Journal of Computer-Aided Molecular Design, 9 (1995) 381–395
ESCOM

J-CAMD 308

PRO_LIGAND: An approach to de novo molecular design. 6. Flexible fitting in the design of peptides

Christopher W. Murray*, David E. Clark and Deirdre G. Byrne

Proteus Molecular Design Ltd., Proteus House, Lyme Green Business Park, Macclesfield, Cheshire SK11 0JL, U.K.

Received 25 June 1995
Accepted 12 July 1995

Keywords: De novo molecular design; Peptide design; Directed tweak algorithm; Force field calculations

Summary

This paper describes the further development of the functionality of our in-house de novo design program, PRO_LIGAND. In particular, attention is focussed on the implementation and validation of the 'directed tweak' method for the construction of conformationally flexible molecules, such as peptides, from molecular fragments. This flexible fitting method is compared to the original method based on libraries of prestored conformations for each fragment. It is shown that the directed tweak method produces results of comparable quality, with significant time savings. By removing the need to generate a set of representative conformers for any new library fragment, the flexible fitting method increases the speed and simplicity with which new fragments can be included in a fragment library and also reduces the disk space required for library storage. A further improvement to the molecular construction process within PRO_LIGAND is the inclusion of a constrained minimisation procedure which relaxes fragments onto the design model and can be used to reject highly strained structures during the structure generation phase. This relaxation is shown to be very useful in simple test cases, but restricts diversity for more realistic examples. The advantages and disadvantages of these additions to the PRO_LIGAND methodology are illustrated by three examples: similar design to an alpha helix region of dihydrofolate reductase, complementary design to the active site of HIV-1 protease and similar design to an epitope region of lysozyme.

Introduction

Computer-aided molecular design (CAMD) is now recognised as being an important component of pharmaceutical research. When applying CAMD to drug design, two possible scenarios can be envisaged. In *direct* drug design, one has knowledge of the structure of the biological target and seeks to design molecules which interact favourably with (or mimic well) particular parts of the target. By contrast, in *indirect* drug design, one typically has a set of active and inactive molecules and tries to design a molecule which has enhanced activity. For either scenario there is a wealth of computational methods to aid the design process.

One particular CAMD method which has received a lot of interest recently is de novo design, in which candidate molecules are constructed automatically from smaller molecular fragments or single atoms (for recent discussions, see Refs. 1 and 2). There are now so many methods of this type that it would be a considerable task to describe them all, but they can be classified according to how the building process is performed. Some programs control the building process with reference to a molecular mechanics-based energy function, reflecting how well the designs interact with the biological target. Programs of this type build up their structures from atoms [3–7] or smaller molecular fragments [8–12]. Other programs use a rule-based approach to derive positions in space where particular chemistries could reside. Structures are then built up, taking care to put favourable chemistries in the correct regions of space. Methods of this type use either real chemical fragments [13–28] or representative templates [29–32] as building blocks. Of course, even with these broad categories there is some overlap between

*To whom correspondence should be addressed.

certain methods and some strategies remain difficult to classify [33–35].

Each of the two broad approaches to de novo design has its advantages and disadvantages. Energy-based methods allow the possibility of a more rigorous assessment of partial designs as they are being built and therefore offer the hope of focussing on the most promising designs. However, energies of interaction between ligands and their receptors are difficult to calculate accurately and using these values to discriminate between structures during the building process can be perilous. Rule-based methods are usually faster and can be applied to a wider range of design problems, such as indirect design, where it might be difficult to calculate energies of interaction.

PRO_LIGAND is our in-house facility for automated molecular design [25–28,36] and forms part of the PROMETHEUS system for molecular design and simulation. It is essentially a rule-based method and, of the other de novo design methods, is most similar to the program LUDI [13]. PRO_LIGAND uses rules to define interaction sites from a receptor or set of active analogues. The interaction sites represent the 3D configuration of physicochemical features that the designed ligand should possess. The set of interaction sites is referred to as the design model and can be thought of as an approximation to an 'ideal drug template'. PRO_LIGAND then selects fragments of appropriate chemistries from prestored libraries and fits the fragments together to form the designs, which may be either peptides or more general organic molecules. In common with other rule-based methods, the fragments are treated as rigid entities, which is obviously a gross approximation for conformationally flexible fragments. In PRO_LIGAND this problem is alleviated somewhat by storing more than one conformation for such flexible fragments. In a previous paper, we have described how libraries of conformations are employed in the building of peptides [28], presenting as an example the building of plausible peptide structures complementary to the active site of HIV-1 protease. This achievement demonstrated that the rule-based approach was a worthwhile strategy for conformationally flexible peptide molecules which, up to that point, had only been treated using energy-based criteria. However, the method had a number of disadvantages, primarily due to its reliance on libraries of prestored conformations. Perhaps the main difficulty with prestoring conformations is achieving a compromise, so that a set of conformations is obtained that is sufficiently small to make the process computationally feasible and sufficiently large to allow conformational space to be spanned adequately.

The purpose of this paper is to present a new approach to the fitting of flexible fragments or templates in de novo design. To some extent, the problem of flexible fragments has been faced already in 3D database software where, at least until recently [37,38], there had been growing consensus that flexible fitting algorithms are superior to a library of conformations [39]. In the case of de novo design, the decision is not quite as simple, since the number of conformations needed is governed by the number of rotatable bonds in the library fragments and the type of fragment included in the library is to some extent controlled by the user when the library is constructed. However, even in de novo design, a number of groups have chosen to use fairly large libraries containing sizeable fragments, which puts them in a similar position to database researchers [17,23]. Thus, we believe that fragments that require a significant number of low-energy conformations may be best fitted by an algorithm that takes account of their flexibility, rather than by treating each low-energy conformation as a rigid fragment.

A number of different approaches to flexible fitting have been described in the 3D database literature [40,41]. We have chosen to use the method described by Hurst [42], which has been demonstrated to be one of the most efficient and effective approaches [40]. Hurst's 'directed tweak' method is a variant of the random tweak algorithm that is used to generate possible polypeptide loop conformations subject to constraints on the first and last peptide in the loop sequence [43,44]. It should perhaps be pointed out that tweak methods have been previously applied in de novo design. Leach and Kilvington have used a variant of the random tweak method for joining previously placed fragments together with aliphatic chains [22]. Our implementation differs from that of these authors because the use of the directed tweak method allows our formulation to be made into a general method for placing and joining fragments, with no restriction on the number of sites that may be fitted using the procedure.

A further methodological advance that is put forward in this paper is the use of force field calculations to relax the fragments after they have been placed. Most programs that use rule-based (i.e., interaction site-based) building procedures treat the fragments as rigid entities even after their placement. No attempt is made to relax the geometries 'on the fly' and once solutions have been produced, the problem of their conformational stability is addressed as a post-processing step. This is a good approach with rigid or reasonably rigid structures, but it is less defensible with flexible molecules such as peptides. We have therefore used the AMBER force field [45] to perform a constrained optimisation of placed fragments upon the interaction sites which they hit in the design model.

The flexible fitting and force field relaxation codes are written in a general way, so that the methods described here are equally applicable to organic or peptidic fragments. In this paper, however, we will concentrate on the design of peptides, since for this class of molecules it is imperative that the problem of flexibility be addressed. We would not expect a significant degradation in per-

formance on applying our code to the design of flexible organic fragments, although the number of structures that would need to be generated to provide a reasonable coverage of the solution space would probably be greater. The choice of peptides also allows us to use the standard AMBER force field without a lengthy consideration or description of our parameters, since these will certainly be accurate enough for our purposes. Oligopeptides can be attractive candidates during the first stages of drug design, since they are easily synthesised [46]. Peptides are also attractive candidates in the design of synthetic vaccines that mimic discontinuous epitopes. In both cases, initial peptides could be used as starting points for the construction of combinatorial libraries which would allow rapid experimental improvement of the designs [47,48].

The next section details the methods used in this paper and is split into four subsections. The first subsection contains a brief description of PRO_LIGAND, so that the paper can be read without prior knowledge of previous work. The following subsection goes into more detail concerning how structures are built in PRO_LIGAND. It also gives a short description of the use of conformational libraries for peptide design in PRO_LIGAND, which has been given in detail elsewhere [28]. The next subsection gives a description of our implementation of the directed tweak method. Finally, a description of the exact approach adopted to relax the peptide structures during the building process is presented.

