Journal of Computer-Aided Molecular Design, 11 (1997) 369–384.

J-CAMD 398

Multiple automatic base selection: Protein–ligand docking based on incremental construction without manual intervention

Matthias Rarey*, Bernd Kramer and Thomas Lengauer

German National Research Center for Information Technology (GMD), Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven, D-53754 Sankt Augustin, Germany

Received 31 January 1997
Accepted 13 March 1997

Keywords: Molecular docking; Flexible docking; Protein–ligand interaction; Molecular flexibility; Conformational analysis; Drug design

Summary

A possible way of tackling the molecular docking problem arising in computer-aided drug design is the use of the incremental construction method. This method consists of three steps: the selection of a part of a molecule, a so-called *base fragment*, the placement of the base fragment into the active site of a protein, and the subsequent reconstruction of the complete drug molecule. Assuming that a part of a drug molecule is known, which is specific enough to be a *good* base fragment, the method is proven to be successful for a large set of docking examples. In addition, it leads to the fastest algorithms for flexible docking published so far. In most real-world applications of docking, large sets of ligands have to be tested for affinity to a given protein. Thus, manual selection of a base fragment is not practical. On the other hand, the selection of a base fragment is critical in that only few selections lead to a low-energy structure. We overcome this limitation by selecting a representative set of base fragments instead of a single one. In this paper, we present a set of rules and algorithms to automate this selection. In addition, we extend the incremental construction method to deal with multiple fragmentations of the drug molecule. Our results show that with multiple automated base selection, the quality of the docking predictions is almost as good as with one manually preselected base fragment. In addition, the set of solutions is more diverse and alternative binding modes with low scores are found. Although the run time of the overall algorithm increases, the method remains fast enough to search through large ligand data sets.

Introduction

A central problem in the area of computer-aided drug design is the prediction of complexes between proteins and potential drug molecules, the so-called docking problem. The binding affinity as well as the three-dimensional structure of a protein–ligand complex are important factors during the design process of new drugs. The three-dimensional structures of target proteins of therapeutical interest are becoming more and more available, making the application of computer-based methods for the search of new drugs possible.

Since Kuntz et al. [1] have presented the DOCK approach, a wide area of different methods have been applied to the docking problem (see Refs. 2–6 for overviews). Recent approaches consider the conformational flexibility of the ligand molecule explicitly and use physicochemical information on molecular interactions in order to place the ligand into the active site.

Recently, evolutionary programming (mostly genetic algorithms) has become popular as an underlying optimization scheme for tackling the docking problem [7–10]. These approaches produce docking predictions of high accuracy. Their major disadvantage is the long total run

*To whom correspondence should be addressed.

Three-dimensional models of all the examples described in this paper as well as the results for additional test cases taken from the PDB are available as VRML files from our WWW pages at http://www.gmd.de/SCAI/alg/reliwe/reliwe_home.html. This site also contains a link to our free web interface to the FLEXX docking software. Licenses for the FLEXX software package are available for SUN, SGI, and PCs running the Linux operation system. Interested readers should contact the corresponding author.

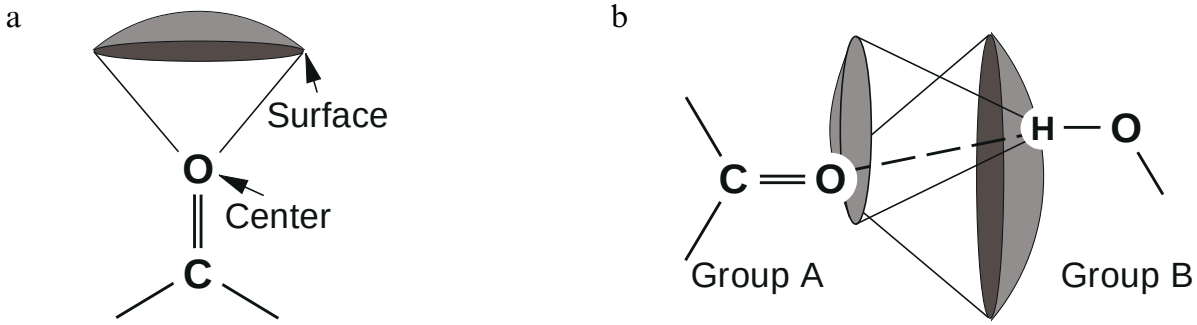

Fig. 1. Interaction model. (a) Example for an interacting group. The interaction geometry is described by an interaction center and an interaction surface. (b) The geometric condition for a match between two compatible interacting groups: the interaction center of the first group lies on the interaction surface of the second and vice versa.

time resulting from the fact that the optimization must be repeated several times in order to obtain a reliable prediction.

An alternative way of solving optimization problems in computer science is to (recursively) divide a problem into smaller subproblems, solve these first, and then construct a solution for the whole problem based on the solutions of the smaller subproblems. The *incremental construction method* is a possible application of this strategy to the docking problem. First, the ligand is cut into small pieces, one of the pieces is placed independently from the rest of the ligand into the active site, and then the remaining part of the ligand is added to the placements of the partial ligand found so far. Because the docking problem is not really decomposable in this sense (since the subproblems are not independent from each other), docking algorithms based on the incremental construction method are heuristic. But even if the global minimum of the applied scoring function is not always found, docking tools based on this method, like the modified DOCK approach of Leach and Kuntz [11], the docking tool FLEXX [12,13], and the recently published tool Hammerhead [14], have proven to be successful in many cases.

According to the description of the incremental construction method, the algorithm of our docking tool FLEXX consists of three phases: *base selection*, *base placement*, and *complex construction*. The topic of this paper is the automation of the base selection phase. The base placement [15] and complex construction algorithm as well as the physicochemical model behind FLEXX have been published previously [13].

So far, only Hammerhead selects base fragments automatically. More precisely, the ligand is cut into a small number of building blocks and the selection of a base fragment is simply avoided by using each of these blocks as a base fragment (or head in their nomenclature).

Before we start describing our method for automatic base selection, we will give a short summary of the model behind FLEXX and the succeeding base placement and complex construction phase. A knowledge of these components of FLEXX is necessary in order to evaluate the eligibility of a part of the ligand as a base fragment. A more detailed description can be found in Refs. 13 and 15.

The physicochemical model

The physicochemical model behind FLEXX can be divided roughly into three parts: the conformational space of the ligand, the model of protein–ligand interactions, and the scoring function. The conformational space of the ligand is approximated by a discrete set of conformations. For each ring system, a set of low-energy conformations is computed with the ring conformation program SCA [16]. SCA is able to handle ring systems composed of up to seven-membered elementary rings, which is sufficient for most applications in drug design.

To each acyclic single bond, a set of low-energy torsion angles is assigned using the MIMUMBA torsion angle database [17]. Generated conformations are only tested for intramolecular clashes, and there is no conformational energy term in the scoring function. Within this model, a drug molecule of typical size, say one flexible ring system and 10 rotatable bonds, has on the order of a few million conformations and the rms distance from the most similar conformation in this discrete set compared to the conformation of a drug molecule in a protein–ligand complex can be up to roughly 0.5 Å.

A subset of protein–ligand interactions is used for placing the ligand or base fragment into the active site. These interactions are modeled as follows: an interacting group of a ligand or a protein is assigned an *interaction type* and an *interaction geometry* consisting of an *interaction center* and an *interaction surface* (see Fig. 1a). Possible interaction surfaces are always parts of spherical surfaces, either a full sphere, a (section of a) spherical cap, or a spherical rectangle. An interaction between an interaction group A of the ligand and an interaction group B of the protein occurs if:

(1) the interaction types of A and B are compatible; and (2) the interaction center of A lies approximately on the interaction surface of B and vice versa (see Fig. 1b).

Possible interaction types are hydrogen bond donors and acceptors, metal and metal acceptors, aromatic ring

atoms, methyl groups, amide groups, and aromatic ring centers. Interaction types are divided into *geometrically restrictive* (or *tight*) and *less restrictive* (or *loose*) types. Hydrogen bonds and salt bridges are classified as tight types while hydrophobic interactions are considered loose.

The scoring function of FLEXX is the function developed by Böhm for the de novo design program LUDI [18,19] with some minor changes.

$$\Delta G = \Delta G_0 + \Delta G_{rot} \times N_{rot} \quad (1)$$

$$+ \Delta G_{hb} \sum_{\text{neutral H–bonds}} f(\Delta R, \Delta\alpha) \quad (2)$$

$$+ \Delta G_{io} \sum_{\text{ionic int.}} f(\Delta R, \Delta\alpha) \quad (3)$$

$$+ \Delta G_{aro} \sum_{\text{aro. int.}} f(\Delta R, \Delta\alpha) \quad (4)$$

$$+ \Delta G_{lipo} \sum_{\text{lipo.cont.}} f^*(\Delta R) \quad (5)$$

The function can be divided into three parts. The first part (1) consists of a fixed term ΔG_0 and a term $\Delta G_{rot} \times N_{rot}$ taking the loss of entropy during ligand binding into account. The second part (2–4) contains the contributions for matched interaction groups like hydrogen bonds (2), salt bridges and charged hydrogen bonds (3), and aromatic interactions (4). Each of these terms consists of a fixed contribution per interaction (ΔG_{hb}, for example) multiplied by a penalty function $f(\Delta R, \Delta\alpha)$. The penalty functions are piecewise linear functions scaling the contribution of an interaction with respect to its geometry. The third part (5) rates the atom–atom contacts between protein and ligand like hydrophobic contacts and forbiddingly close contacts (clashes). The second and third parts of the scoring function are called the *match score* and the *contact score*, respectively.

