of 419). However, this was the 41st most frequently generated structure.

Test runs 8–10

In a separate run (results not included in Table 1), a purine-like template, with fused six- and five-membered rings, was docked into the pteridine pocket using only a 5 Å N3 constraint and the site steric constraints with no restriction on the number of contacts with the protein.

In run 8, the naphthalene template in the first library set was replaced with one of the docked templates and the same parameters were used as in run 5. The result is that a different ring system is placed in the pteridine pocket (Fig. 1(8)).

In run 9, both the naphthalene and the purine-like templates were included in the first set. Now either a naphthalene or a purine-like template was placed in the pocket (Fig. 1(9)).

In run 10, a number of other templates (14) were introduced in the second set (with no constraints on the average deviations). The templates were a collection of single and double (fused) ring systems (Fig. 3). Their coordinates were extracted from various molecular files, their atom types were corrected, hydrogen atoms were added to fill the spare valences and then minimized by molecular mechanics using the COSMIC force field. Thus, greater diversity is introduced compared with the original MTX molecule (Fig. 1(10)).

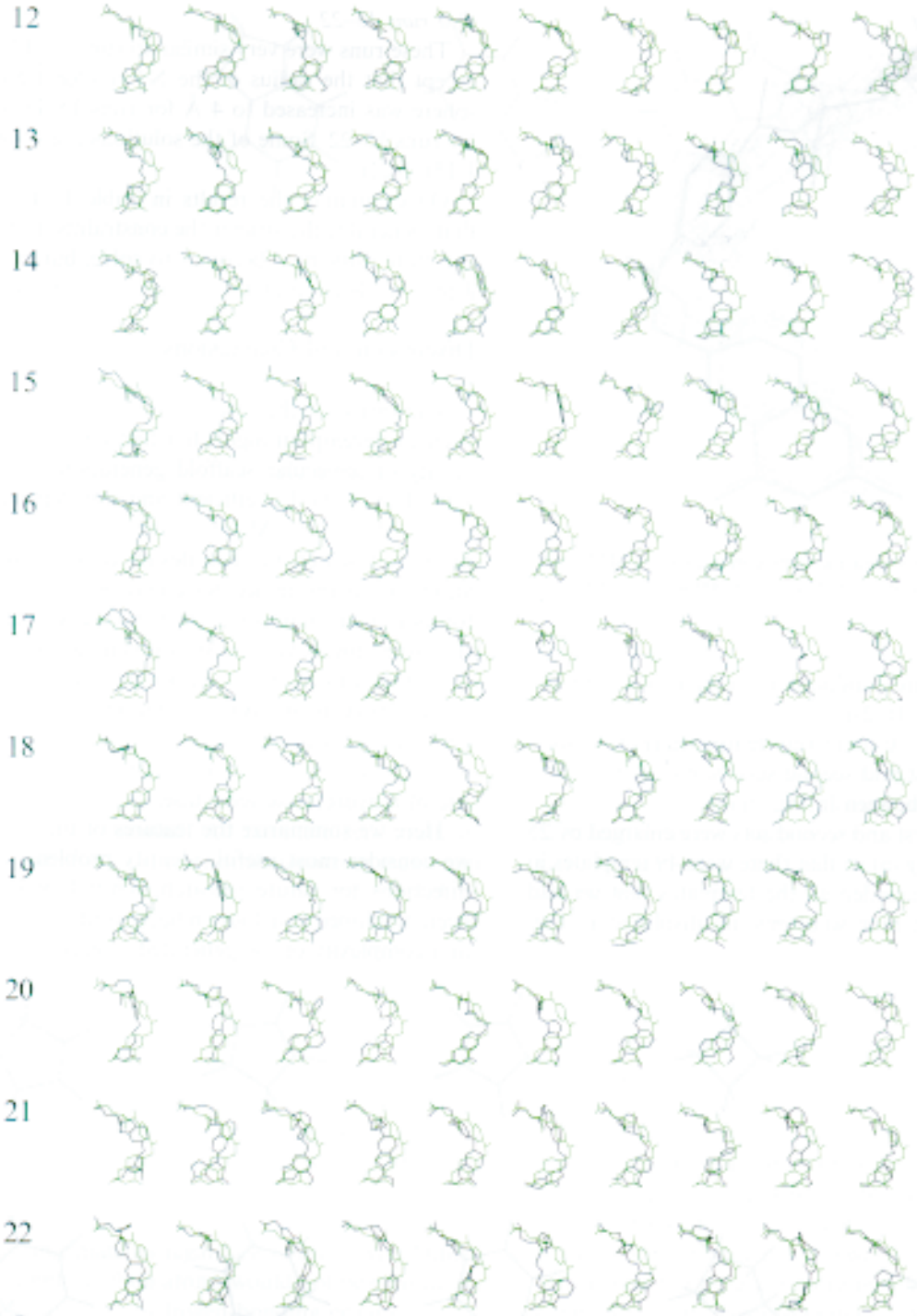
Test runs 11–14

In these runs, an additional sphere radius of 3 Å was defined inside the pocket which should be matched by a ring centre. This is an alternative way to enhance the positioning of rings in the pteridine pocket. The sphere was located at N3, but the exact position is not important.

In run 11, naphthalene was included in the first set and the 14 ring templates were used in the second template set. The 3.0 Å N3 sphere succeeds in filling the pteridine pocket with a ring template (Fig. 1(11)).