The methods are then applied to three design problems and the outcome is described in the Results section. The first application is similar design to an alpha helix region of dihydrofolate reductase (DHFR) which primarily serves the purpose of demonstrating some problems with the methods. The second example is complementary design to the HIV-1 protease active site and the third is similar design to some sparse features in an epitope region of lysozyme. The latter two examples were used in our previous peptide design paper [28] to illustrate the use of libraries of conformers and were chosen here to allow an effective comparison of the old and new methodologies without extensive discussion of the absolute quality of the solutions. The Results are followed by a Discussion section, where the methods and their performance are assessed in a general sense. The last section presents the Conclusions drawn from the paper.

Methods

Overview of PRO_LIGAND

PRO_LIGAND currently comprises the following modules:

Design-base Generation takes molecular structures and a command file to produce a *design base*, which contains all the necessary structural information needed in the design process. It can extract the desired active-site atoms in a receptor or can derive a pharmacophore from a set of active molecules and conformations or a Molecular Field Analysis model [25,26].

Design-model Generation takes as input the design base, a rule file and a command file and produces a *design model* whose physicochemical characteristics are either similar or complementary to those of the design base, according to the user's choice. The design model is a template that describes the idealised steric and hydrogen bonding features of the chemical structures to be designed. These features are represented as *interaction sites* [13,14,49]. The rules used to derive the positions and types of interaction sites are derived from statistical analyses of nonbonded contact geometries found within the Cambridge Structural Database [50]. Hydrogen bond acceptor and donor sites are represented by **A–Y** and **D–X** vectors, respectively, while lipophilic regions are represented by **L** and **R** sites for aliphatic and aromatic regions, respectively. The rules used to generate the sites are encoded in the rule file and can be readily adapted to give the design model different characteristics [25].

Structure Generation uses libraries of predefined fragments with appropriate atoms marked as interaction sites to fit suitable chemistries on top of the design model [25,28]. This is the module that is used to build peptides and will be described in more detail below.

Structure Refinement takes sets of structures consistent with the design model and uses a genetic algorithm to mix and mutate the structures so as to produce structures which score more highly [27].

Structure Analysis takes sets of structures and uses a unified and varied set of tools to cluster, score and sort the structures according to user-defined criteria. This module is the subject of a further paper [36].

In design applications in our company, all five modules are run. The Structure Generation and Structure Analysis modules and, to a lesser extent, the Structure Refinement and Design Model modules are run repeatedly, in order to generate a representative sample of solutions. At various stages, promising solutions are further examined using traditional computational simulation techniques. The outcome of these procedures is a set of best designs, which are synthesised together with a few simple analogues of those designs.

Structure Generation

The purpose of this subsection is to provide background information about how structures are built in PRO_LIGAND, so that the methodological improvements of the following sections can be presented effectively. A brief outline of how peptides are built using libraries of conformers will be covered here as well [28].

The Structure Generation module searches through libraries of fragments, which in this case are peptide residues. The fragments are chosen randomly, although the

user has the ability to make sure that particular fragments or sets of fragments are selected for attempted fitting before others. The fragments have certain atoms labelled as interaction sites, for instance a C=O group might be labelled **A–Y** or a carbon atom in an alkyl chain might be labelled **L**, and it is these atoms that are matched to corresponding sites specified in 3D coordinate space in the design model. How the atoms are labelled is to some extent arbitrary and many different labelling schemes are possible. We employ different schemes in the same library file and these are searched through in a random order, unless the library file specifies certain schemes as being of higher rank than others. The peptide library is also split into three sections, reflecting the gross extent of labelling within those sections. The most extensively labelled library is appropriate for design models containing a high density of interaction sites (such as the HIV-1 protease example in this paper) and the more sparsely labelled library is more suitable for design models containing a lower density of sites (such as the lysozyme example in this paper). Until the development of our flexible fitting approach, multiple conformations were stored for peptides and other flexible fragments. These conformations were accessed in a random order.

Once a particular library fragment with a particular conformation and labelling scheme has been selected, the program attempts to fit the fragment interaction sites onto the design model. A distance matrix is constructed for the fragment interaction sites and this is matched (subject to user-defined distance tolerances) against a distance matrix for the design model, ensuring a one-to-one correspondence between the interaction site types. The matching process is accomplished using a rapid subgraph isomorphism algorithm [51,52], which is completely general and has been used widely for 2D and 3D database searching. The algorithm will detect all possible hits in turn and, since it is deterministic, it is necessary to periodically shuffle the interaction sites in the design model to prevent any bias. Once a hit is generated, the fragment is superimposed onto the selected design model sites and the hit is subjected to a series of checks. If the hit fails any of the checks, it is rejected and a new hit is searched for using the subgraph isomorphism algorithm until all hits have been detected by the algorithm or the number of detected hits exceeds a user-defined maximum. The checks employed for the hits are:

(1) a check to ensure the correct chirality of the hit – this is necessary because the subgraph isomorphism algorithm works on distance matrices only and since the fragments are treated as rigid, any fragment with more than three nonplanar interaction sites is 'pseudo-chiral';

(2) a check is made that the partially built structure does not contain any forbidden substructures, as specified by the user;

(3) a check is made that there are no clashes between the new fragment and any previous fragment (which cannot be resolved by bond formation);

(4) a check is made that the fragment does not impinge on any excluded regions of space, e.g., a clash with the receptor;

(5) after bond formation and correction of geometry about new bonds, a check is made that the new geometry for the built structure can still fit onto the interaction sites that constituted the hit.

A fragment that passes this series of tests is accepted and included in the growing structure. It is important to realize that any structure that passes these tests is immediately accepted. This constitutes a depth-first strategy to the combinatorial fragment placement problem. The depth-first strategy is the reason that the library fragments, the conformations and the different labelling schemes must be accessed in a random order, so that all areas of the solution space are accessible to the program, unless the user has specifically cut off those areas by introducing a ranking.

After the addition of every fragment, the design model and its distance matrix are amended by deleting design model sites that are close to the new fragment and by adding additional sites marking possible positions of attachment for new fragments onto the partially built structure. In the case of peptides, these join sites are chosen so as to ensure that only a polypeptide can be built. Different libraries of fragments are used at various stages in the program and these libraries have different numbers of join sites marked on the fragment. The place libraries have no join sites and some interaction sites, the place-join fragments have one join site and some interaction sites, the place-bridge fragments have more than one join site and some interaction sites, and the bridge fragments have more than one join site and no interaction sites. The order in which these libraries are used is controlled to a large degree by the user. This allows the user to specify a large number of different building strategies. To date we have found that a 'growing' strategy, where one fragment is placed and subsequent fragments are place-joined, is superior to other strategies, especially for peptides. In fairness it should be pointed out that, although the strategy of multiple placement followed by bridging produces generally inferior solutions, it is much faster.

The conformations in the peptide library were obtained by generating random conformations of a corresponding dipeptide and extracting the relevant portion of the dipeptide to obtain the fragment conformation. Conformations were rejected on the basis of energy and similarity to other conformations. The conformational library has between a few tens and a few hundred conformations for the standard amino acid fragments. For full details and a discussion of this approach, the reader is referred to Ref. 28.

Later in this paper, some comparison of the scores for various job runs is attempted. For this reason we reproduce here the form of the scoring function used by PRO_LIGAND. The score, S, for a molecule or fragment is calculated as:

$$S = \sum_{1}^{NA} W_A + \sum_{1}^{ND} W_D + \sum_{1}^{NAl} W_{Al} + \sum_{1}^{NAr} W_{Ar} + \sum_{1}^{NRot} W_{Rot} + \sum_{1}^{NAsym} W_{Asym} + \sum_{1}^{NComp-1} W_{Disj} \quad (1)$$

where NA and ND are the numbers of hydrogen bond acceptor and donor sites hit by the structure, NAl and NAr are the numbers of lipophilic aliphatic and lipophilic aromatic interaction sites hit by the structure, NRot and NAsym are the number of rotatable bonds and the number of asymmetric carbon atoms in the structure and NComp is the number of disjoint components in the structure. W_A is the contribution to the score for each hydrogen bond acceptor site hit and the other weights refer to their respective features. The values used for the weights can be amended by the user and the values used in this paper were –0.1 for W_{Rot} and W_{Asym}, +1.0 for W_A and W_D, +0.25 for W_{Al} and W_{Ar} and –2.0 for W_{Disj}. The scoring function is intended to give a very quick general idea about the relative quality of solutions rather than an estimate of the binding affinity. We are currently investigating the incorporation of such an estimate along the lines set out by Böhm [16].