An important feature of the scoring function is its additivity in the ligand atoms. The shapes of the penalty functions are of no further importance for the topic of this paper and thus will not be described here. For a full description of the model of interactions and the scoring function including the definition of the penalty functions f,f*, see Ref. 13.

Ligand fragmentation

In a first step, the ligand is decomposed into *components* by cutting at each acyclic rotatable bond (see Fig. 2a for an example). All acyclic bonds except for double bonds, triple bonds, and bonds to methyl and amine groups are defined as rotatable. Note that amide bonds are rotatable and, in fact, we have different torsion angles at an amide bond to allow for a slight twist in the amide plane. Ring systems are always located completely inside a component because parts of a ring system cannot be considered independently when ring conformations are constructed. The components of a molecule are the building blocks for generating fragmentations. We define an *unrooted component tree* in the following way: vertices are the components themselves and edges connect adjacent components i and j, in the sense that a bond connects atoms from i and j. An example of a component tree is shown in Fig. 2b. A *fragmentation* is a partition of the components of a molecule, such that every part, called *fragment*, is connected in the component tree.

The component tree becomes a rooted tree by defining the root to be an arbitrary component of the base fragment. The complex construction algorithm adds fragments to the placed part of the ligand, such that the set of placed components grows from the root to the leaves of the rooted component tree.

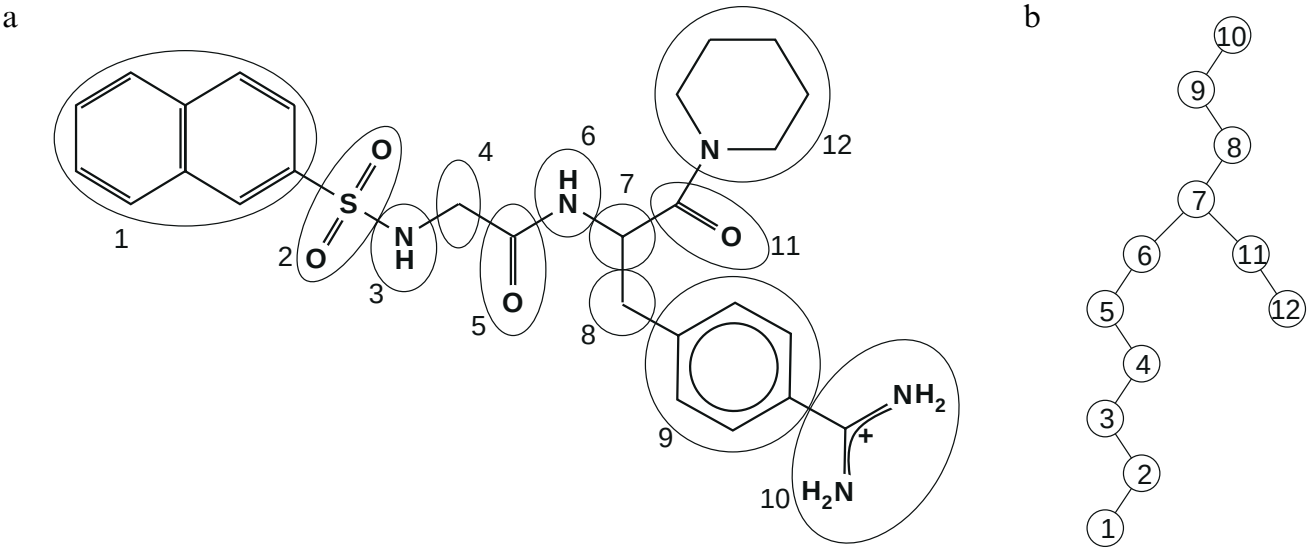

Fig. 2. Component tree of the thrombin inhibitor NAPAP: (a) decomposition of the ligand into 12 components; (b) tree structure of the ligand defined by the decomposition.

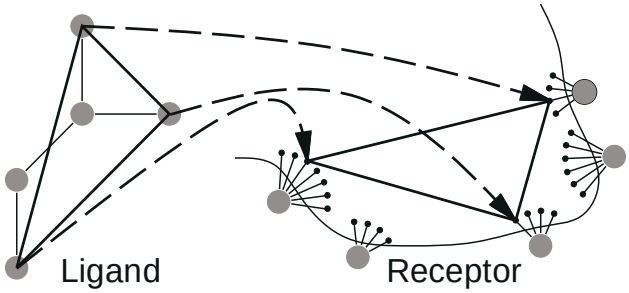

Fig. 3. The fragment placing algorithm: mapping three interaction centers (grey spheres) of the ligand onto three discrete interaction points in the active site (black dots) defines a unique transformation of the ligand into the active site.

In our experience, the complex construction algorithm produces the best results if the added fragments are small. Placing large fragments causes redundant computing, because components of the fragment near the atom with which the component is connected to its ancestor in the component tree are placed identically for multiple conformations of the added fragment. Thus, in the current fragmentation scheme in FLEXX, all fragments except for the base fragment consist of only one component. Note that the fragmentation is computed after a base fragment has been selected, which gives much more freedom in the selection process compared to a computation in the opposite order, namely performing the fragmentation before the base selection.

The base placement algorithm

To describe our docking algorithm, we use the following terms. A conformation of the ligand in combination with an orientation in the active site of the protein is called a *placement* or a *solution*. If only a part of the ligand is considered, we have a *partial placement*. With each partial placement, we associate the subset of the components describing the considered part of the ligand. Bonds connecting placed components with those that are not placed yet are called *outgoing bonds*.

In order to place the base fragment into the active site, we have adapted an algorithm from the area of computer vision, called *pose clustering* [20]. The main idea of the algorithm is to form triples of interactions between the ligand and the protein. From each triple, a transformation is computed such that all three interactions can occur simultaneously (see Fig. 3). Because the base fragment may be able to form more than three interactions, the computed transformations are clustered subsequently.

If the base fragment is very small, it may happen that three interactions cannot be formed simultaneously. Then FLEXX switches to another mode which forms only pairs of interactions. Now, computing a transformation directly is not possible anymore. The remaining degree of freedom (rotation about the axis through the two interaction centers of the ligand) is fixed in a few favorable positions, leading to a set of placements from each matched pair of interactions. For distinction, we call these two variants the *line* and the *triangle* variant of the base placement algorithm. A complete description of the triangle variant of the base placement algorithm can be found in Ref. 15. Deriving the line variant from it is straightforward.

The complex construction algorithm

After a set of placements for the base fragment has been computed, FLEXX starts the complex construction phase. The *i*th iteration starts with a set of partial placements with ligands constructed up to and including fragment i–1, called a *solution set* in the following, and ends with a set of placements with ligands constructed up to and including fragment i. Each iteration consists of the following steps for each placement in the solution set: adding the next fragment in all possible conformations, searching for new interactions, optimizing the positions of the partial ligand, selecting a new solution set, and clustering the solution set.

For the extension to handle multiple base fragments, which will be explained below, only the selection step and the clustering within the complex construction phase are of importance and will be described here. The remaining part of the algorithm does not have to be modified for this extension (see Ref. 13 for a complete description of the algorithm).

For selecting the solution set, we use a very simple greedy criterion: the scoring function is evaluated for each partial placement generated and the 500 best-scoring placements are taken into the solution set.

The clustering is based on a complete-linkage hierarchical clustering using the rms deviation between different placements as the distance measure. To avoid the clustering of placements which differ only in the directions of outgoing bonds, additional constraints restricting the angle and the distance between the next atoms connected to the outgoing bonds must be fulfilled inside a cluster. After clustering, only the best-scoring placement from each cluster is taken into the solution set.

Methods for automating the base selection

What is a good base fragment?

Before we can automate the base selection, we have to point out the major desirable features of a base fragment. Thinking reverse, a *good* base fragment is a fragment which leads to low-energy docking solutions. Thus, we can derive the following favorable features for a base fragment:

Placeability: The base placement algorithm must be able to place the fragment. This means that, on the one hand, the number of interactions that can be formed must be large enough and, on the other hand, the number of conformations of the fragment must be small enough.

Specificity: Due to the incremental construction method, the fragment must be specific enough, such that it can be placed independently from the rest of the ligand. This means that the fragment alone must have energetically favorable positions in the active site which agree with their positions if the full ligand is placed.

Placeability can be transformed easily into straightforward criteria by analyzing the base placement algorithm. A fragment must have at least two (three) interaction centers, one (two) of them must be centers of tight interactions, such that it can be placed with the line variant (triangle variant) of the base placement algorithm.