In the second run in the series, 14 ring templates (Fig. 3) were included in both sets one and two. The N3 3 Å constraint was used to force ring placement in the pteridine pocket; this is simpler than docking the templates individually and then using them as in run 10. More

Fig. 1. The first 10 solutions generated in each of runs 1–22 (black) and MTX as a reference molecule (green). (1) The default template set contained methane and formaldehyde-like templates, set number one contained naphthalene and set number two contained benzene. (2) The naphthalene template is required in the pocket; a 2.0 Å constraint was used for the average deviation between the non-hydrogen atoms of naphthalene and the pteridine ring in MTX. (3) The average deviation of the naphthalene template from its desired position is calculated limiting the correspondences to those from the same symmetrically equivalent class. (4) The correspondences of the atoms of naphthalene and pteridine between which the average deviation is measured are specified exactly. Tolerances of 2.0 Å (5), 1.5 Å (6) and 1.0 Å (7) were used for the average standard deviations of the naphthalene and the benzene templates to the corresponding moieties in MTX; the deviation was calculated between all non-hydrogen atoms. (8) The naphthalene template in the first library set is replaced by a purine-like template which matches the pteridine with an average deviation of 2 Å using all non-hydrogen atoms; a similar constraint applies to the benzene template as well. (9) Both the naphthalene and the purine-like templates are included in the first set; 2.0 Å is used to constrain their average deviation to non-hydrogen atoms of pteridine; a similar constraint applies to the benzene template as well. (10) The naphthalene and the purine-like templates are included in the first set; 2.0 Å is used to constrain their average deviation to non-hydrogen atoms of pteridine; 14 templates (Fig. 3) are used in the second template set with no constraints on the average deviations. (11) Naphthalene is included in the first set; the 3.0 Å N3 ring-centre sphere is used to constrain it in



the pteridine pocket; 14 templates (Fig. 3) are used in the second template set. (12) Fourteen templates (Fig. 3) are included in both sets one and two; the N3 3.0 Å constraint is used to force ring placement in the pteridine pocket. (13) Twenty-four templates (Figs. 3 and 4) are included in both sets one and two; the N3 3.0 Å constraint is used to force ring placement in the pteridine pocket. (14) Forty-nine templates (Figs. 3–5) are included in both sets one and two; the N3 3.0 Å constraint is used to force ring placement in the pteridine pocket. (15) Naphthalene is included in the first set; the 4.0 Å N3 ring-centre sphere is used to constrain it in the pteridine pocket; 14 templates (Fig. 3) are used in the second template set. (16) Fourteen templates (Fig. 3) are included in both sets one and two; the N3 4.0 Å constraint is used to force ring placement in the pteridine pocket. (17) Twenty-four templates (Figs. 3 and 4) are included in both sets one and two; the N3 4.0 Å constraint is used to force ring placement in the pteridine pocket. (18) Forty-nine templates (Figs. 3–5) are included in both sets one and two; the N3 4.0 Å constraint is used to force ring placement in the pteridine pocket. (19) Naphthalene is included in the first set; the 5.0 Å N3 ring-centre sphere is used to constrain it in the pteridine pocket; 14 templates (Fig. 3) are used in the second template set. (20) Fourteen templates (Fig. 3) are included in both sets one and two; the N3 5.0 Å constraint is used to force ring placement in the pteridine pocket. (21) Twenty-four templates (Figs. 3 and 4) are included in both sets one and two; the N3 5.0 Å constraint is used to force ring placement in the pteridine pocket. (22) Forty-nine templates (Figs. 3–5) are included in both sets one and two; the N3 5.0 Å constraint is used to force ring placement in the pteridine pocket.

Fig. 2. The 26 structures with the same connectivity as MTX (with atoms C15 and N23 deleted) found among 10 000 solutions generated in run 7'.

diverse solutions are produced in comparison with the previous tests (Fig. 1(12)).

In the next run, 10 more ring templates (Fig. 4) were included in the first and second sets. Some of the resulting structures can be seen in Fig. 1(13).

In run 14, the first and second sets were enlarged by 25 more templates (Fig. 5) so that there were 49 templates in each. In fact, we included all the templates that we had prepared. The first nine solutions are displayed in Fig. 1(14).

Test runs 15–22

These runs were very similar to runs 11, 12, 13 and 14 except that the radius of the N3 ring-centre constraint-sphere was increased to 4 Å for runs 15–18 and to 5 Å for runs 15–22. Some of the solutions are shown in Figs. 1(15)–1(22).

On examining the results in Table 1, it can be seen that, generally, the stricter the constraints in the problem, the more difficult it becomes to solve, but in all cases at a reasonable computational cost.

Discussion and Conclusions

The thrust of the research outlined in this paper has been to develop an algorithm that could control the diversity of molecular scaffold generation in de novo design. The method is efficient and not dependent on the nature of the site. Much of the impetus for molecular diversity research in drug design came from the realization that many molecules could possess similar values for molecular similarity at their molecular surfaces but, at the same time, have ostensibly dissimilar molecular structures. Our strategy in this paper has been to examine how this diversity in structure generation can be controlled by performing a detailed analysis for scaffold generation in a well-documented binding site, namely the methotrexate site of dihydrofolate reductase.

Here we summarize the features of the algorithm that we consider most useful, identify problems and outline directions for future research. Several procedures have been developed to achieve reliable control on the diversity and complexity of the generated molecules.