The directed tweak method and its implementation

Flexible fitting has been a research interest in 3D database searching for several years. Traditional approaches centred on representing conformational space by a set of conformers and this is analogous to our strategy for peptides, where multiple conformers for each residue are stored. This type of strategy has been criticised as being too expensive in terms of both search time and storage [39], although it should be noted that recent papers by Smellie et al. indicate that acceptable results can be obtained using judicious conformational sampling [37,38]. The problem of excessive disk space for conformers has been circumvented by the generation of distance keys from multiple conformations and then regenerating these conformations once hits have been detected [54]. Improved methods have centred on investigating the flexibility of the molecule in a way designed to home in on the query in question. Clark et al. [40] have recently compared flexible fitting approaches and have concluded that the directed tweak algorithm [42] is amongst the best available. Therefore we have chosen to adapt this method for application to the de novo design of peptides.

The directed tweak algorithm is based on the random tweak method used for loop modelling in proteins by Levinthal and co-workers [43,44]. Given the protein sequence and position of the end points of the loops, random tweak uses a torsional space minimiser to adapt the loop's conformation to overlap with the end points. It has been adapted by Kilvington and Leach to generate bridges between chemical groups using hydrocarbon chains [22]. The directed tweak algorithm generalises the random tweak method to allow the fitting of an arbitrary number of constraints.

The first part of our flexible fitting algorithm is the generation of an upper and lower distance bounds matrix for the fragment. A good bounds matrix would contain the maximum and minimum distances between every pair of atoms in the fragment as exhibited in an exhaustive collection of low-energy conformers for that fragment. These minimum and maximum distances can then be used to assess whether it is possible for a fragment to achieve a particular geometry, although consistency with the distance bounds matrix is not a sufficient criterion for defining the existence of a match. In our implementation, a peptide fragment is read in from our library and a conformation for that fragment is chosen at random. The predefined rotatable bonds for that fragment are read in as well. The program generates upper and lower distance bounds for the fragment by one of two methods: the first uses distance geometry methodology and the second uses a variant of the tweak algorithm. The two methods used will be described in detail later.

Once a fragment has been chosen and its distance bounds matrix has been calculated, one of the fragment's possible interaction site labellings is chosen at random, and the distance bounds matrix for this labelling is extracted. The subgraph isomorphism algorithm is readily adapted to find hits consistent with the distance bounds matrix rather than an exact distance matrix [53]. Specifically, a hit is designated when there is a one-to-one correspondence between the fragment interaction sites and a set of corresponding design model interaction sites, such that the distance between any pair of the design model interaction sites of the specified type is within the upper and lower bound of the distances between the corresponding pair of fragment interaction sites. These upper and lower bounds are first widened by a user-defined tolerance, which can be different for different types of interaction site. We generally use tolerances in the region of 0.5–0.7 Å for both flexible and rigid fitting. An additional option that has been used on the jobs run for this paper scales the tolerances up to nscale times, gradually increasing the tolerances to the user-defined maximum value; this means that the hits of highest quality are chosen in preference to those of lower quality, which has proved valuable in peptide building runs. The detection of hits using the subgraph isomorphism algorithm is very quick, but it should be remembered that, as with the embedding problem in distance geometry, there is no guarantee that a conformation is attainable by the fragment that is consistent with the fragment distance bounds matrix and

within the tolerance required by the subgraph isomorphism.

Now the program must generate conformations consistent with the position of designated interaction sites in the design model. The directed tweak algorithm is employed for this purpose. We minimise the following cost function, F, with respect to rotations about the rotatable bonds

$$F = \frac{1}{N} \sum_{i \neq j} \frac{2}{T_i + T_j} \left(d_{ij} - d_{ij}^0\right)^2 \quad (2)$$

where d_{ij} and d_{ij}^0 are the distances between the interaction sites i and j in the fragment and design model, respectively, T_i is the fitting tolerance used for interaction site i, and the summation is over all unique pairs of interaction sites. N is a normalisation coefficient given by the square root of the number of unique interaction site pairs. The derivative of this expression with respect to particular rotatable bonds is required and this reduces to a summation of terms of the form:

$$f_{IJKL} = \frac{\partial}{\partial \theta_{KL}} \left(d_{IJ} - d_{IJ}^0\right)^2 \quad (3)$$

where a switch has been made to an atom-based notation, with I and J being atoms containing interaction sites and K and L being the atoms connected by a particular rotatable bond. If I and J are on the same side of the K-L bond, the derivative is zero; if they are on different sides, the derivative is given by:

$$f_{IJKL} = \frac{2\left(d_{IJ} - d_{IJ}^0\right)\left(\mathbf{d}_{KL} \times \mathbf{d}_{IK}\right) \cdot \mathbf{d}_{IJ}}{d_{IJ} d_{KL}} \quad (4)$$

Details on the derivation of this expression are given in Ref. 42. The expression is invalid when d_{IJ} is zero, which only occurs when two interaction sites are on the same atom; this means that the interaction sites are on the same side of the rotatable bond and so are not considered anyway. The sign of the expression depends on the relationship of the atoms in the bonding tree for the fragment and the expression given in Eq. 4 is correct if I is on the K side of the rotatable K-L bond.

The conformation is minimised with respect to the torsions using a steepest descent algorithm. In the PRO_LIGAND program, the cost function has a different form compared to those normally employed in 3D database methods. The number of interaction sites to be matched in our application can vary from 5 to 15, so that the number of terms in the cost function can be greater than 100. In database applications, the cost function will typically have 10 or less terms. The main effect of this difference is that there is a far greater number of local minima in our application than there would be in a normal database search. When a local minimum is located that does not meet the criteria set out below for an acceptable conformation, the minimisation procedure is repeated from different initial geometries until either an acceptable conformation is located or the number of repetitions exceeds the maximum allowed. The maximum number of repetitions is given by:

$$n = A + BN_{rot} \quad (5)$$

where N_{rot} is the number of rotatable bonds and A and B are user-defined parameters with default values of 6 and 4, respectively. As an additional option, a search over all n repetitions can be performed and, if more than one acceptable conformation is located, the one which best fits the geometric constraints of the designated design model sites is chosen. This breadth-first search over tweak starting geometries is a departure from our usual depth-first strategy, but gives slightly superior results in some applications.

We find that convergence is reached in about 5–20 steps of steepest descent minimisation and we abandon the optimisation if the number of steps exceeds a default value of 30. Convergence is reached if the maximum angle change is less than a default of 0.5° or the norm of the gradient is less than a default value of 0.005. When the optimisation process ends, the hit is accepted if the following quantity is less than zero for all unique interaction site pairs:

$$\Delta_{ij} = \|(d_{ij} - d_{ij}^0)\| - 1/2\,(T_i + T_j) \quad (6)$$

This corresponds to the new fragment geometry being an acceptable hit to a rigid match implementation of the subgraph isomorphism algorithm.

The geometry will also not be accepted if it is inconsistent with the original distance bounds matrix. This ensures that geometries containing internal clashes are not used. A more theoretically satisfying approach is to include van der Waals clashes in the cost function. We do this by optionally including the following term:

$$F' = \frac{c}{N'} \sum_{I \neq J} \frac{1}{d_{IJ}^2} - \frac{1}{\left(d_{IJ}^v\right)^2} \quad (7)$$

where d_{IJ} is the distance between nonbonded atoms I and J and d_{IJ}^v is the sum of the appropriate van der Waals radii for the two atoms. Negative contributions are not considered and N' is a normalisation constant equal to the square root of the number of positive contributions to the term. A user-defined constant gives a weight to the term and the default value is 0.25. The term can be used from the beginning of the tweak optimisation or can be brought in after a certain number of iterations. Our experience with the bump-checking term is similar to

TABLE 1
COMPARISON OF SELECTED DISTANCE BOUNDS (Å) FOR THE TRYPTOPHAN FRAGMENT USING DIFFERENT METHODS

Atom pair[a]	Distance geometry	Distance geometry 2	Tweak	Systematic search increment		
				30°	15°	5°
1-2	3.70–8.56	3.70–8.55	3.70–8.53	4.75–8.52	4.60–8.53	4.42–8.53
1-3	3.60–7.40	3.60–7.40	4.18–7.39	4.85–7.38	4.79–7.38	4.73–7.29
1-4	3.65–6.10	3.65–6.10	5.26–6.08	5.38–6.07	5.38–6.08	5.37–6.08
1-5	3.65–5.05	5.01–5.01	5.01–5.01	5.01–5.01	5.01–5.01	5.01–5.01

[a] The atom numbering is as in Fig. 1.