The run time of the base placement algorithm essentially grows linearly with the number of conformations of the base fragment. To avoid extremely long run times in some cases, we restrict the number of conformations of a base fragment to be at most 30.

Specificity is much harder to describe. The main reason is that specificity depends on the protein in addition to the fragment. In principle, the probability that a fragment can be placed correctly into the active site independent from the rest of the ligand increases with the number of interactions the fragment is able to form. The problem is to avoid the selection of a fragment which makes interactions only to surrounding water molecules instead of the protein. Therefore, the protein must be taken into account, and a simple way of doing so is to select a set of base fragments instead of a single one. Then, the subsequent algorithmic phases (base placement and complex construction) select the base fragment implicitly.

As a result of these observations, we define the following scoring function that estimates the eligibility of a fragment for being a base fragment. Parameters of the function are the number of centers of tight interactions n_{ric}, the total number of interaction centers n_{ic}, and the number of conformations n_c of the fragment:

$$f_{bs}(n_{ric},n_{ic},n_c) = 1000 \min(n_{ric},4) + 100 \min(n_{ic},6) - n_c$$

In the selection process, we are looking for fragments maximizing f_{bs}. f_{bs} can be motivated as follows. If we are comparing two fragments for their eligibility of being a base fragment, we are considering the three parameters in the order given above. The first criterion is the value of n_{ric}. This criterion becomes less important if the value exceeds a given threshold (4 in this case). Then (or in the case that both fragments have equal n_{ric} values), we are comparing the second criterion, which is the value of n_{ic}. Only if again both fragments exceed the threshold (6 in this case), we take the fragment with fewer conformations (n_c).

Rules for selecting a set of fragments

As mentioned above, we want to select a set of base fragments instead of a single one in order to avoid an analysis of the protein during the base selection. The set of base fragments should have the following properties, called the *set properties*:

(1) *No base fragment is fully contained in another base fragment*. Assume we have two fragmentations A and B, such that the base fragment of A is contained in the base fragment of B. Then, using fragmentation B will always lead to better docking results than using A, because more information is available during the base placement algorithm when using fragmentation B.

(2) *Each component occurs in at most two base fragments*. This property ensures that the base fragments are distributed over the ligand.

(3) *Each component in a base fragment must be either necessary for the connectivity of the fragment or it must have interaction centers*. This third property avoids the fact that an unimportant component is added to a fragment only to fulfill the first property.

The base selection algorithm

Given a ligand and its component tree, the goal of the base selection algorithm is to find a set of base fragments with high f_{bs} values which has the set properties. Assume first that we want to select a single base fragment. Because a fragment must be connected in the underlying component tree, the set of possible base fragments is small. In addition, we restrict the size of the base fragment to contain at most three components. Thus, we can simply enumerate all the possible fragments and select the fragment with the maximal f_{bs} value.

To select a set of base fragments, we iterate the selection process for a single base fragment. In each iteration, the algorithm selects the best-scoring fragment not violating the set properties with the previously selected fragments. The iteration terminates if the set of base fragments cannot be extended or the set has reached a fixed maximum size (four, in practice).

Extending the complex construction algorithm to multiple base fragments

In this section, we explain the extension of the subsequent phases of the docking algorithm in order to deal with a set of base fragments. The base placement algorithm does not have to be changed. This algorithm is simply executed for each of the selected base fragments and the placements are merged to a single solution set.

In the complex construction algorithm, the overall loop of adding a fragment to each solution of the previous iteration (or to each solution of the base placement phase in the first iteration) remains the same. The steps which must be adapted to multiple base fragments are the selection step and the clustering step at the end of an iteration.

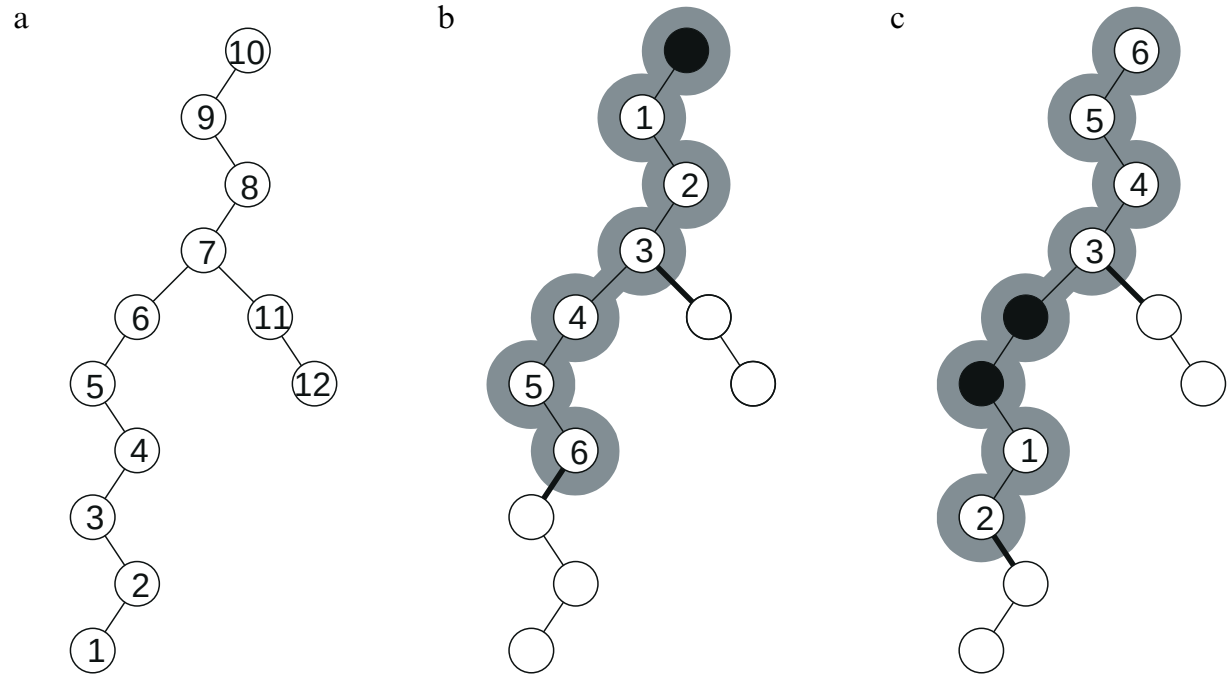

Fig. 4. (a) The component tree of NAPAP, already shown in Fig. 2b. (b) and (c) show the already placed components of two placements after six iterations of the complex construction algorithm with two different base fragments. The base fragments are denoted by black nodes. The nodes contain the number of the iteration of the complex construction algorithm, in which the corresponding component was placed. Nodes without a number are not placed yet. The set of placed components is highlighted in dark grey. Note that this subset relationship occurs first in the sixth iteration. The outgoing bonds used in the distance computation are shown in bold.

Selecting the set of survivors

To select the solution set, which will be further considered in the next iteration, we use a variant of our greedy strategy selecting the k best-scoring solutions. For the selection process, only the second and third terms of the scoring function (match and contact scores) are of importance and will be considered in the following. The first part (entropy) is constant for a fixed ligand. We will call the sum of the match and contact scores the *partial scoring function* E.

The problem arising with the greedy strategy is that the scoring values cannot be compared directly when different fragments are involved. Assume, for example, that we have two fragments A and B, such that A contains a charged group and B contains a hydrophobic ring. A is able to form much stronger interactions than B. Thus, if scoring values of placements for A and B are compared directly, even an energetically optimal placement of B will have a lower scoring value than most of the placements of A. This would result in eliminating all placements of B directly.

An obvious way to avoid such a phenomenon is to normalize the scoring function. We first estimate the maximal score for a fragment and then use the scoring value relative to this maximum for selecting favorable solutions. We have tried this approach with little success. Small fragments with only a few interacting groups are not as specific as large fragments and thus can be placed such that all interacting groups form interactions, which in turn causes a high relative score. This results in an overestimation of small fragments over large fragments. A more successful approach is to estimate the score of the whole ligand, given a partial placement.

Estimating the score for the whole ligand Let $C = \{c_0,...,c_{k-1}\}$ be the components of a ligand and let P be the set of all (partial) placements generated up to an arbitrary point in time during the docking algorithm. For a partial placement $p \in P$, we divide the set of components C into three disjoint sets $C_p(p)$, $C_b(p)$, $C_m(p)$ such that $C = C_p(p) \cup C_b(p) \cup C_m(p)$. $C_p(p)$ contains the components that are placed in p. C_b contains those components that are not placed in p but are placed in some other placement $p' \in P$. $C_m(p)$ contains the remaining components of the ligand. In the following, we will only consider the partial scoring function consisting of the match score and the contact score. The entropic part of the scoring function is constant for a fixed ligand and need not be considered.

Recall that the scoring function is additive in the ligand atoms. Thus, we can compute or estimate the score for each component separately and add these values to obtain an estimate of the score for the whole ligand. Applying the partial scoring function to the placement p, we get the score for each component of $C_p(p)$. For the remaining components, we estimate a score.