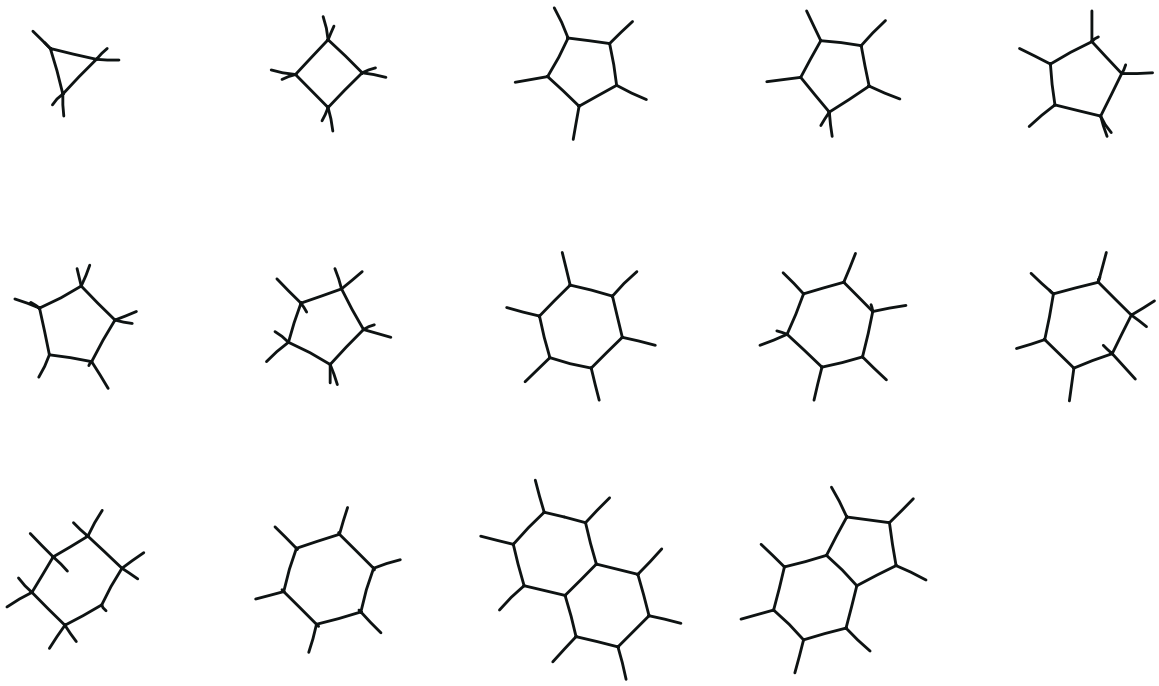


Fig. 3. The set of 14 templates used in runs 10–22.

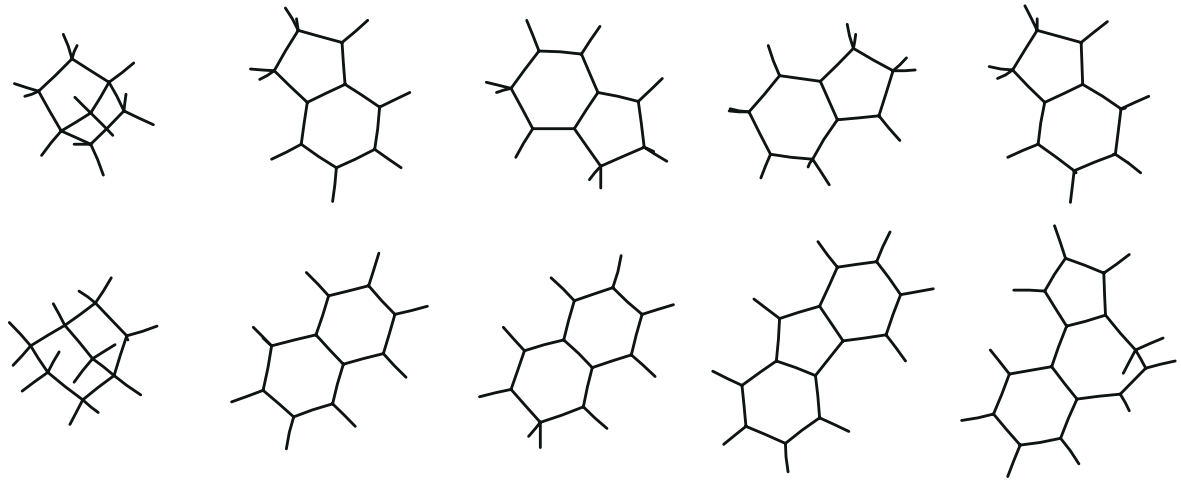


Fig. 4. The 10 additional templates used in runs 13–22.

Template sets

A number of template sets and joining rules can be defined and used in combination to control the range of molecules which are searched. Ring systems are predefined. The ring-system complexity is one of the most elusive and difficult factors to control when fusion-, bridge- and spiro-structures are built by template joining. Here it is not an issue; the decision about which systems are acceptable and which are not is made before the run is initiated.

The division of the template sets into default (0th) and non-default is not essential. All template sets could be treated in a uniform manner; the minimum and maximum number of templates present in the structure could be specified for each template set individually. This would simplify the concept and broaden the possibilities for structure connectivity control.

Hydrocarbon generic templates

Hydrocarbon generic templates are normally used. Templates containing heteroatoms are handled consistently and atom-type and bond-order information is retained throughout, although not used in the scoring function. Because atom types can easily be corrected during secondary structure generation [8,9], it is reasonable to explore all possible matches regardless of atom type. Otherwise, for example, carbon atoms would not be considered as possible matches to hydrogen-bonding groups.

We are investigating other forms of the objective function where atom types are used explicitly. The advantage would be that the chemical characteristics of interacting atoms would be consistent (e.g. H-bond donor against H-bond acceptor) without reassignment. Reactive, toxic and other undesirable combinations of atoms, however, should be considered in the objective function. The optimization could be frustrated and the effect on efficiency is unclear. Nevertheless, it would be interesting to compare different objective functions. When atom types are taken into account in the scoring function, it would still be advisable to examine alternative atom placements [8,9] allowing one to retain properties essential for binding.

It appears, from the work of Böhm [37], that a reasonably good accuracy can be achieved in predicting affinity constants even at such a crude level of detail. The approximation is derived by taking into account hydrogen bonds, ion bridges buried surface area and rotatable bonds. The full utilization of the atom-type information would make it possible to calculate atom-type dependent chemical properties such as charges, electrostatic potentials, etc. and make the calculation more reliable. As an alternative, a hydrocarbon set of templates could be used and an atom assignment procedure [2,36] invoked each time a template is changed.

Single joining operation

We use one very simple joining operation and relatively large fragments are used. This has certain advantages and disadvantages.

Well-refined templates taken from a library are expected to be more accurate than the same templates built on the fly. This also makes the algorithm faster since they do not have to be rebuilt repetitively. The program code is much simpler; there are fewer cases and exceptions. This makes the program more reliable. Greater structure complexity control can be exercised. Various ring systems of interest are predefined and there are no super-complex molecules produced in abundance.

On the negative side, a larger fragment database should be maintained which is difficult to be exhaustive. Also, since the pieces are larger it would be more difficult to position them in the site because of fewer acceptable modes; this may somewhat bias the algorithm towards using the smaller templates.