Hurst's [42], i.e., we find that it increases the tendency of the algorithm to fall into local minima and therefore slows down the program. For this reason we tend to favour bump-checking as a post-processing step.

The tweak algorithm is extremely fast but, since it is necessary to repeat the procedure many times during structure generation, it becomes the most expensive part of our de novo design program. Profiling indicates that most time is spent on forming rotation matrices to allow the conformation to be changed and it is difficult to see how this procedure could be further improved. We have attempted to reduce the number of iterations needed for optimisation as well as the number of false hits detected by the program. The latter difficulty is caused by having poor upper and lower bounds for the fragments. This can cause unattainable design model interaction sites to become registered as hits. Initially, we employed a fairly naive implementation of distance geometry methods to obtain the bounds. This approach worked as follows: on picking the fragment, all 1-2 and 1-3 distances were immediately translated into upper and lower bounds plus or minus a very small value, which in our case was 0.002 Å. Any 1-4 distances which did not span a rotatable bond were also translated immediately into an upper and lower bound plus or minus 0.002 Å. The remaining 1-4 bounds were derived directly from the corresponding 1-2 and 1-3 bounds using the equations supplied by Wenger and Smith [55]. All remaining upper bounds were set to a very large default and the lower bounds were set to a default of the sum of the van der Waals radii for the atoms concerned. The bounded distance matrix was then tightened using the standard triangle smoothing algorithms of distance geometry (see for example Ref. 56).

One problem with this approach is the fact that it takes no account of 1-5, 1-6 etc. relationships that do not cross rotatable bonds. Consequently, we have adapted the method so that all fixed distances are included in the original bounded distance matrix before the triangle smoothing step. However, it is known that triangle smoothing does not yield tight distance bounds and for this reason, other methods of distance bounds generation are used in database searching applications [41]. The best method for obtaining the tightest bounds would be to use a systematic search with a small torsion angle increment and to store the distance bounds for the fragment in the library files. However, whilst this would be practical for peptides, it would inhibit the ease of addition of other flexible fragments to the fragment libraries. In database applications, Balducci and Pearlman use the MAXMIN algorithm to obtain the bounds (R. Balducci and R.S. Pearlman, MAXMIN algorithm, unpublished results) and a recent paper has used a neural network for the location of upper and lower bounds, although results in this case were mixed [57]. An alternative approach would be to adapt our implementation of the directed tweak algorithm to calculate distance bounds. This can be done very simply by performing a tweak optimisation for each unique pair of atoms.

For lower bounds, the geometry is tweaked so as to minimise the distance, d_{IJ}, between atoms I and J. The optimisation is terminated when the distance becomes less than the sum of the van der Waals radii for the two atoms. Upper bounds are calculated by minimising $1/d_{IJ}$, using the tweak algorithm. Since the cost function contains only one term, very few local minima are located and it has generally not been found necessary to repeat the tweak optimisation with different starting geometries. Of course, a systematic search will be superior to our implementation of the tweak algorithm since, in the latter, cooperative effects are not taken into account (e.g., it may only be possible to produce a conformation in which

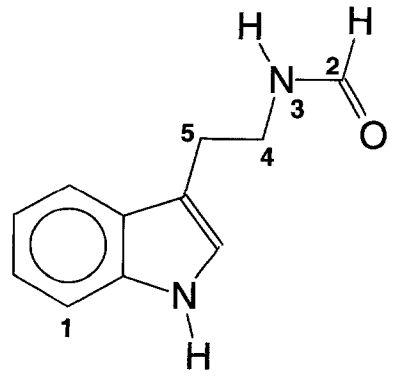

Fig. 1. Tryptophan fragment with the atoms referred to in Table 1 indicated.

TABLE 2
COMPARISON OF SELECTED DISTANCE BOUNDS (Å) FOR THE ASPARAGINE FRAGMENT USING DIFFERENT METHODS

Atom pair[a]	Distance geometry	Distance geometry 2	Tweak	Systematic search increment		
				30°	15°	5°
1-2	2.00–8.18	2.00–8.18	2.00–8.12	3.71–8.12	3.69–8.12	3.62–8.12
1-3	2.85–6.97	2.85–6.97	2.85–6.94	3.36–6.93	3.24–6.93	3.21–6.93
1-4	2.75–5.79	2.75–5.78	2.75–5.75	3.36–5.74	3.36–5.74	3.32–5.74
1-5	2.80–4.52	2.80–4.52	3.57–4.51	3.77–4.50	3.77–4.51	3.77–4.51

[a] The atom numbering is as in Fig. 2.

two atoms are close together by inducing a clash between two other atoms). We take these cooperative effects into account partially by subsequent use of a triangle smoothing algorithm, although this offers little improvement over the tweak bounds themselves.

Table 1 gives some distance bounds obtained using different methods for the trytophan fragment shown in Fig. 1. Table 2 and Fig. 2 contain similar information on the asparagine fragment. The systematic searches were performed at 5°, 15° and 30° increments using the Search_Compare module of the InsightII program [58]. It can be seen that both distance geometry-based approaches give very poor lower bounds in some instances (e.g., atom pair 1-4 in Table 1 and atom pair 1-5 in Table 2). In contrast, whilst the bounds from the tweak method are not as tight as those generated by the systematic search, they are certainly more satisfactory than those produced by the distance geometry methods. Directed tweak has the advantage of being much faster and more convenient than systematic search, which would not be practical for fragments containing even moderate numbers of rotatable bonds. Table 3 contains details on the speed of the bounds calculation and how it scales with the number of rotatable bonds. Currently, the bounds are recalculated every time a fragment is read in, which means that the calculation of the bounds can be an expensive part of the flexible fitting. If this becomes a problem, the program can be adapted so that the bounds are only calculated once for each fragment.

As a final point, glycine and proline are treated as special cases in which the conformational library is used instead of directed tweak. This is necessary in the case of proline because, although the five-membered ring has flexibility, it does not contain any rotatable bonds according to our definition. As pointed out by Hurst [42], flexible ring systems can be treated by the directed tweak method by deleting an appropriate bond in the ring and imposing a distance constraint in the tweak cost function. However, we have chosen not to implement this method in our program.

Force field relaxation

This section details our force field approach to the relaxation of placed fragments. This approach is to carry out the difficult part of the problem, the initial placement of fragments in a favourable position in the active site, using a rule-based method. This is followed by a local force field-based relaxation of the geometry that is subject to constraints. Our method is therefore still rule-based and this does not represent a significant departure in strategy. It should be stressed that our aim is to make small changes to the geometry to improve contacts with the interaction sites and to detect distorted and unrealistic geometries. The accuracy of our force field and its implementation is therefore only of relevance insofar as these goals are addressed. In addition, we make no explicit attempt to consider the receptor or to use force field energies as a basis for deciding between good and bad fragment placements. Again, this reflects the goal of our approach, which is to decide on the quality of a particular fragment placement by how well the relaxed fragments hit the specified interaction sites.

The fragments are read in with AMBER atom types already assigned. When a fragment is positioned and has passed the tests outlined above, a force field relaxation is invoked provided that the partially built structure is not disjoint. AMBER parameters are assigned to the fragment. Figure 3 gives a typical partially built structure and, since our definition of the peptide residue fragments retains the amide bond, there is an aldehyde group at one of the terminal ends of the structure. In order to prevent this leading to some unhelpful distortions, the relevant CH bond is stretched to 1.53 Å and the hydrogen atom type is changed so that it corresponds to a united atom methyl.