For a component c of $C_m(p)$, we estimate the maximal score the component can achieve. For each interaction center of c, we assume that it can be involved in a fixed

number of interactions with optimal score. The number of interactions is one or two, depending on the size of the interaction surface. The sum of these optimal scores for each interaction center forms an estimate of the match score of c.

To estimate the contact score of c, we have to estimate the number of protein atoms which can be in contact with an atom a of c. The protein atoms forming a contact to a lie in a shell around a. Thus, the more the atoms of the ligand located around a, the fewer the contacts to the protein that can be formed. Let nb(a) be the number of bonds to non-hydrogen atoms attached to a. Then we estimate the number of protein atoms getting in contact with a by max $\{0{,}5 - \mathrm{nb(a)}\}$. For each contact we assume the maximal score possible (ΔG_{lipo}).

Now we consider the components of $C_b(p)$. For the components of $C_m(p)$, we have estimated the score which can be achieved in the best case. For most components, this estimate is greater than the score which can be achieved for it later. A component c of $C_b(p)$ is already placed in some other placements. We can estimate the score of c by the best score achieved for c in those placements.

To compute this estimate for all components placed in any of the placements during a docking run, we maintain the best score achieved for each component during the computation. These *best-score values* are the estimates for the scores of the components of $C_b(p)$. In order to avoid computed results depending on the order in which the placements are generated, we update the best-score values only at the beginning of each iteration of the complex construction algorithm.

Two-step selection process For a further improvement of the complex construction algorithm, we introduce a two-step selection process. In this way we avoid that fragmentations leading to low-energy placements are deleted from the solution set too early in the construction process.

In the first selection step, we keep the 100 best-scoring solutions for each fragmentation separately. In the second step, we keep the 400 best-scoring solutions from those not belonging to the best 100 with respect to their fragmentation. Thus, the number of solutions taken into account in the next iteration is $400 + 100n_f$, where n_f is the number of selected base fragments.

Clustering partial solutions

The fact that different parts of the ligand are placed and compared during one iteration of the complex construction algorithm makes modifications of the cluster algorithm necessary. Because base fragments can contain different numbers of components, the set of considered components of one partial placement can be a proper subset of the components of a second partial placement. Figure 4 gives an example for such a situation.

In order to enable the cluster algorithm to deal with a set of placements with different sets of placed components, we have to change the definition of distance between placements and the way a representative placement is selected from a cluster of placements.

Let p and p' be two placements and c(p) and c(p') the set of placed components. So far, the distance function used for clustering was the rms distance between the two placements together with additional constraints for outgoing bonds (see the description of the complex construction algorithm). Having a set of base fragments, c(p) and c(p') need not be identical anymore. If neither $c(p) \subseteq c(p')$ nor $c(p') \subseteq c(p)$, we set the distance to infinity. Otherwise, the distance is computed by measuring the rms deviation of the atoms of $c(p) \cap c(p')$. Additional constraints for outgoing bonds are applied to those bonds which are outgoing both in p and in p'. In Figs. 4b and c, the outgoing bonds are marked by bold lines. For comparing the placements in Figs. 4b and c, the additional constraints are applied to the outgoing bond from node 7 (having the label 3 in Figs. 4b and c).

After clustering, a cluster is reduced to a single placement by selecting a representative out of the cluster. The distance function implies that, inside a cluster C, the components of two placements p,p' are either identical or the components of p form a subset or superset of the components of p'. The representative is selected from the set of placements with the maximum number of components C'. Inside C', the placement with the highest score is selected to be the representative. All the other placements are eliminated.

Results

In this section, we compare the results of the docking algorithm with automatic and with manual base selection. We perform this comparison on a set of protein–ligand complexes together with the manually selected base fragments that we also considered in Ref. 13.

Preparing the test set

For each complex of our test suite, we have performed the following steps.

The ligand is extracted from the PDB file and converted into SYBYL mol2 file format [21]. Hydrogens are added in reasonable geometries and correct atom types, bond types, and formal charges are assigned. Then the ligand is energy-minimized using the Tripos force field [21].

For the protein, a so-called receptor description file (rdf) is manually generated containing information to resolve ambiguities in the PDB file and to assign all kinds of physicochemical information to the protein needed by FLEXX (atom and bond types, formal charges, interacting groups, protonation).

TABLE 1
SUMMARY OF DOCKING RESULTS

PDB	Manual base selection						Automatic base selection					
	No. of sol.[a]	Lowest energy		Best prediction			No. of sol.[a]	Lowest energy		Best prediction		
		ΔG[b]	Rms[c]	ΔG[d]	Rms[e]	Rank[f]		ΔG[b]	Rms[c]	ΔG[d]	Rms[e]	Rank[f]
5tim	6	–16.30	1.99	–10.35	0.87	3	6	–16.30	1.99	–10.35	0.87	3
1ldm	47	–32.89	0.74	–32.89	0.74	1	47	–32.89	0.74	–32.89	0.74	1
2phh	178	–30.49	0.61	–30.49	0.61	1	178	–30.49	0.61	–30.49	0.61	1
3ptb	30	–26.21	0.54	–26.21	0.54	1	30	–26.21	0.54	–26.21	0.54	1
1ulb	184	–18.03	5.41	–17.06	0.40	2	184	–18.03	5.41	–17.06	0.40	2
3tpi	215	–24.02	0.58	–24.02	0.58	1	525	–24.02	0.58	–24.02	0.58	1
4ts1	201	–25.71	0.90	–25.51	0.44	2	246	–25.71	0.90	–25.51	0.44	2
4dfr	77	–65.46	0.89	–65.46	0.89	1	218	–65.46	0.89	–65.46	0.89	1
1stp	130	–26.96	1.47	–26.00	1.01	2	211	–27.59	0.80	–27.59	0.80	1
1dwd	335	–44.64	1.02	–41.65	0.61	11	474	–44.64	1.02	–41.65	0.61	11
1dwc	218	–33.83	4.53	–31.95	2.00	6	371	–35.73	8.12	–31.95	2.00	21
6rsa	132	–39.36	0.39	–39.36	0.39	1	237	–39.36	0.39	–39.36	0.39	1
1rnt	124	–42.37	1.88	–40.18	1.46	4	215	–42.37	1.88	–40.18	1.46	5
1tmn	237	–38.67	0.87	–38.67	0.87	1	517	–38.67	0.87	–38.67	0.87	1
4tln	332	–28.86	3.68	–8.69	0.92	180	389	–28.86	3.68	–14.58	1.80	237
3cpa	73	–33.89	3.34	–27.47	0.89	25	165	–33.89	3.34	–27.47	0.89	26
2ctc	158	–36.07	1.67	–34.45	0.85	2	158	–36.07	1.67	–34.45	0.85	2
4phv	29	–37.63	1.71	–37.45	1.04	2	28	–38.41	1.38	–33.31	1.26	7
121p	465	–49.44	2.36	–39.13	1.14	11	572	–65.92	2.24	–39.13	1.14	57

[a] Number of solutions produced.
[b] ΔG predicted of the solution with the lowest score.
[c] Rms deviation of the solution with the lowest score.
[d] ΔG predicted of the best prediction.
[e] Rms deviation of the best prediction.
[f] Rank by score of the best prediction.

All protein atoms with a distance of up to 6.5–8 Å from any atom of the co-crystallized ligand are selected as active site atoms. The distance value is chosen by a visual inspection of the protein such that the whole pocket is part of our active site definition. In addition, a surface file containing all atoms that are solvent-accessible is generated. Since this is done only once for a protein, the run time (about 30–60 s on a workstation) is not taken into account in the results presented below.

Some selection examples

Figures 5, 6, 8, 10, 11, and 13 present some examples of multiple automatic base selections. In all the test cases, the manually selected fragment (highlighted in red in Figs. 5–13) is selected as one of the automatically selected base fragments as well. To emphasize the ability of the algorithm to select overlapping base fragments, we remark that overlapping base fragments are selected in nine of the 13 test cases in which more than one base fragment is selected.

Quality

Table 1 details the comparison of the docking results obtained with FLEXX with manual and automatic base selection. The results obtained with manual base selection differ slightly from those presented in Ref. 13 because of minor improvements in the code and in the static data files (torsion angles and interaction geometries). We will not discuss these minor changes here.

We recognize that the solution sets with automatic base selection (column 8) are larger than those with manual base selection (column 2). During the complex construction phase with automatic base selection, more placements are maintained because of two reasons. First, two placements a and b can only be clustered during the complex construction phase if the placed part of the ligand in a is part of the placed part of the ligand in b or vice versa. Second, the 100 best-scoring placements from each fragmentation are stored separately such that they cannot be deleted in the selection step (see the 'Methods for automating the base selection' section).