Penalty function scoring and optimization

The objective function accommodates very diverse

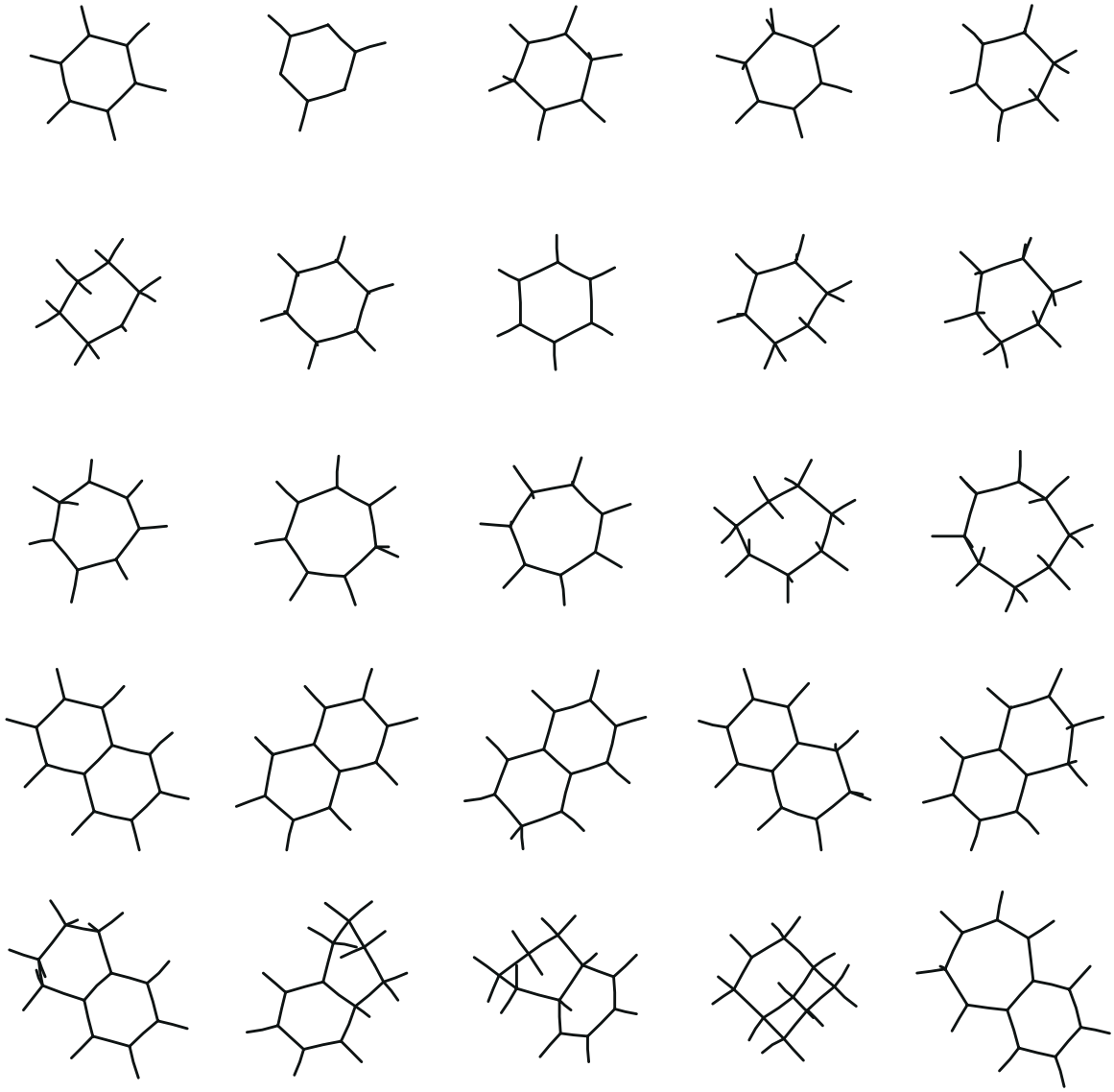


Fig. 5. The 25 additional templates used in runs 14–25.

requirements. Affinity is only one of the aspects to be considered. Apart from that, it also has to account for selectivity and chemical complexity considerations. Moreover, it is difficult to quantify objectively the relative contributions, and to interpret the meaning of the global minimum, when a set of miscellaneous unrelated constraints are used. Some constraints are softer and some are harder.

A penalty function allows reasonable results without fine adjustment and justification of weighting coefficients. It has a degenerate global minimum equal to zero for solutions. New constraints, such as the number of stereocentres, can easily be included. The disadvantage is that this requires more effort to set the relevant constraints for a particular receptor site.

Inclusion and exclusion sphere queries are incorporated into the objective function, and by using flexible notation the structure generation can be directed and more focused. Simulated annealing is a reliable optimizer; it converges asymptotically to the global minimum. The objective functions we use are degenerate, so it is likely that each run will produce a different solution if many exist.

Rings

Several mechanisms are used to control the incorporation of rings into the structures. First, rings are already present in some of the templates we use. This, combined with the possibility of defining different template sets from which different parts of the ligands are generated, provides a good way of controlling how many and which rings are incorporated. We believe that the algorithm is immune to a further increase in the number of sets and the number of templates. In fact, there is no need to

increase significantly the sets containing acyclic templates as used in the MTX test. Also, two sets of ring templates are more or less sufficient. One or two more sets could be added, for example, containing both acyclic and ring templates with the ratio set to approximately one half. Second, new rings can be formed on the fly during the annealing. This is an option which we rarely use. Third, it is possible to add more rings to each of the solutions in a separate postprocessing stage [38]. Taken to the extreme, this option allows one to use only a few acyclic templates for the annealing. This is efficient, but the ring positions cannot be specified and it is not possible to say whether rings can be placed appropriately at the second stage. The conformations of only a small number of them would allow one to generate the ring system. Of course, it is possible to have the ring system properly positioned and then generate acyclic chains which are constrained by ligand points to one bond (or one of several bonds) in that system. Alternatively, a few ring templates can be included among the acyclic ones and then ring-centre constraints can be used. The first of the three options is the simplest.

Acknowledgements

The authors wish to thank Rhône-Poulenc Rorer (N.P.T.) and the Wellcome Trust through the PRF scheme (P.M.D.) for personal financial support. Part of this work was carried out in the Cambridge Centre for Molecular Recognition funded by BBSRC. Drawings were produced with the SP3 program written by Dr. T.D.J. Perkins.