In addition, atoms which hit interaction sites are tethered to those sites by a simple harmonic term ar^2, where r is the distance between the interaction site and a is the force constant. The choice of force constant is

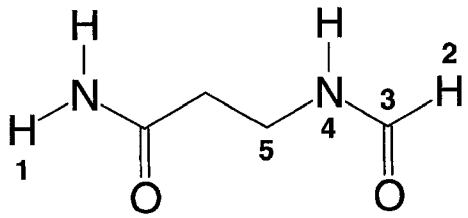

Fig. 2. Asparagine fragment with the atoms referred to in Table 2 indicated.

critical. Too large a value will distort the geometry in an attempt to hit the interaction sites more closely, while too small a value will converge the peptide to its nearest local minimum and cause a loss of contact with the interaction sites. If the force constant is set to be 1 kcal mol^{-1} $Å^{-2}$, a movement of 1.0 Å away from the interaction sites will introduce an energy penalty of 1 kcal mol^{-1}. We therefore use values of between 1 and 5 for the force constant, reflecting the approximate strength of a hydrogen bond. The value of the force constant is scaled slightly, so that interaction sites which have tighter fitting tolerances have proportionately higher force constants.

A general conjugate gradient minimiser is used to optimise the energy function and this gives the user control over the number of iterations (usual value = 25) and the tightness of convergence (usual value = 0.1). Convergence of the function is not a requirement for acceptance of the fragment, but once the optimisation has stopped, a number of simple checks are performed. The fragment is rejected if it does not meet the condition outlined in Eq. 6, i.e., the fragment is no longer in contact with the interaction sites. In addition, for the examples given in this paper, the fragment geometry before and after relaxation is rigidly superimposed onto the interaction sites and the rmsd (root-mean-squared deviation) of the two superimpositions is compared. If the rmsd of fitting increases by more than a factor of 3 as a result of the relaxation, then it is assumed that the original geometry was highly strained and the fragment is also rejected. We do not believe this rejection criterion is matched very often and intend to remove it from future implementations of the program.

TABLE 3
SPEED OF CALCULATION OF DISTANCE BOUNDS FOR AMINO ACID FRAGMENTS[a]

Residue	CPU time (s)	No. of rotatable bonds	No. of atoms
Ala	0.04	2	12
Arg	0.96	5	26
Asn	0.14	3	16
Asp	0.08	3	14
Cys	0.06	3	13
Gln	0.28	4	19
Glu	0.19	4	17
His	0.23	3	19
Ile	0.40	5	21
Leu	0.40	5	21
Lys	0.86	6	24
Met	0.38	5	19
Phe	0.29	3	22
Ser	0.06	3	13
Stat	1.36	8	28
Thr	0.15	4	16
Trp	0.46	3	26
Tyr	0.35	4	23
Val	0.21	3	18

[a] CPU times were measured on an R4000 Silicon Graphics Indy and are averages of 100 invocations of the distance bounds code. Stat refers to the unnatural amino acid statine.

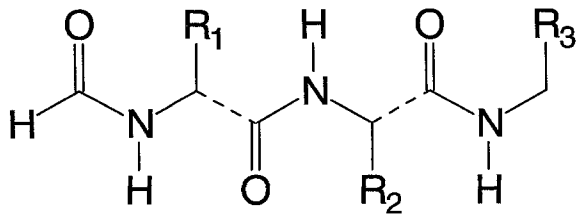

Fig. 3. A typical partially built peptide structure in PRO_LIGAND.

Results

The purpose of this section is to compare the performance of the different methods presented here for the construction of peptides. A question we would like to address is which methods are better; this mirrors a more general and difficult question we would like to answer concerning the relative performance of different de novo design packages. The problem is that it is very difficult to reach conclusions based on a comparison of a few well-chosen structures from runs of the various programs. For this reason, we have tried to make a comparison of the overall results from one particular run, rather than concentrating on the best structures. In our jobs, all structures built by the program are considered as solutions; structure building ceases when a structure cannot be extended anymore. Although the program is capable of only considering structures that fulfil a minimum score or a minimum number of residues, we have not used these options in compiling our data since they might confuse the judgement of the overall performance of the program. We hope that some of the data we present will help other workers interested in doing de novo design with flexible fragments to reach a judgement on some aspects of PRO_LIGAND's performance. Of course, the best solutions produced by the program and the way these solutions interact with the active site are extremely relevant issues, but the highest quality solutions produced by the conformation library for several examples have been presented previously [28] and these are a reasonable indication of the type of solutions that the program can be expected to produce.

The three test cases we have studied here are the rebuilding of a simple helix extracted from dihydrofolate reductase, the building of peptide-like structures complementary to the active site of HIV-1 protease and similar design to an epitope region of lysozyme.

Similar design to a helix

This example was chosen to illustrate the ability of the various methods to reconstruct a typical protein structure segment. Whilst the rebuilding of a helix might appear a very simple test case, it is actually a fairly discriminating

TABLE 4
DESCRIPTION OF DIFFERENCES BETWEEN THE VARIOUS JOBS REFERRED TO IN TABLES 5 AND 6[a]

Job name	Description
C	Conformation library
T1	Flexible fitting using the distance geometry-based distance bounds and a maximum of 10 normal tweak iterations, followed by up to five iterations with the nonbonded term
T2	As for T1, but using up to 15 normal tweak iterations only
TB2	As for T2, but performing a breadth-first search over all tweak starting geometries
G1	As for T2, but performing conjugate gradient relaxation of geometries with a tethering force constant of 3 kcal mol^{-1} $Å^{-2}$
G2	As for G1, but with a tethering force constant of 1 kcal mol^{-1} $Å^{-2}$
NT2	As for T2, but using tweak-derived distance bounds
NTB2	As for TB2, but using tweak-derived distance bounds
NG2	As for G2, but using tweak-derived distance bounds

[a] More details on some of the options relevant to all the jobs are given in the text.

test of the conformational library and of our method in general. The design model was constructed from a helical sequence in DHFR with the hydrogen bond donor, hydrogen bond acceptor and lipophilic sites being produced at the positions of appropriate atoms in the helical structure. Specifically, the sequence was Gly-Arg-Val-Tyr-Glu-Gln-Phe-Leu-Pro-Lys-Ala, which corresponds to residues 97–107 in Brookhaven file 4DFR [59].

Different jobs were run on this similar design problem using our most extensively labelled library. The library was ranked in such a way that glycine, valine and alanine were considered only after all other residues were exhausted. This is because these residues are 'promiscuous' and fit almost anywhere along the sequence, leading to a diversity which is of little interest. The ranking procedure is facilitated by pre-organisation of the libraries into different subdirectories which the user ranks by specification of different commands in the input specification [28]. As a result of the ranking of the fragments, solutions which contain large numbers of promiscuous fragments are relatively poor solutions since their inclusion implies that no other more interesting fragments could be inserted. The tolerances used for fitting the interaction sites were set at 0.5 Å and all jobs used five scalings (i.e., nscale = 5) of these tolerances during the fitting. Data on nine jobs is given in Table 5 and a description of the differences between the various jobs is given in Table 4. Other parameters referred to in the Methods section and not given in Table 4 were set at their default values.

A number of points are worth noting from Table 5. The most striking observation is that methods which do not employ a force field relaxation cannot grow peptides longer than seven residues. The reason for this is that the imperfections in the fitting to the interaction sites cumulate as each new fragment is added. Unless this gradual drift from the interaction sites is corrected by the force field relaxation, eventually it becomes impossible to join a new fragment to the evolving structure. The stronger the tethering force applied, the closer the peptide will match the helix structure. However, one should be careful about hailing this as a good result, since it is more or less what one would expect with large tethering forces. Nonetheless, it is encouraging that comparatively small tethering forces are required to keep the evolving peptide from

TABLE 5
RESULTS FOR BUILDING 1000 PEPTIDES ONTO A SIMILAR DESIGN MODEL FOR HELICAL SEQUENCE Ala-Lys-Ile-Leu-Phe-Gln-Glu-Tyr-Val-Arg-Gly USING DIFFERENT METHODS

Property of solutions	C	T1	T2	TB2	G1	G2	NT2	NTB2	NG2
Highest score	18.7	20.0	21.5	21.3	32.1	26.1	20.9	21.7	30.5
10th highest score	13.9	16.1	18.8	20.1	29.6	21.1	18.7	19.8	21.6
Average score	4.5	4.4	4.4	5.8	7.7	4.5	4.4	6.2	4.5
No. of 6-residue solutions	4	7	12	16	–	–	8	18	–
No. of 7-residue solutions	2	2	7	19	–	–	8	11	–
No. of 8-residue solutions					–	–			–
No. of 9-residue solutions					25	2			3
No. of 10-residue solutions					31	1			1
No. of 11-residue solutions					7				1
Gly percentage	29	18	19	18	16	18	21	19	19
Time per structure (s)	235	5	5	6	18	16	10	13	21

A solution is defined as any structure that could not be extended under the conditions of the particular job.

drifting too far from the hit interaction sites. It can also be seen that tweak methods perform considerably better than the conformational library, consistently outscoring the conformer approach and producing more six- and seven-residue solutions with lower percentages of promiscuous fragments. In addition, tweak methods which do not use a non-bonded repulsion term for intrafragment clashes perform better (compare T1 with T2). This is in agreement with Hurst's original observations, the explanation being that there is a greater tendency for the tweak method to become trapped in suboptimal local minima when the cost function contains a large number of terms. It can also be seen that the breadth-first search to locate the best fit amongst all random starting geometries offers some advantage (compare T2 with TB2) over simply accepting the first acceptable geometry. There is some evidence that the tweak methods employing the tighter bounds are superior, but the difference is minimal. The conclusion here is that the detection of false hits which meet the bounds but which cannot be realised by the fragment is not a major problem in this similar design example.