Besides the number of solutions, Table 1 contains the energy or scoring value and rms deviation from the crystal structure for the highest ranking solution (with the lowest scoring value) as well as these values plus the rank for the best prediction. The rms deviations are computed without hydrogen atoms and under consideration of internal symmetries of the ligand. As the best prediction, we normally take the highest ranking solution with an rms deviation less than 1.0 Å. In those cases where no solution fulfills this criterion, a solution with low rms and high rank is selected. In the case of 1rnt, we take a solution with higher rms although a solution with rms below 1.0 Å is found (with lower rank), because the selected sol-

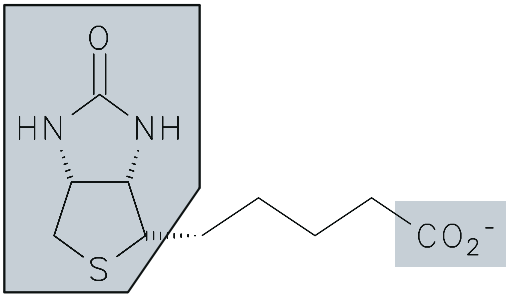

Fig. 5. Base fragments of 1stp: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

ution differs only in the conformation of the ribose unit, which does not make any interactions with the protein.

For the five test cases with small ligands (5tim, 1ldm, 2phh, 3ptb, 1ulb), the results do not differ between manual and automatic base selection. This is not surprising because the automatic base selection first selects the whole ligand and then no other selection is possible due to the selection rules described in the 'Methods for automating the base selection' section. Thus, the subsequent computations for small ligands are identical with manual and with automatic base selection.

In nine of the remaining 14 test cases with larger ligands, the highest ranking solution and the best prediction are identical for manual and automatic base selection (3tpi, 4ts1, 4dfr, 1dwd, 6rsa, 1rnt, 1tmn, 3cpa, 2ctc). In these cases, the placements generated by the base fragment, which is also selected manually, dominate the resulting set of solutions.

The results for the remaining five test cases (1stp, 1dwc, 4tln, 4phv, 121p) differ with manual and automatic base selection. These five test cases and the 1tmn complex will be discussed in more detail. 1tmn is selected for discussion because of an interesting input situation for the base selection algorithm.

1stp

1stp contains the structure of the complex between streptavidin and biotin. FLEXX selects two base fragments: the bicycle containing the ureido group and the carboxylate group (Fig. 5). Manually, we have selected the bicycle. For the crystal structure, both fragments form hydrogen bonds to the protein: the bicycle to Asn[23], Ser[27], Tyr[43], Ser[45] and Asp[128] and the carboxylate group to Asn[49] and Ser[88]. Performing the base placement, nearly correct positions are found for both fragments. The docking procedure that starts with the less specific carboxylate group produces a slightly better result with a lower energy as well as a smaller rms deviation. This is an example of a tightly binding ligand, in which the selection of the base fragment is of minor importance. Both possible selections lead to at most the same result.

1dwc

The complex between thrombin and the inhibitor argatroban is one of our more complicated test cases. The difficulties arise from the interaction geometries in the crystal structure. In our interaction model, these interaction geometries yield only weak-scoring contributions such that the total score of the crystal configuration is about –20 kJ/mol. This is much higher than the score for the structures favored by FLEXX, which achieve less than –30 kJ/mol. The base fragments selected manually and automatically are shown in Fig. 6.

FLEXX places the guanidino group of argatroban such that two chelate-like hydrogen bonds to Asp[189] in the S1 pocket can be formed (see Fig. 7). Although the following alkyl chain grows into the same direction for the FLEXX solutions and the crystal structure, the different placement of the guanidino group causes higher rms deviations. With manual base selection, a binding mode comparable to the crystal structure is found at rank 6 with an rms deviation of 2.0 Å as shown in Fig. 7.

With automatic base selection, FLEXX selects the four base fragments shown in Fig. 6. Starting with the base fragment containing the carboxylate group and the piperidine ring leads to the highest ranking solution. This solution was not found with manual base selection. Because the guanidino group is not placed in the S1 pocket, this is a solution of less practical importance. The place-

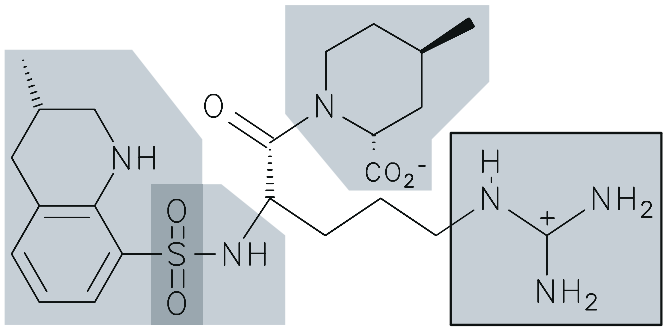

Fig. 6. Base fragments of 1dwc: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

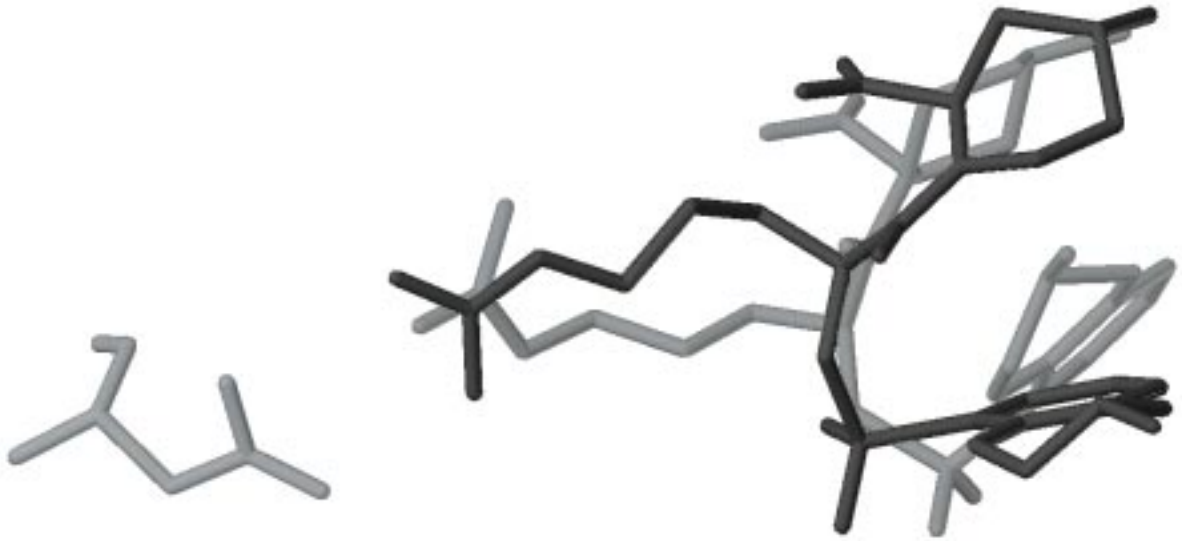

Fig. 7. 1dwc: comparison of the crystallographic orientation (light grey) to the placement predicted by FLEXX with rank 6 (dark grey). Asp^{189} is shown on the left side.

ment with an rms deviation of 2.0 Å is moved from rank 6 to 21. This example shows that alternative placements can be found with automatic base selection. Even if these placements are not of practical interest, they are helpful in studying the difficulties for scoring the placements.

1tmn

1tmn is the complex between thermolysin and an inhibitor (*N*(1-carboxy-3-phenyl)L-leucine-L-tryptophan). The base fragments selected manually and automatically are shown in Fig. 8.

Consider the two N-CH-CO_2^- fragments contained in the ligand. Taking the first such fragment (between the leucyl and the phenyl-propyl group) as the base fragment yields a correct solution while taking the second (between the leucyl and the tryptophyl group) does not. In contrast to 1stp, 1tmn is an example where the base selection is critical. Because the two fragments can only be distinguished if a larger portion of the ligand is considered, selecting a single base fragment leading to the correct solution is rather complicated. Here, the advantage of the multiple base selection becomes obvious (see Fig. 9).

4tln

The complex between thermolysin and a small inhibitor (L-leucyl-hydroxylamine) is a surprisingly complicated test case for FLEXX. Because the ligand is small, neither the ligand flexibility nor the selection of an appropriate base fragment is problematic. The reasons for the high rms deviation are an energetically unfavorable torsion angle in the ligand crystal structure as well as the fact that the ammonium group of the ligand is not able to form hydrogen bonds at the crystal position (see also Ref. 13).

FLEXX automatically selects two base fragments, both containing the central NH-C–O^- fragment, the first together with the hydroxy group and the second together with the ammonium group (see Fig. 10). In the resulting list of docking solutions, both fragmentations lead to energetically comparable results. Because of the restricted set of solutions during complex construction, the most similar one (rms deviation of 0.92, rank 180) with manual base selection falls out of the solution set. The highest ranking solution remains the same with manual and automatic base selection.

4phv

With 17 rotatable bonds contained in the ligand, the complex between HIV-1 protease and the inhibitor L700,417 has the largest complexity in our test set. FLEXX selects four base fragments, the amide group and the indanyl group from each side of the ligand (see Fig. 11).

Due to the smaller set of solutions per fragmentation when more than one base fragment is selected, the position of the main chain of the ligand differs slightly more from that in the crystal structure than is the case with one manually selected base fragment. Nevertheless, both structures should be similar enough to fall into the same local minimum if a subsequent energy minimization would be performed.