References

1 Dean, P.M., Barakat, M.T. and Todorov, N.P., In Dean, P.M., Jolles, G. and Newton, C.G. (Eds.) New Perspectives in Drug Design, Academic Press, London, U.K., 1995, pp. 155–184.
2 Barakat, M.T. and Dean, P.M., J. Comput.-Aided Mol. Design, 9 (1995) 341.
3 Martin, Y.C., J. Med. Chem., 35 (1992) 145.
4 Blaney, J.M. and Dixon, J.S., Perspect. Drug Discov. Design, 1 (1993) 301.
5 Slater, P.E. and Timms, D., J. Mol. Graphics, 11 (1993) 248.
6 Verlinde, C.L.M.J. and Hol, W.G.J., Structure, 2 (1994) 577.
7 Lewis, R.A. and Leach, A.R., J. Comput.-Aided Mol. Design, 8 (1994) 467.
8 Lewis, R.A. and Dean, P.M., Proc. R. Soc. London, B236 (1989) 125.
9 Lewis, R.A. and Dean, P.M., Proc. R. Soc. London, B236 (1989) 141.
10 Lewis, R.A., J. Comput.-Aided Mol. Design, 4 (1990) 205.
11 Lewis, R.A., Poe, D.C., Huang, C., Ferrin, T.E., Langridge, R. and Kuntz, I.D., J. Mol. Graphics, 10 (1992) 66.
12 Lewis, R.A., J. Mol. Graphics, 10 (1992) 131.
13 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 593.
14 Pearlman, D.A. and Murcko, M.A., J. Comp. Chem., 14 (1993) 1184.
15 Leach, A.R. and Kilvington, S.R., J. Comput.-Aided Mol. Design, 8 (1994) 283.
16 Tschinke, V. and Cohen, N.C., J. Med. Chem., 36 (1993) 3863.
17 Nishibata, Y. and Itai, A., Tetrahedron, 47 (1991) 8985.
18 Moon, J.B. and Howe, W.J., Protein Struct. Funct. Genet., 3 (1991) 681.
19 Gillet, V., Johnson, A.P., Mata, P., Sike, S., Zsoldos, Z. and Johnson, A.P., J. Comput.-Aided Mol. Design, 7 (1993) 127.
20 Rotstein, S.H. and Murcko, M.A., J. Comput.-Aided Mol. Design, 7 (1993) 23.
21 Rotstein, S.H. and Murcko, M.A., J. Med. Chem., 36 (1993) 1700.
22 Bohacek, R.S. and McMartin, C., J. Am. Chem. Soc., 116 (1994) 5560.
23 Clark, D.E., Frenkel, D., Levy, S.A., Li, J., Murray, C.W., Robson, B., Waszkowycz, B. and Westhead, D.R., J. Comput.-Aided Mol. Design, 9 (1995) 13.
24 Glen, R.C. and Payne, A.W.R., J. Comput.-Aided Mol. Design, 9 (1995) 181.
25 Goodford, P.J., J. Med. Chem., 28 (1985) 849.
26 Danziger, D.J. and Dean, P.M., Proc. R. Soc. London, B236 (1989) 101.
27 Miranker, A. and Karplus, M., Protein Struct. Funct. Genet., 11 (1991) 29.
28 Dean, P.M. (Ed.) Molecular Similarity in Drug Design, Blackie, London, U.K., 1995.
29 Nilsson, N.J., Principles of Artificial Intelligence, Springer, Berlin, Germany, 1982.
30 Kirkpatrick, S., Gellatt Jr., C.D. and Vecchi, M.P., Science, 220 (1983) 671.
31 Goldberg, D.E., Genetic Algorithms, Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, U.S.A., 1989.
32 Morley, S.D., Abraham, R.J., Haworth, I.S., Jackson, D.E., Saunders, M.R. and Vinter, J.G., J. Comput.-Aided Mol. Design, 5 (1991) 475.
33 Van Laarhoven, P.J.M. and Aarts, E.H.L., Simulated Annealing: Theory and Applications, Reidel, Dordrecht, The Netherlands, 1987.
34 Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, Cambridge, U.K., 1986.
35 Bolin, J.T., Filman, D.J., Matthews, D.A., Hamlin, R.C. and Kraut, J., J. Biol. Chem., 257 (1982) 13650.
36 Gillet, V., Newel, W., Mata, P., Myatt, G., Sike, S., Zsoldos, Z. and Johnson, A.P., J. Chem. Inf. Comput. Sci., 34 (1994) 207.
37 Böhm, H.-J., J. Comput.-Aided Mol. Design, 8 (1994) 243.
38 Leach, A.R. and Lewis, R.A., J. Comput. Chem., 15 (1994) 233.
39 Meng, E.C., Shoichet, B.K. and Kuntz, I.D., J. Comput. Chem., 13 (1991) 505.
40 Morgan, H.L., J. Chem. Doc., 5 (1965) 107.
41 Barnard, J.M., In Ash, J.E., Warr, W.A. and Willet, P. (Eds.) Chemical Structure Systems. Computational Techniques for Representation, Searching, and Processing of Structural Information, Ellis Horwood, New York, NY, U.S.A., 1991, pp. 9–56.
42 Perkins, T.D.J. and Pean, P.M., J. Comput.-Aided Mol. Design, 7 (1993) 155.
43 Oshiro, C.M., Kuntz, I.D. and Dixon, J.S., J. Comput-Aided Mol. Design, 9 (1995) 113.

Appendix

Note 1

In this note we describe the various terms in the objective function.

Distance term F_1

The requirement for atoms in the structure to occupy, or be excluded from, certain positions in space is specified by spherical shells in a manner very similar to the definition of site and ligand points [8,9]. We will use these terms interchangeably, although they may not be exactly equivalent. The position of the centre, the inner radius, r'_i, and the outer radius, r''_i, of each spherical shell, i, are specified.