The timings for the jobs are for one processor of a Convex Exemplar machine, which is about three times faster than an R4000 Silicon Graphics machine for these applications. The timings for the conformation library are very slow, but can be reduced to 18 s per structure through the use of a prescreening option which screens out conformations that can never hit the interaction sites. This prescreening option is particularly useful in this example, because only a very small number of the prestored conformations will be able to meet the tight constraints of the similar design interaction sites. However, the option is less useful in more realistic problems and makes virtually no difference to the timing in the HIV-1 protease and lysozyme examples which follow.

Complementary design to HIV-1 protease

HIV protease has been a popular target for structure-based drug design (see for example Refs. 60–62) owing to the availability of accurate crystal structures and the belief that inhibition of the protease will attenuate the spread of viral infection. The protease has also been widely used as a test case for de novo design programs.

In our previous publication [28] we grew peptides complementary to the protease active site, giving encouraging results. The protease was therefore a natural choice for testing the new methodologies described in this paper. The design model was constructed in the same manner as before [25,28]. In brief, 491 atoms, including one water molecule, were extracted from the active site region of a cleaned-up structure of HIV-1 protease complexed with the inhibitor acetyl pepstatin (PDB entry 5HVP) [63]. This design base was used to create a design model containing 855 hydrogen bond donor, hydrogen bond acceptor and lipophilic sites complementary to the active site. In general, atoms with suitable chemistry and orientation within the active site region lead to clusters of appropriate interaction sites. The design model produced is therefore much more difficult than the simple and restrictive similar design model used in the helix example. For each fragment, many possible orientations and conformations will constitute acceptable hits for regions of the design model and this leads to a much larger search space of solutions to be considered.

Peptides were grown emanating from a seed molecule in the catalytic region of the enzyme. The seed was an

TABLE 6
RESULTS FOR BUILDING 200 PEPTIDES ONTO A COMPLEMENTARY DESIGN MODEL FOR HIV PROTEASE WITH AN ETHANOL SEED

Property of solutions	C	T1	T2	TB2	G2	NT2	NTB2	NG2
Highest score	12.0	11.0	12.0	11.5	11.0	11.5	11.5	11.0
10th highest score	10.5	10.0	10.0	10.0	10.0	10.0	10.5	10.0
Average score	7.5	7.5	8.0	7.5	7.0	8.0	7.5	6.5
No. of 6-residue solutions	40	49	59	63	44	53	51	31
No. of 7-residue solutions	3	2	0	1	1	0	1	0
Ala percentage	27	28	31	29	32	28	30	33
Gly percentage	23	31	29	32	31	31	30	32
Ile percentage	11	6	4	4	8	5	5	6
Leu percentage	14	13	10	10	8	7	9	8
Met percentage	29	30	32	31	30	31	32	32
Phe percentage	6	4	5	4	4	6	5	5
Pro percentage	1	2	4	5	4	4	3	3
Stat[a] percentage	0	0	0.3	0.3	0	0	0	0
Trp percentage	4	1	0	0	0	2	2	1
Val percentage	6	4	4	5	3	4	5	1
Time per structure (s)	641	115	111	110	151	120	117	151

[a] Stat refers to the unnatural amino acid statine.

ethanol molecule extracted from the crystallographic coordinates of acetyl pepstatin; more details of this procedure have been given by Frenkel et al. [28]. This seeding strategy was adopted because peptides do not constitute good designs for this problem, being substrates of the enzyme rather than inhibitors. A sequential growing strategy was adopted and the moderately and extensively labelled libraries were used. The libraries were ranked so that extensively labelled fragments and fragments with larger side chains were searched through first. Tolerances of 0.7 Å were used and, in this instance, the tolerances were scaled three times during the subgraph isomorphism search in order to locate the tightest hits first. The maximum number of hits tried for each fragment was increased over the helix example since the design model contains a greater density of sites, leading to more possibilities. Results are given in Table 6 and the key to the different runs is given in Table 4.

The results for the HIV-1 protease case are less clear-cut than the helix example. Here it is especially important not to take the raw scores too seriously, since the differences between the various runs are comparatively minor. From our experience with this example, the statistical fluctuation in the highest score statistics from runs with identical parameters (but different random number seeds) would be expected to be about 1.0. As a result, the average score is probably the most reliable of the score statistics. In addition, the problem of diversity needs to be addressed, since a relatively poor diversity in the highest scoring solutions would indicate a poorer method. It appears that tweak methods are of similar quality to the conformational library and in some cases could even be better. However, a certain amount of caution should be exercised here, since the geometries may not be as good for the tweak methods as the conformational approach. The conclusion we draw is that the tweak methods are giving results that are at least as good as with the conformational library approach, but that are obtained about five times quicker.

Similar design to epitope regions of lysozyme

The identification of epitopes is a key aspect in the design of synthetic vaccines, but there is no guarantee that isolated epitopic sequences will elicit an immune response when synthesised. This is because antibodies recognise the 3D arrangement of key features, not the sequence itself. It would be extremely useful if mimics of epitopes could be constructed given the 3D coordinates of a known or predicted epitope region. We have previously grown peptides that match a small discontinuous region in lysozyme, growing our peptides to hit a sparse set of key features [28]. This gives a design problem with very different characteristics to those of the HIV-1 protease problem, since an extremely large number of solutions and orientations are available for peptides which hit subsets of these sites. It is therefore of interest to check the performance of the flexible fitting methodology outlined in this paper on this difficult problem.

Full details of the design approach are given in Ref. 28, but we give a few details here to aid the reader. We chose to look at residues 117–125 of the crystal structure of lysozyme complexed with the antibody D1.3 (PDB entry 1FDL) [64], marking as hydrogen bond interaction sites the carbonyl backbone group on Gly^{117}, the carboxylate group on Asp^{119}, the N-H backbone group on Val^{120}, the amide group on the side chain of Gln^{121}, and the N-H bonds on the side chain of Arg^{125}. Some of the atoms not in contact with the antibody were specified as lipophilic sites to encourage growth between the desired features. Since the arginine is four residues away from the glutamine, the backbone atoms joining these residues were not included in the design base. Peptides were grown to fit these features using the moderately and sparsely labelled libraries. The libraries were labelled so that polar residues were searched in preference to lipophilic residues, although full details of the library ranking are not important here. Maximum tolerances of 0.7 Å were used in this application and three scalings of these tolerances were adopted. Results are given in Table 7; the key to the different runs is again given in Table 4.

There is some danger in over-interpreting the results in this case, since there is considerable stochastic fluctuation in the quality of the results from any one run. In fact, runs of many tens of thousands of structures would probably be necessary to gain good stability in the gathered statistics because the search space for the solutions is so large and difficult to sample. However, some trends can be delineated. Firstly, the introduction of breadth (compare T2 with TB2 and NT2 with NTB2) is affecting the quality of the best structures. This is because by introducing breadth we have adversely restricted the diversity of the solutions, and this might also explain why the conformational library is giving slightly better results than the tweak methods. In this example, the design model is very sparse and the interaction sites are non-ideally positioned relative to each other (stemming as they do from a discontinuous design base). Methods which optimise the conformations of the fragment will tend to generate less diverse solutions, since they are unable to accept the intermediate suboptimal fragment fits necessary to form the best overall structures. This effect will disfavour the geometry relaxation approach and to a lesser extent all the tweak methods relative to the conformational library. It might also be possible to infer that the methods employing the superior bounds are giving slightly better results than those which are not. However, the most striking observation is that the best tweak methods are between five and ten times quicker than the conformational library approach, with only a marginal degradation of performance.