Because the two substituents at the central carbon atom (note that the complete ligand has no symmetry) are identical, there are two very similar binding modes which can be turned into each other by simply rotating the ligand by 180° around the central carbon atom. If one indanyl group is selected as the base fragment, only one of these two binding modes is found because the indanyl

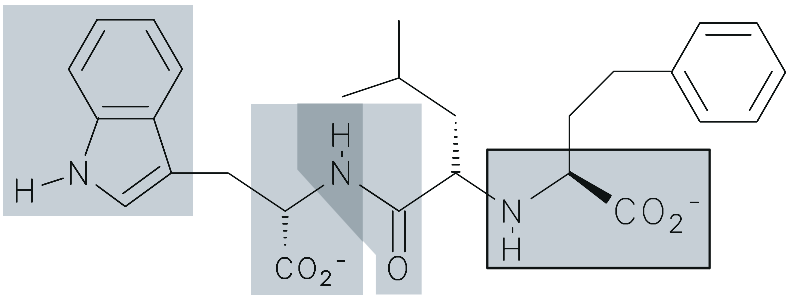

Fig. 8. Base fragments of 1tmn: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

group fits slightly better into the pocket formed by chain A than into the pocket formed by chain B (the pockets formed by chains A and B are geometrically very similar but not identical). In contrast, with multiple base fragments selected, both binding modes are detected at ranks 7 and 9 (see Fig. 12).

This test case outlines another advantage of multiple base selection. Because there is no bias to a special part of the ligand during the docking computation, the possibility of finding different binding modes increases.

121p

121p contains the structure of the complex between an oncogene protein (H-RAS P21) and an inhibitor (guanosine 5'-β,γ-methylene-triphosphate). FLEXX selects the four base fragments shown in Fig. 13.

Docking with the manually selected guanine group leads to a placement with 1.14 Å rms from the crystal structure at rank 11. The same solution is found with multiple base fragments, but now at rank 57. The lower rank results from the fact that with multiple base fragments alternative solutions with lower energy are found. One example of such a solution is the highest ranking one with scoring value −66 kJ/mol (compared to −50 kJ/mol for the highest ranking solution with manual base selection). This placement is generated by building up the ligand in the opposite order (starting with the terminal phosphate group). Thus, the terminal phosphate group can be placed at a nearly ideal position. This solution cannot be found when the terminal phosphate group is added in the last iteration. The discussed placements are shown in Fig. 14.

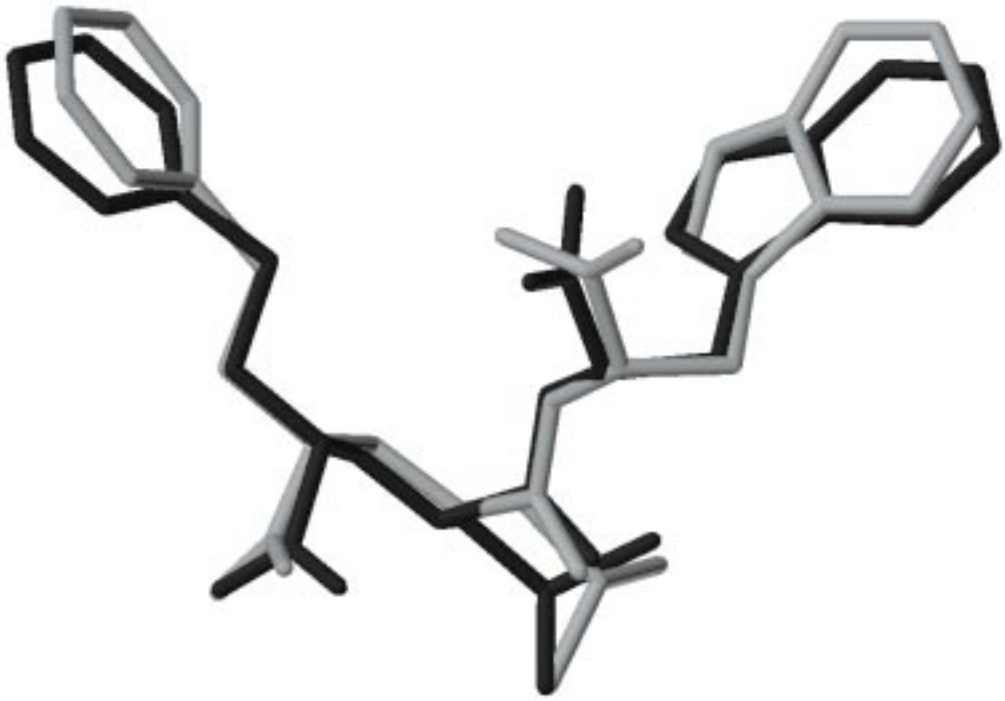

Fig. 9. 1tmn: comparison of the crystallographic orientation (light grey) to the placements predicted by FLEXX with highest rank – manual and automatic base selection (dark grey).

Run time

Table 2 contains the elapsed run times of the 19 test cases on a Sun UltraSPARC workstation with a single 167 MHz processor and 64 MB of main memory. The times for loading the input data, assigning the physicochemical information and selecting a set of base fragments as described above are about 10 s per test case and are omitted here.

The time needed for placing base fragments increases from 21 s to 1:14 min on average. This results from two effects: first the base placement must be performed once for each fragmentation. But even the time for each single base placement increases from 21 s to 27 s on average. Automatically selected base fragments have larger conformational sets compared to the manually selected ones. This results from the weighting in the base selection function, in which the number of conformations of a fragment plays only a minor role compared to the number of interaction centers.

The complex construction time increases only slightly from 33 s to 54 s on average (only test cases with nonzero complex construction time are considered). This is mainly due to the larger solution set which will be considered in each iteration if more than one base fragment is selected. Recall that, if k base fragments are selected, $400 + 100 * k$ solutions are considered in each iteration.

In summary, the total run time for a docking run increases from 45 s to 2:08 min on average.

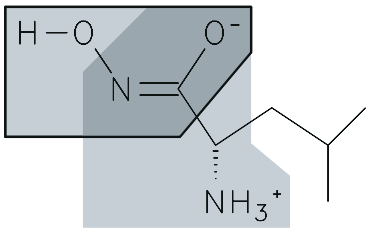

Fig. 10. Base fragments of 4tln: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

Discussion

The major drawback of docking by incremental construction is the necessity of selecting a base or anchor fragment. The results of a docking run depend strongly on this selection, which underlines the importance of this step. On the other hand, the automatized selection of a single base fragment which leads to a successful docking is a complicated task. The reason is that, in principle, comprehensive information about the ligand and the protein is necessary to make this selection.

A simple way to avoid this problem is to delay the selection of the base fragment into the following phases of the algorithm. We achieved this by selecting a set of potentially successful base fragments and applying the base placement algorithm to each of them. The real selection is implicitly done during the complex construction, in which only the best-ranking placements survive.

TABLE 2
SUMMARY OF RUN TIMES

PDB	No. of comp.[a]	No. of frag.[b]	Manual base selection[c]		Automatic base selection[c]	
			BP[d]	CC[e]	BP[d]	CC[e]
5tim	1	1	2.30	0.00	2.40	0.00
1ldm	2	1	10.92	0.00	10.83	0.00
2phh	3	1	26.23	0.00	25.92	0.00
3ptb	2	1	14.21	0.00	13.58	0.00
1ulb	1	1	36.81	0.00	36.14	0.00
3tpi	8	4	4.72	20.91	29.18	56.73
4ts1	5	3	3.22	12.36	1:12.78	18.23
4dfr	10	4	15.26	47.28	3:08.27	1:16.18
1stp	6	2	4.05	11.24	5.20	48.46
1dwd	12	4	21.08	59.50	53.52	1:43.01
1dwc	12	4	22.86	1:10.12	3:14.43	2:27.53
6rsa	4	3	19.77	9.57	20.20	16.12
1rnt	8	4	17.51	25.79	1:03.87	1:10.77
1tmn	15	4	15.58	1:05.07	43.74	2:34.66
4tln	5	2	2.60	6.24	1:23.70	12.72
3cpa	8	4	6.27	19.72	46.14	50.03
2ctc	5	1	1:02.38	4.32	1:02.98	4.52
4phv	18	4	1:18.15	1:13.89	2:53.90	2:43.03
121p	11	4	31.24	41.91	4:45.19	1:43.23

[a] Number of components of the ligand.
[b] Number of fragmentations.
[c] Run times in min:s.ms.
[d] Base placement.
[e] Complex construction.

First of all, this method leads to a completely automatic docking tool with only slightly differing results compared to docking with manual base selection and with an acceptable sacrifice of computation time. Besides an automation of the base selection, the strategy of docking with multiple base fragments has additional advantages. As we have seen in the case of 4phv, multiple binding modes are much easier to detect. Obviously, docking by incremental construction is biased towards the base fragment. Starting with a specific fragment leads to placements in which this fragment has a favorite position in the active site. In contrast, placements in which this fragment forms only minor interactions to the protein are eliminated. This bias does not occur if multiple base fragments are selected. The importance of this effect becomes clear in the test cases 4phv, 1tmn, and 121p.