Unions of spherical shells

We define $(a_1 + a_2 + \cdots + a_m)_n$ as the sum of the least n elements from the array a_i consisting of m elements for $n < m$ and a normal summation if n is not specified or $n \geq m$. For example, $(1+2+3+4)_2 = 3$, $((1+2)+3+4)_2 = 6$, $((1+2)+(3+4))_2 = 10$, $((1+4)+2+3)_2 = 5$. Furthermore, we define

$$f^n_{i_1,\ldots,i_m} = \left(\sum_{j=1}^{N} \left(\sum_{k=1}^{m} f\left(r_{i_k j}, r'_{i_k}, r''_{i_k}\right) \right) \right)_n \qquad \text{(A1)}$$

where N is the number of relevant atoms in the ligand (see the next section), n is the number of ligand atoms required to satisfy all constraints defined by spherical shells $i_1,\ldots,i_m$, $r_{i_k j}$ is the distance between sphere i_k and ligand atom j, and r'_{i_k} and r''_{i_k} are the inner and outer radii of the spherical shell i_k. If the superscript n is equal to one, it can be omitted.

The ligand should have a specified number of atoms, n, inside the union of all shells $i_1,\ldots,i_m$. To calculate the sum for $n=1$, $m=1$, from all atoms in the ligand (N), the atom ($n=1$) is found which least violates the constraint imposed by the only spherical shell ($m=1$) and the corresponding penalty is added to the objective function. If, for example, an atom is required to be hydrogen bonded to a protein atom, a shell can be centred at the protein atom, the inner radius set to 2.5 Å and the outer one to 3.5 Å. The proposed structures must then have an atom that is inside that shell.

Atom subset specification

Terminal and non-terminal atoms can be distinguished and only the relevant ones are used to match a particular union of spheres. Ring centres can be used in this manner.

Before the score is calculated, the lengths of the bonds that involve hydrogen atoms are made comparable to the lengths of the bonds between non-hydrogen atoms (a value of 1.4 Å is used). Thus, the search for matching atoms is extended one level further, which improves the efficiency. Care should be taken to keep track of which of these hydrogen atoms do match some of the spheres and actually affect the value of F_1. These are treated as non-hydrogen atoms in the calculation of other terms of the objective function.

Thus, the positions of hydrogen atoms are not constrained by inclusion and exclusion spheres. This could be modified by using both the original and the modified position for the H atom and testing constraints on H atoms with the first set of coordinates and constraints on non-H atoms with the second one.

Fine- and coarse-grained queries

It is possible to search simultaneously for structures which satisfy one of several ligand-point sets. Coarse-grained queries are composed of several fine-grained alternatives. Here, we give several examples of how coarse-grained queries are defined.

Example 1

Consider two pairs of ligand points and let one matched point be required from each pair. If in (i,j), i is a point from the first pair (1 or 2) and j is a point from the second one (3 or 4), there are four ways, (1,3), (1,4), (2,3) and (2,4), in which a complete match can be achieved. Four different queries can be defined and searched for. If, however, there is no preference for one match over another, the search can be performed over the four queries at the same time. If F_1 is defined as $F_1 = (f_1 + f_2)_1 + (f_3 + f_4)_1$, it becomes zero if and only if one of the four sets is matched. Which one is not known beforehand and the distribution between the four sets will depend on their accessibility. The chance of finding a solution is greater than the case where only one query is searched for.

Alternatively, the four constraint sets could be coded separately; a constraint-violation penalty is calculated for each of them; and the minimum of the four values is considered: $F_1 = ((f_1 + f_3) + (f_1 + f_4) + (f_2 + f_3) + (f_2 + f_4))_1$. This form allows for more general queries to be used. If, for example, only three out of the four sets are of interest, it is possible to distinguish between them, while with the previous definition it is not.

Example 2

Consider another example. Let there be 10 ligand points; then a selection of any five of them defines a solution. This can be coded as a sum of 252 terms $F_1 = (f_{1,2,3,4,5} + f_{1,2,3,4,6} + \cdots + f_{6,7,8,9,10})_1$. Alternatively, however, the summation could be just extended over the five least violated constraints $F_1 = (f_1 + \cdots + f_{10})_5$.

Recognized expressions and input format

The input format of our current implementation allows one to evaluate expressions of the form

$$F_1 = \sum\left(\sum\left(\sum f^n_{i_1,\ldots,i_m}\right)\right)_k \qquad \text{(A2)}$$

with cases $n = 1$ and $n = N$ in Eq. A1 considered.

Ligand refitting

When the atoms in the ligand which match the spherical-shell constraints have been determined, the atom and shell-centre pairs can be used to refit the structure. F_1 is recalculated after the refit. This procedure could be extended to perform several iterative refits. By default, only the initial structure is refitted and a value of 1.00 is used for the weighting factor w_1.

Bond-path term F_2

The number of bonds between the atoms that match unions of shells is found and a penalty of +1 is added if that number is outside the specified limits.

Example

If a bond vector is specified by two spheres, then it may be helpful to constrain to one the number of bonds between the atoms matching each sphere. Otherwise, if tolerance radii of the two spheres are large, various undesirable alternatives may be suggested. A default value of 1.00 is used for the weighting factor w_2.

Repulsive intermolecular steric term F_3

The shape of the site is also taken into account. A cuboid that encloses the site atoms is defined, and the block of space within it is mapped onto a regular Cartesian grid. A shell with controllable thickness is constructed around the receptor. First, the lattice points that are within its outer surface are labelled. The atomic radii used in this operation are calculated as a linear function, $ar_{\mathrm{acc}} + b$, of the atom-type dependent radii, r_{acc}, of the receptor atoms, where a and b are parameters. In a second pass, smaller atomic radii, $cr_{\mathrm{acc}} + d$, are used and the points inside that surface are unlabelled. The shell thickness changes by altering the coefficients. It is also possible to specify radii individually for each receptor atom which are used without modification. This procedure can be used to generate lattice maps both for protein sites and for sites derived from active-compound similarity superpositions. The default parameters in the first case are $a = 1.0$, $b = 4.5$ Å, $c = 1.0$, $d = 0.0$ Å, $r_{\mathrm{acc}} = 3.0$ Å for C, S and P and $r_{\mathrm{acc}} = 2.5$ Å for N and O atoms (H atoms are ignored). In the second case they are $a = 0.0$, $b = 1.5$ Å, $c = 0.0$, $d = 0.0$ Å, $r_{\mathrm{acc}} = 0.0$ Å (H atoms are ignored).