TABLE 7
RESULTS FOR BUILDING 1000 PEPTIDES ONTO A SIMILAR DESIGN MODEL FOR A DISCONTINUOUS EPITOPE REGION ON LYSOZYME USING DIFFERENT METHODS

Property of solutions	C	T1	T2	TB2	G1	G2	NT2	NTB2	NG2
Highest score	7.0	7.5	7.5	6.5	6.0	8.0	7.0	6.5	6.0
10th highest score	5.5	5.5	5.5	4.5	5.0	5.0	5.5	5.0	5.5
No. of 3-residue solutions	41	26	32	13	19	15	28	22	29
No. of 4-residue solutions	3	2	4	1	2	1	4	2	0
Gly percentage	4	8	15	15	7	11	10	15	10
Time per structure (s)	87	10	9	10	10	10	17	17	18

Discussion

The results generally indicate that the methodological changes outlined in this paper have merit. Firstly, the tweak methods are considerably quicker than the conformational library, allowing more structures to be obtained and more options to be explored in a particular amount of time. This is clearly an issue with both the lysozyme and the protease example, where many structures are needed to obtain the best results. Secondly, the preparation of fragments is now very straightforward. This opens the way to taking fragments directly from a commercial or proprietary database and using them to perform de novo design with flexible fitting. In this regard, the distance bounds calculation using the tweak method is of considerable importance since it will allow us to calculate distance bounds automatically after the programmatic location of the rotatable bonds. In contrast, the preparation of a compact list of conformations would require more work and more care. The quality of the results appears to be generally comparable with that obtained using the conformational library.

The question arises as to why the flexible fitting is so much faster than the conformational library, since our application is different from a 3D database. The critical factor is the percentage of fragments which are registered as hits by the subgraph isomorphism algorithm and then have to be optimised using the tweak algorithm. The tweak algorithm is time-consuming because, in our implementation, it has to be repeated many times to overcome trapping in local minima. However, the subgraph isomorphism algorithm is also very time-consuming since, in our case, there are hundreds of design model interaction sites that must be considered. If the percentage of fragments that pass this test is low (i.e., a very tightly constrained design model) then the tweak methods are very much faster than the conformational library. This is illustrated by the helix example, although in this case prescreening options can be used to speed up the conformational library approach. If the percentage that passes the graph theory test is higher, the speed-up is less spectacular, although it is still considerable. When we move towards larger libraries derived from external databases, it is to be expected that the graph theory test will eliminate more fragments, making the tweak approach even faster in relative terms. The conformational library approach could be improved by using the distance bounds matrix to reject poor fragments before invoking the conformational library, although this dual approach would require constructing the conformers for each fragment and this would be inconvenient in general applications. Another compromise alternative would be to have a small conformational library and perform a tweak search in the vicinity of a chosen conformation, with each conformation having its own set of tighter distance bounds. Whilst this could be quite efficient, it would tend to make the definition of fragments more difficult and the derivation of the distance bounds would not be straightforward.

Another interesting issue is why the flexible fitting approach does not give better results than the conformational library. We did expect the tweak method to be superior, since this was the prevailing dogma in 3D database applications and many of the different options implemented and tested here represent attempts to improve the performance of the algorithm in line with our expectations. Smellie et al. have recently published papers supporting the opposing view that, for molecular applications with realistic thresholds, good conformational libraries are competitive with flexible fitting methods and even offer some advantages [37,38]. Our work could be seen to support this view – at least as far as the *quality* of the results is concerned. However, we should be careful about attaching too much weight to this argument since our application is very different. If we consider the HIV-1 protease example, each fragment has about 10 sites marked on it and is being fitted onto a design model of 855 sites. Interaction sites of the same type are arranged in clusters and can be within 0.1–0.5 Å of each other. It can be seen from Tables 2 and 3 that even the improved distance bounds are still considerably worse than the systematic search distance bounds. In addition, the systematic search distance bounds are still a considerable simplification of the conformational space available. (For example, it is very unlikely that any fragment can simultaneously meet all the lower bounds in the distance bounds matrix, since cooperative effects will not allow

this.) As a result, it is to be expected that the fail rate for the location of viable tweak hits, given an initial hit from the subgraph isomorphism algorithm, will be very high. In fact, this fail rate will be so high that often the program will never find an acceptable hit before exceeding any reasonable maximum number of subgraph isomorphism hits specified by the user. This will cause the program to miss good fragments on some occasions. This deficiency will tend to be offset by the fact that the entire conformational space is available to the tweak algorithm rather than the restricted sampling offered by the conformational library. Because of these competing factors, it was probably not correct to expect that the tweak method would give a superior performance in our application. There is also a possibility that our application to peptides has favoured the conformational library approach, since the restricted conformational space available to amino acids in peptides may not be typical of general fragments. However, the performance of the algorithm is still satisfactory and we believe the flexible fitting methodology is a considerable advance over the conformational library.

It should also be remembered that there are some general approximations in our interaction site-based approach that may be affecting the quality of the flexible fitting and relaxation results. The main approximation is that specific sites are chosen during the subgraph isomorphism fitting process onto which the fragments are placed, tweaked and/or relaxed. There is no opportunity for a partially built structure or a fragment to change the design model interaction sites which it has matched. This means that the specific positioning of the interaction sites may restrict the solution space open to the program and as a result some optimal solutions could be missed. One solution is just to produce a greater density of sites, but this tends to slow down the program without enhancing the overall quality of solutions. The approximation is particularly worrying for the lipophilic sites, which are treated as specific points in space whereas, physically, they should be diffuse regions of space. A strategy could be formulated to give structures the opportunity to switch sites during building or perhaps to define more general regions of space for lipophilic sites by introducing appropriate changes to the tweak cost function, but we have not pursued this approach as yet.

An additional question that should be addressed is the performance of the relaxation methods adopted in this paper. The initial results with the helix are spectacularly good, but this should be disregarded because it is a 'rigged' test case as far as this kind of approach is concerned. The results for the two real cases (i.e., with non-ideal interaction site positioning) actually appear to be worse than those of the other methods. A sceptical response would be to conclude that the geometries produced by the program are very poor and the relaxation method causes such geometries to be rejected. A more likely explanation is that the conditions and parameters which were optimised for the helical case are not appropriate for the other examples and that the whole approach is too great a simplification for the more difficult examples. In addition, the relaxation method restricts diversity by pulling similar structures into similar conformations and this certainly will affect the quality of the overall results. This is because, as the structure is being built, it may not be advantageous to use the optimal positioning for a particular fragment since this may prevent the formation of a particularly good interaction by another fragment added later on in the building process. We conclude that it is perhaps better in many applications to leave the optimisation of the ligand geometries and calculation of interaction energies as a post-processing step, although it was certainly worthwhile to discover this piece of information.

Of the tweak methods adopted in this paper, it seems clear that the use of a nonbonded clash term in the tweak cost function (at least in this implementation) was not helpful, since it degraded the performance of the program by introducing more local minima into the optimisation problem. The breadth-first search over tweak starting geometries gives better results in some applications and worse ones in others, so the conclusion is that it is sometimes useful. The use of the improved distance bounds does appear to generally improve the results but the improvement is marginal; however, it would clearly be helpful to tighten the bounds further and we are investigating this possibility.

Conclusions

We have found that the directed tweak method appears to offer advantages in speed and convenience over the use of libraries of conformations in de novo design of peptides. The results obtained by the method are of comparable quality to those obtained with a conformational library approach. We have also found little benefit for general applications in our implementation of a force field relaxation methodology. We are now in a position to perform flexible fitting in de novo design using external databases as a source of molecular fragments and we are actively pursuing this line of research.

References

1 Verlinde, C.L.M.J. and Hol, W.G.J., Structure, 2 (1994) 577.
2 Lewis, R.A. and Leach, A.R., J. Comput.-Aided Mol. Design, 8 (1994) 467.
3 Nishibata, Y. and Itai, A., Tetrahedron, 47 (1991) 8985.
4 Nishibata, Y. and Itai, A., J. Med. Chem., 36 (1993) 2921.
5 Rotstein, S.H. and Murcko, M.A., J. Comput.-Aided Mol. Design, 7 (1993) 23.
6 Pearlman, D.A. and Murcko, M.A., J. Comput. Chem., 14 (1993) 1184.