Relation to other docking methods

There are several alternative approaches to docking. Here, we want to discuss only the relation of FLEXX to the main developments in recent years.

As mentioned in the Introduction section, genetic algorithms are widely used for tackling the docking problem [7–9]. In contrast to the deterministic optimization algorithm used in FLEXX, genetic algorithms are randomized, which complicates a comparison. A single run of a genetic algorithm has comparable run times to FLEXX, but a correct solution is found in a single run only with a certain probability. Because this probability of success is unknown for a specific test case, iterating the genetic algorithm is unavoidable.

Let us consider what happens if the test cases become more and more complicated. For a genetic algorithm, the probability of success decreases. To predict the correct binding mode, the number of iterations can be increased.

Because FLEXX is based on a heuristic approach, it may happen that the correct binding mode is not found. As for genetic algorithms, this becomes more probable for complicated test cases. At first glance, genetic algorithms seem to have an advantage because the number of iterations can easily be increased, and because FLEXX's algorithm is deterministic we cannot iterate (each iteration leads to the same result). Increasing the number of iterations of a randomized algorithm is a simple way to increase the coverage of the solution space. But this can also be done by adapting some parameters of FLEXX, for example the number of solutions further considered in each iteration.

Compared to genetic algorithms, the incremental construction method is a more systematic search strategy. For example, if two solutions in a solution set differ only in the placement of one component of the ligand, the incremental construction method handles the common part of both ligands together. If the genetic algorithm is iterated, similar solutions can be produced in subsequent iterations. Avoiding this redundant computing leads to much faster run times of FLEXX compared to genetic algorithms.

AutoDock is a tool for flexible ligand docking based on simulated annealing [22]. Simulated annealing is a randomized approach and thus the same arguments apply as for genetic algorithms. An advantage of AutoDock is that it does not need any kind of specific interactions like hydrogen bonds or salt bridges for placement. This enables the tool to dock ligands which bind via geometrically unspecific interactions like hydrophobicity or electrostatics completely. This must be paid for by long computation times. Fortunately, this situation does not occur often in drug design.

Hammerhead [14] is a new docking tool which is based on an incremental construction strategy. Thus, the method is similar to the Leach and Kuntz approach [11] as well as to FLEXX. Besides an alternative base fragment placement algorithm, the main difference compared to FLEXX is the fragmentation and construction strategy. Hammerhead divides the ligand into larger fragments containing up to three rotatable bonds. Each of these fragments is placed as a base fragment. Inside a fragment, a coarse-grained grid of torsion angles is used (120° increments). The fragments are linked during the placement of the next fragment. After each placement, a numerical energy

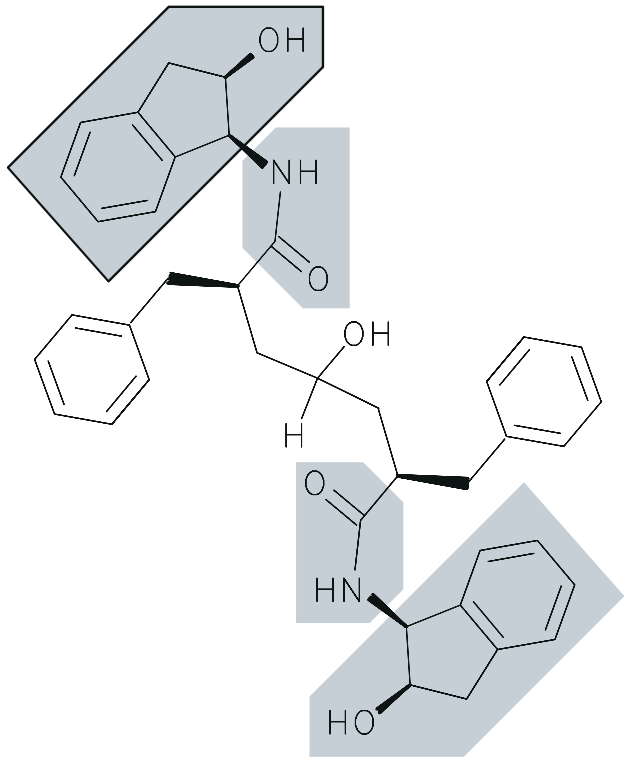

Fig. 11. Base fragments of 4phv: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

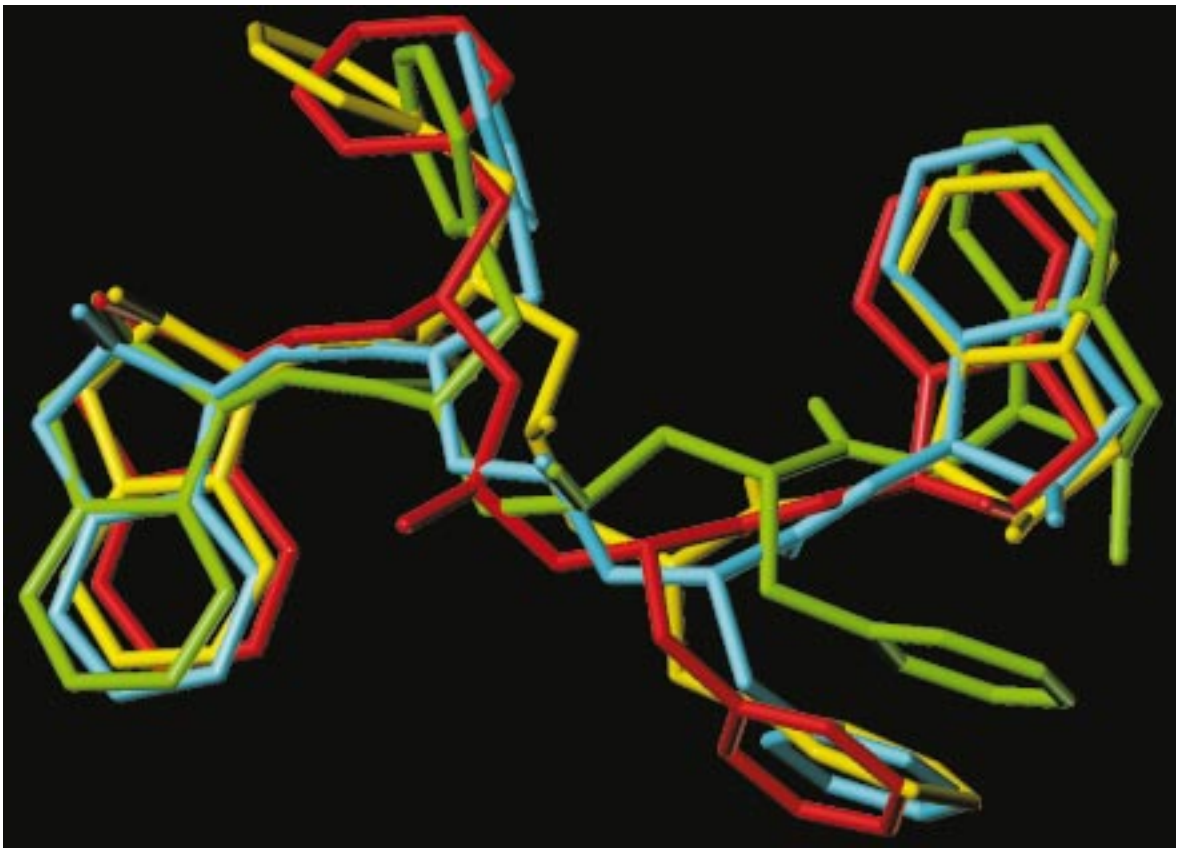

Fig. 12. 4phv: comparison of the crystallographic orientation (yellow) to the placements predicted by FLEXX – manual base selection, rank 2 (red), automatic base selection, rank 7 (cyan) and rank 9 (green).

minimization is performed. Because the main topic of this paper is the base selection, we only compare the fragmentation and base selection strategies here.

In contrast to Hammerhead, FLEXX follows the strategy of dividing the ligand into the smallest possible fragments (called components) observing some rules (e.g. not to cut ring systems, rigid bonds, or bonds to end-standing atoms). Base fragments can be assembled from these components, which enables the selection of more specific and probably overlapping base fragments. This additional degree of freedom during the selection is important for ligands which do not have a clearly dividable structure (see the 1tmn test case, for example).

Further improvements of FLEXX

With a multiple base selection, the time efficiency of the base placement algorithm becomes more important and we have to think about speeding up this phase. One step in this direction, which we took already, is the iter-

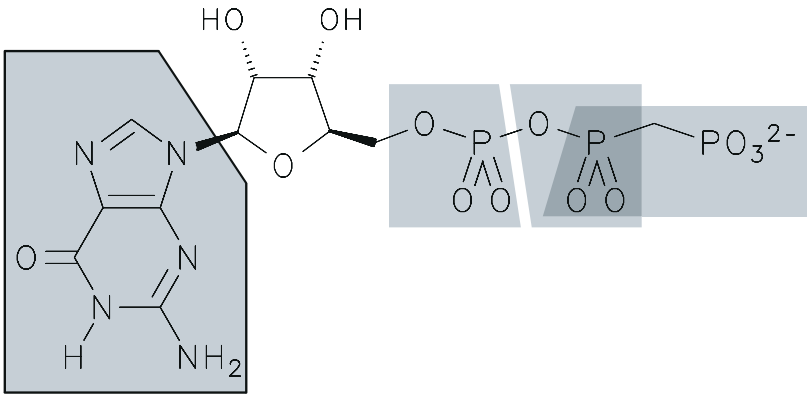

Fig. 13. Base fragments of 121p: manually selected are surrounded by boxes with a black border; automatically selected are surrounded by grey boxes.