Next, the points situated at the surface of the accessible volume are identified. A breadth-first search is then used to find the number of lattice steps required to reach a point lying outside the surface from the nearest surface point. This number is multiplied by the grid step and the result is stored for each point. This value approximates the distance to the site surface and is used as a penalty, accumulated over all ligand atoms lying outside the surface. If an atom protrudes beyond the lattice box, the distance to the nearest lattice point on the box surface is found and added to the penalty value stored at that point. A default value of 0.50 is used for the weighting factor w_3.

Attractive intermolecular steric term F_4

A contact score [39] is calculated at each lattice point which is inside the accessible volume. A value of +1 is added to the score of a lattice point for each receptor atom within a specified distance range. The contact score of an atom from the ligand is set to the value stored at the nearest grid point. The total contact score of the ligand is the sum of contact potentials of non-hydrogen atoms in the skeleton lying inside the accessible surface. A default value of 0.10 is used for the weighting factor w_4.

Intramolecular steric term F_5

The distances between non-bonded atoms in the structure are required to be above a specified minimum, r_{vdW}, and the amount by which r_{vdW} is violated is accumulated as a penalty. Expressed in terms of f, this term reads as

$$F_5 = \sum_{i>j} f\left(r_{ij}, r_{\mathrm{vdW}}, \infty\right) \qquad \text{(A3)}$$

where the summation is taken over all pairs of non-hydrogen atoms i and j. A single cutoff value (the default value is 2.5 Å) is used for all non-hydrogen atoms. Hydrogen atoms are ignored. A default value of 0.50 is used for the weighting factor w_5.

Torsion energy term F_6

An upper limit, $e_{\max}$, of the torsion energy, e, is set and a penalty term, $f(e,0,e_{\max})$, is added to the objective function. The torsion energy function and parameters of the COSMIC force field [32] are used to calculate e. By default, this term is not calculated, but when it is included a default value of 0.01 is used for the weighting factor w_6.

Number of atoms F_7

Non-hydrogen atoms are counted and a linearly increasing penalty is added if the number is outside the set lower and upper bounds. A default value of 1.00 is used for the weighting factor w_7.

Number of rings term F_8

A similar penalty term is used to confine the number

of rings within specified limits. A default value of 1.00 is used for the weighting factor w_8.

Template copies term F_9

The number of copies of each template incorporated into the structures can be restricted between a minimum and maximum value with linearly increasing penalties for an insufficient or redundant number of copies. A default value of 1.00 is used for the weighting factor w_9.

Template deviation from original position F_{10}

The average deviation is calculated for the atoms of each template from the original positions in the template file. If the deviation is larger than the amount specified for the template (the calculation is not performed if a negative value is specified), a linearly increasing penalty is added to the score. The incorporation of this term makes it possible to direct templates in the structure near predefined positions in the site where they would achieve a favourable interaction.

For each atom in the structure, the atom in the prepositioned template used to calculate its contribution to the average deviation can be selected in three ways. It can be the nearest atom, the nearest symmetrically equivalent atom or the atom with the same number. To clarify the second option we shall mention that, for example, all carbon atoms in benzene are viewed as symmetrically equivalent. A default value of 5.00 is used for the weighting factor w_{10}.

Bond order mismatches F_{11}

A penalty of +1 is added to the objective function for each bond which has a different bond order specification at each of its constituent atoms. This can occur when joins between fragments are not checked for consistency. This term is one of the mechanisms used to ensure that the final structure contains no bond order mismatches.

Simplex optimization

After the score is calculated, the Metropolis condition is checked and the structure is updated. When occasionally a structure with a score lower than the best obtained so far is found, its score is further optimized by the simplex method [34] by performing rigid-body rotations, translations and bond rotations. This procedure speeds up convergence in the region of low scores where simulated annealing is not very good at finding exact local minima. Simplex minimization is only attempted if the score is below a specified threshold (the default value is 5.0), since high-score structures are more unlikely to yield a solution. The structure can also be optimized after each transition move before the Metropolis condition is checked, or if the transition has been accepted; however, these two options slow the algorithm and do not improve its convergence (their default score-threshold values were set to −1.0). The other default values for the simplex method are: a maximum of 200 iterations; simplex parameters of 1.0 Å for translational, 90° for rotational and 180° for torsional degrees of freedom; and convergence criteria of 0.01 for the difference of either the rms of the variables or the function values in two successive iterations.

Note 2

Canonicalization

We use a modification of the canonicalization algorithm proposed by Morgan [40]. There are two stages in this method. During the first stage, the atoms are partitioned into classes by comparing their environments. This is done iteratively by progressively taking into consideration more and more distant atomic layers. An initial value equal to the number of incident bonds, and called extended connectivity, is assigned to each atom. At each iteration this value is replaced by the sum of the values of its neighbours. After the iteration is completed, the number of different labels currently in use is found. The process continues until this number no longer increases. In the second stage, the atoms are renumbered. Number 1 is awarded to the atom that finishes with the highest value from the first stage. Its neighbours are assigned the next numbers in decreasing order of their extended connectivities. The same procedure is applied to assign numbers to the neighbours (not yet numbered) of atoms 2, 3, and so on. Conflicts occur when two atoms with the same extended connectivity values compete for the next number. These problems are resolved by selecting the atoms with lower element numbers, or some other property that can be used to set the priority. In certain cases, the algorithm fails; then the extended connectivity values can oscillate between successive iterations, or atoms that are not symmetrically equivalent cannot be distinguished (i.e. they have the same extended connectivity at the end of stage one) [41].

To avoid this, a modification has been implemented. A vector composed of the old values of its neighbours is formed for each atom and used to assign new extended connectivity values. The number of vector components is set to the maximum number of neighbours encountered and, where necessary, the vectors are padded with zeros. The components are ordered in increasing extended connectivity values. Two vectors are compared, starting from the smallest elements and proceeding to the next ones, only if the values are equal. One vector is larger than another if the first non-equal element of the first one is larger than the corresponding element of the second one. Vectors are ordered and consecutive integer numbers are assigned to the unique vectors. These numbers are used as new extended connectivity values. The second stage of the original algorithm is performed without alteration.