7 Gehlhaar, D.K., Moerder, K.E., Zichi, D., Sherman, C.J., Ogden, R.C. and Freer, S.T., J. Med. Chem., 38 (1995) 466.
8 Moon, J.B. and Howe, W.J., Protein Struct. Funct. Genet., 11 (1991) 314.
9 Moon, J.B. and Howe, W.J., In Wermuth, C.G. (Ed.) Trends in QSAR and Molecular Modelling 92 (Proceedings of the 9th European Symposium on Structure–Activity Relationships: QSAR and Molecular Modelling), ESCOM, Leiden, 1993, pp. 11–19.
10 Miranker, A. and Karplus, M., Protein Struct. Funct. Genet., 11 (1991) 29.
11 Caflisch, A., Miranker, A. and Karplus, M., J. Med. Chem., 36 (1993) 2142.
12 Rotstein, S.H. and Murcko, M.A., J. Med. Chem., 36 (1993) 1700.
13 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 61.
14 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 593.
15 Böhm, H.-J., In Kubinyi, H. (Ed.) 3D QSAR in Drug Design: Theory, Methods and Applications, ESCOM, Leiden, 1993, pp. 386–405.
16 Böhm, H.-J., J. Comput.-Aided Mol. Design, 8 (1994) 243.
17 Böhm, H.-J., J. Comput.-Aided Mol. Design, 8 (1994) 623.
18 Lewis, R.A., Roe, D.C., Huang, C., Ferrin, T.E., Langridge, R. and Kuntz, I.D., J. Mol. Graphics, 10 (1992) 66.
19 Roe, D.C. and Kuntz, I.D., J. Comput.-Aided Mol. Design, 9 (1995) 269.
20 Tschinke, V. and Cohen, N.C., J. Med. Chem., 36 (1993) 3863.
21 Ho, C.W.M. and Marshall, G.R., J. Comput.-Aided Mol. Design, 7 (1993) 623.
22 Leach, A.R. and Kilvington, S.R., J. Comput.-Aided Mol. Design, 8 (1994) 283.
23 Eisen, M.B., Wiley, D.C., Karplus, M. and Hubbard, R.E., Protein Struct. Funct. Genet., 19 (1994) 199.
24 Bohacek, R.S. and McMartin, C., J. Am. Chem. Soc., 116 (1994) 5560.
25 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.
26 Waszkowycz, B., Clark, D.E., Frenkel, D., Li, J., Murray, C.W., Robson, B. and Westhead, D.R., J. Med. Chem., 37 (1994) 3994.
27 Westhead, D.R., Clark, D.E., Frenkel, D., Li, J., Murray, C.W., Robson, B. and Waszkowycz, B., J. Comput.-Aided Mol. Design, 9 (1995) 139.
28 Frenkel, D., Clark, D.E., Li, J., Murray, C.W., Robson, B., Waszkowycz, B. and Westhead, D.R., J. Comput.-Aided Mol. Design, 9 (1995) 213.
29 Gillet, V.J., Johnson, A.P., Mata, P., Sike, S. and Williams, P., J. Comput.-Aided Mol. Design, 7 (1993) 127.
30 Gillet, V.J., Newell, W., Mata, P., Myatt, G., Sike, S., Zsoldos, Z. and Johnson, A.P., J. Chem. Inf. Comput. Sci., 34 (1994) 207.
31 Mata, P., Gillet, V.J., Johnson, A.P., Lampreia, J., Myatt, G.J., Sike, S. and Stebbings, A.L., J. Chem. Inf. Comput. Sci., 35 (1995) 479.
32 Leach, A.R. and Lewis, R.A., J. Comput. Chem., 15 (1994) 233.
33 Cohen, A.A. and Shatzmiller, S.E., J. Mol. Graphics, 11 (1993) 166.
34 Cohen, A.A. and Shatzmiller, S.E., J. Comput. Chem., 15 (1994) 1393.
35 Glen, R.C. and Payne, A.W.R., J. Comput.-Aided Mol. Design, 9 (1995) 181.
36 Clark, D.E. and Murray, C.W., J. Chem. Inf. Comput. Sci., in press.
37 Smellie, A., Kahn, S. and Teig, S.L., J. Chem. Inf. Comput. Sci., 35 (1995) 285.
38 Smellie, A., Kahn, S. and Teig, S.L., J. Chem. Inf. Comput. Sci., 35 (1995) 295.
39 Pearlman, R.S., In Kubinyi, H. (Ed.) 3D QSAR in Drug Design: Theory, Methods and Applications, ESCOM, Leiden, 1993, pp. 41–79.
40 Clark, D.E., Jones, G., Willett, P., Kenny, P.W. and Glen, R.C., J. Chem. Inf. Comput. Sci., 34 (1994) 197.
41 Moock, T.E., Henry, D.R., Ozkabak, A.G. and Alamgir, M., J. Chem. Inf. Comput. Sci., 34 (1994) 184.
42 Hurst, T., J. Chem. Inf. Comput. Sci., 34 (1994) 190.
43 Shenkin, P.S., Yarmush, D.L., Fine, R.M., Wang, H. and Levinthal, C., Biopolymers, 26 (1987) 2053.
44 Fine, R.M., Wang, H., Shenkin, P.S., Yarmush, D.L. and Levinthal, C., Protein Struct. Funct. Genet., 1 (1986) 342.
45 Weiner, S.J., Kollman, P.A., Case, D.A., Singh, U.C., Ghio, C., Alagona, G., Profeta Jr., S. and Weiner, P., J. Am. Chem. Soc., 106 (1984) 765.
46 Siani, M.A., Weininger, D. and Blaney, J.M., J. Chem. Inf. Comput. Sci., 34 (1994) 588.
47 Gallop, M.A., Barrett, R.W., Dower, W.J., Fodor, S.P.A. and Gordon, E.M., J. Med. Chem., 37 (1994) 1233.
48 Gordon, E.M., Barrett, R.W., Dower, W.J., Fodor, S.P.A. and Gallop, M.A., J. Med. Chem., 37 (1994) 1385.
49 Klebe, G., J. Mol. Biol., 237 (1994) 212.
50 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.W., Rodgers, J.R. and Watson, D.G., Acta Crystallogr., B35 (1979) 2331.
51 Ullmann, J.R., J. Assoc. Comput. Machinery, 23 (1976) 31.
52 Brint, A.T. and Willett, P., J. Mol. Graphics, 5 (1987) 49.
53 Clark, D.E., Willett, P. and Kenny, P.W., J. Mol. Graphics, 10 (1992) 194.
54 Murrall, N.W. and Davies, E.K., J. Chem. Inf. Comput. Sci., 30 (1990) 312.
55 Wenger, J.C. and Smith, D.H., J. Chem. Inf. Comput. Sci., 22 (1982) 29.
56 Dress, A.W.M. and Havel, T.F., Discrete Applied Math., 19 (1988) 129.
57 Jordan, S., Leach, A.R. and Bradshaw, J., J. Chem. Inf. Comput. Sci., 35 (1995) 640.
58 Insight II, v. 2.3.0, Biosym Technologies, Inc., San Diego, CA.
59 Bolin, J.T., Filman, D.J., Matthews, D.A., Hamlin, R.C. and Kraut, J., J. Biol. Chem., 257 (1982) 13650.
60 Appelt, K., Perspect. Drug Discov. Design, 1 (1993) 23.
61 Fitzgerald, P.M.D., Curr. Opin. Struct. Biol., 3 (1993) 868.
62 Redshaw, S., Exp. Opin. Invest. Drugs, 3 (1994) 273.
63 Fitzgerald, P.M.D., McKeever, B.M., Van Middlesworth, J.F., Springer, J.P., Heimbach, J.C., Leu, C.-T., Herber, W.K., Dixon, R.A.F. and Darke, P.L., J. Biol. Chem., 265 (1990) 14209.
64 Fischmann, T.O., Bentley, G.A., Bhat, T.N., Boulot, G., Mariuzza, R.A., Phillips, S.E.V., Tello, D. and Poljak, R.J., J. Biol. Chem., 266 (1991) 12915.