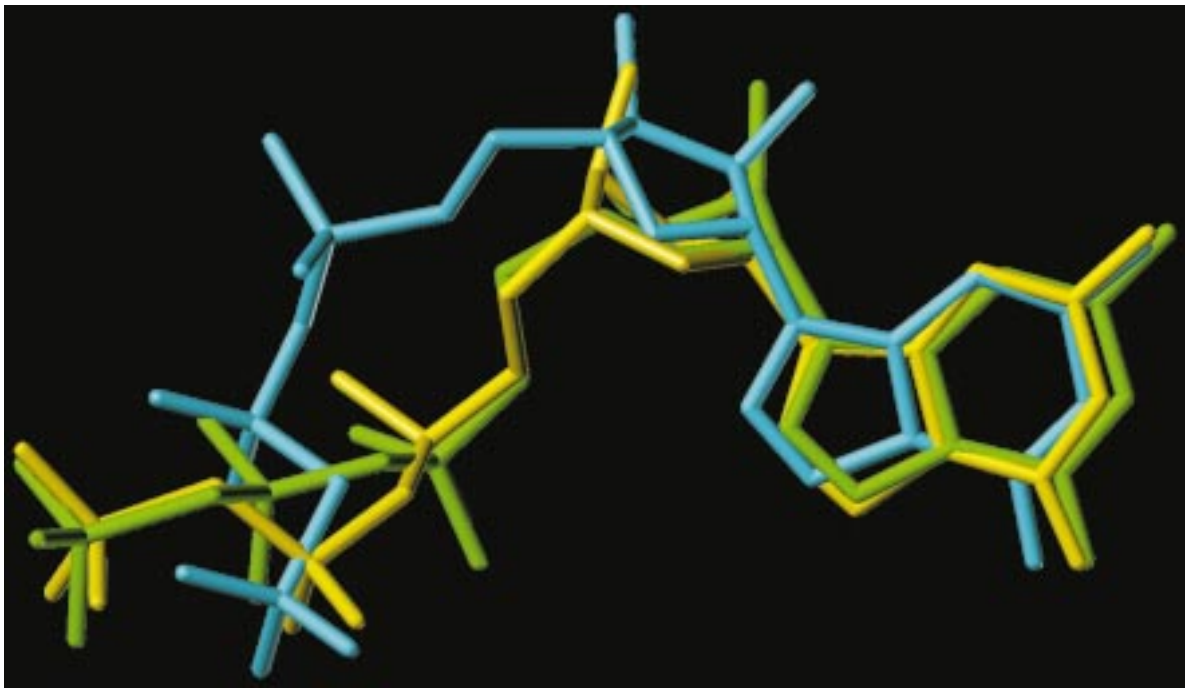

Fig. 14. 121p: comparison of the crystallographic orientation (yellow) to the placements predicted by FLEXX – manual base selection, rank 11 and automatic base selection, rank 57 (green), automatic base selection, rank 1 (cyan).

ative usage of the hierarchical cluster algorithm. Presently, the cluster algorithm is called a few times with an increasing distance threshold, which is much faster than calling the algorithm only once with the final distance threshold.

Currently, we are improving our interaction and scoring model (B. Kramer, in preparation). The major goal is to develop a consistent model comprising the scoring function, the handling of protein–ligand overlaps, and the placement of fragments. This model will enable the (almost) direct placement into local minima of the scoring function without a time-intensive numerical minimization.

The main restriction of FLEXX as well as of most other approaches to docking is the rigidity of the protein. However, FLEXX's time efficiency does not result from grid-based scoring evaluation, which would make the integration of protein flexibility complicated. Future work will include the consideration of protein flexibility without losing time efficiency.

Conclusions

This paper can be considered as the third in a series describing a highly efficient approach to the prediction of protein–ligand complexes (the docking problem). The approach is based on the incremental construction strategy, consisting of three phases: the selection of a base fragment, the placement of this base fragment into the active site of a protein, and the incremental construction of the complex based on the computed placements for the base fragment. The base placement and complex construction algorithm have been published in previous papers [13,15].

In this paper, we have presented a method for automatically selecting base fragments. First a scoring function is defined, which selects base fragments with ideal properties for placement with respect to the base placement algorithm.

One important property of a good base fragment is the specificity of the fragment. Here, specificity means that the fragment forms interactions to the protein in the protein–ligand complex such that it can be placed independently from the rest of the ligand. Because this property is hard to predict without actually placing the fragment, we extend our docking algorithm to handle not just a single base fragment but a set of base fragments. Then, the selection of the best base fragment is done implicitly during the adapted base placement and complex construction algorithm.

With this approach to the base selection, our docking tool FLEXX works without manual intervention. We have evaluated the quality of the docking predictions as well as the run time behavior of our method on a set of 19 complexes with known structure from the PDB. Compared to previous results, in which the same complexes are docked with a manual base selection, the algorithm with automatic base selection exhibits only a minor loss in quality. Although the computation time increases with the use of multiple base fragments, the approach needs only slightly more than 2 min on average on a single processor work-

station. Using parallel computers or workstation clusters enables the docking and energy estimation of about 700 ligands per processor per day.

Acknowledgements

This work was performed as part of the RELIWE Project, which is partially funded by the German Federal Ministry for Education, Science, Research, and Technology (BMBF) under Grant No. 01 IB 302 A. The cooperation partners in the RELIWE Project are BASF AG, Central Research Ludwigshafen, E. Merck Preclinical Research Darmstadt, EMBL Heidelberg, and GMD institutes IPSI (Institute for Integrated Publication and Information Systems) Darmstadt and SCAI (Institute for Algorithms and Scientific Computing) Sankt Augustin. We thank all our partners in the project for excellent cooperation, especially Gerhard Klebe from the University of Marburg for helpful comments on this work.

References

1 Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R.L. and Ferrin, T.E., J. Mol. Biol., 161 (1982) 269.
2 Blaney, J.M. and Dixon, J.S., Perspect. Drug Discov. Design, 1 (1993) 301.
3 Colman, P.M., Curr. Opin. Struct. Biol., 4 (1994) 868.
4 Kuntz, I.D., Science, 257 (1992) 1078.
5 Lengauer, T. and Rarey, M., Curr. Opin. Struct. Biol., 6 (1996) 402.
6 Lybrand, T.P., Curr. Opin. Struct. Biol., 5 (1995) 224.
7 Jones, G., Willet, P. and Glen, R.C., J. Mol. Biol., 245 (1995) 43.
8 Oshiro, C.M., Kuntz, I.D. and Dixon, J.S., J. Comput.-Aided Mol. Design, 9 (1995) 113.
9 Clark, K.P. and Ajay, J. Comput. Chem., 16 (1995) 1210.
10 Gehlhaar, D.K., Verkhivker, G.M., Rejto, P.A., Sherman, C.J., Fogel, D.B., Fogel, L.J. and Freer, S.T., Chem. Biol., 2 (1995) 317.
11 Leach, A.R. and Kuntz, I.D., J. Comput. Chem., 13 (1992) 730.
12 Rarey, M., Kramer, B. and Lengauer, T., In Rawlings, C. et al. (Eds.) Proceedings of the Third International Conference on Intelligent Systems in Molecular Biology, AAAI Press, Menlo Park, CA, U.S.A., 1995, pp. 300–308.
13 Rarey, M., Kramer, B., Lengauer, T. and Klebe, G., J. Mol. Biol., 261 (1996) 470.
14 Welch, W., Ruppert, J. and Jain, A.N., Chem. Biol., 3 (1996) 449.
15 Rarey, M., Wefing, S. and Lengauer, T., J. Comput.-Aided Mol. Design, 10 (1996) 41.
16 De Clercq, P.J., Tetrahedron, 40 (1984) 3717.
17 Klebe, G. and Mietzner, T., J. Comput.-Aided Mol. Design, 8 (1994) 583.
18 Böhm, H.-J., J. Comput.-Aided Mol. Design, 6 (1992) 593.
19 Böhm, H.-J., J. Comput.-Aided Mol. Design, 8 (1994) 243.
20 Linnainmaa, S., Harwood, D. and Davis, L.S., IEEE Trans. Pattern Anal. Machine Intell., 10 (1988) 634.
21 SYBYL, Tripos Associates Inc., St. Louis, MO, U.S.A., 1994.
22 Morris, G.M., Goodsell, D.S., Huey, R. and Olson, A.J., J. Comput.-Aided Mol. Design, 10 (1996) 293.