Note 3

Search parameters

The algorithm depends on two sets of parameters. The first set reflects various structural constraints and is used to calculate the objective function. These *functional parameters* help the algorithm to distinguish an acceptable solution from a non-solution and to estimate how far a non-solution is from a solution. We have given default values for some functional parameters. Such parameters, for example, are the weighting factors, w_i; others are dependent on the protein site being studied (the set of spherical shells, for example) and are supplied by the user. These parameters are considered in the next section. The second set is composed of *search parameters*, which affect the search performance of the algorithm once the structural constraints are defined; they are not used in the calculation of the objective function. The aim, when the search parameters are varied, is to find values for which the number of function evaluations per solution is minimal. The default parameters we use will work well for many problems, but are probably optimal for none. There are several reasons which make us reluctant to insist on the specific default values as universal search parameters. First, each set of functional parameters defines a different objective function and extrapolation to other functions may be unreliable. Second, the number of search parameters is large; this makes it quite difficult to examine their values systematically even for a limited number of functional-parameter sets. Perhaps, another optimization in parameter search space is required to tune the search. Third, the structure generation algorithm uses random numbers. A sufficiently large number of runs have to be performed to estimate reliably the average number of function evaluations per successful structure.

Functional parameters

In this section we describe several strategies for defining constraints for structure generation. These are well known, but it may be helpful to recapitulate these ideas. We also outline how various options implemented in the program can be utilized to control the diversity of the generated structures. These possibilities are not immediately apparent from the previous description.

Hydrogen-bond constraints

Consider the problem of finding ligands which form a hydrogen bond with a particular protein atom. There are several possibilities of how this can be specified. First, a spherical shell can be defined and centred at the protein atom with radii 2.5 and 3.5 Å; additional constraints can be used to account for directionality. Second, the positions of atoms from known ligands forming the desired bonds can be used to define the constraints. Third, complementary points with a high probability of forming hydrogen bonds could be found algorithmically and used as constraints. One possibility is to use the positions of the group minima from the GRID program [27]. Fourth, single molecular templates can be docked to satisfy these constraints; terminal bonds from templates which are successfully docked can be coded as sets of two spheres themselves and generate solutions matching the indirect query. Fifth, docked templates can also be included in a separate set and used to generate structures which will always contain one of them. The functional term, F_{10}, can be used to direct the algorithm to position templates near their original position, i.e. where they were docked in the site. If there is only one set of templates, the ligands can be positioned so that the template which is from this set is initially placed at its docked position.

Metal ions

Metal ions can be treated in a similar manner, with a minor peculiarity. When the lattice map of the site is generated, a radius of 1.5–2.0 Å is used for the metal to allow the ligand to approach the metal ion closely; ligand points are then selected appropriately.

Water molecules

The presence or absence of bound water molecules changes the shape of the site. Water molecules can form hydrogen bonds with the receptor and be found in internal cavities. Different models of the same site, with varying numbers and positioning of the water molecules, can be examined. If there are n water molecules, then the number of modified sites is 2^n when a subset of the water molecules is used. The empty site without any water molecules is used to outline the available space. Consider a single water molecule. Two sets of constraints are defined. The first set is satisfied if an atom from the ligand is positioned in the immediate vicinity of the water molecule. Thus, the presence of the water molecule is ignored and the ligand forms direct hydrogen bonds with the complementary protein atoms. One way to specify this is by a single sphere with a centre at the position of the water molecule; the corresponding term is labelled f_1. The second set is satisfied if a ligand atom is able to form a hydrogen bond with the water molecule, and all other ligand atoms are at a respectable distance away from the water molecule. In this case, the water molecule is considered part of the site. One sphere or spherical shell, f_2, is used to specify that one atom should form a H-bond with the water molecule and another one, f_3^N, to preclude the other ligand atoms from approaching the water molecule too closely (since the water molecule is not included in the site-shape definition). The functional term can then be defined as $F_1 = (f_1 + (f_2 + f_3^{\mathrm{N}}))_1$. Similar terms are added if more water molecules are considered. By this procedure, all 2^n water-modified sites can be explored simulta-

neously. The proportion of the structures generated for each depends on its accessibility.

Docking of molecules

A molecule can be docked into its site as a rigid structure. A sample conformation of the molecule can be included as the only fragment in the template library. Only rigid-body translations and rotations are allowed and structures composed of one fragment are generated. This is equivalent to rigid docking of the molecule. Within the same setting, the template can be specified as flexible and bond rotations allowed; this would correspond to a flexible docking procedure.

Another popular method to account for flexibility is to use several representative rigid conformers [42]. These could be docked in separate runs with or without allowing small conformational changes during the annealing. When such changes are allowed, conformations can be explored in the vicinity of (or even far from if enough transitions are generated) the initial structures. This would ensure that conformations 'between' the samples can be better approximated. One other possibility is to include all representatives as templates in a single run. Here again flexibility in the representatives may be allowed as well.

Furthermore, conformers of several different compounds can be sampled during the annealing [43] in a similar manner. Thus at the end, a conformer of one of the compounds will eventually satisfy all constraints. If annealing is repeated several times, possible models will be generated for different compounds docked in the site. The algorithm will select the compounds for which a solution can be generated from those for which this is not possible.

In all the cases considered so far, ligands with only one fragment in them have been used. A molecule can be considered as composed of, say, two moieties. Two template sets can then be defined, one for each moiety. In each set there is either one or several representative conformers of the substructure. Bond-order labels are used to define how to connect the templates from different sets together. Each solution will be a combination of representative templates of each moiety. Again flexibility can be allowed for the representative templates during the annealing. In addition, if all terminal atoms provided by the substructures are used, solutions with a connectivity different from the original can be generated. Cases with more than two substructures can be treated in the same manner.

Synthetically simple molecules

In each set, templates can be included which are not just conformations of the original substructure, but have different connectivity. Connections can be defined at positions corresponding to a particular chemical reaction. Solutions are then generated in a combinatorial manner from templates from each set and complying with specified reaction mechanisms. This will ensure their synthetic accessibility.