Journal of Computer-Aided Molecular Design, 11 (1997) 209–228.

J-CAMD 395

A comparison of heuristic search algorithms for molecular docking

David R. Westhead*, David E. Clark** and Christopher W. Murray***

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

Received 20 November 1996
Accepted 4 February 1997

Keywords: Ligand–protein docking; Molecular recognition; Evolutionary algorithms; Simulated annealing; Tabu search

Summary

This paper describes the implementation and comparison of four heuristic search algorithms (genetic algorithm, evolutionary programming, simulated annealing and tabu search) and a random search procedure for flexible molecular docking. To our knowledge, this is the first application of the tabu search algorithm in this area. The algorithms are compared using a recently described fast molecular recognition potential function and a diverse set of five protein–ligand systems. Statistical analysis of the results indicates that overall the genetic algorithm performs best in terms of the median energy of the solutions located. However, tabu search shows a better performance in terms of locating solutions close to the crystallographic ligand conformation. These results suggest that a hybrid search algorithm may give superior results to any of the algorithms alone.

Introduction

The safe and effective action of a pharmaceutical agent within the human body depends upon the selective recognition of the drug molecule by the appropriate target receptor. This *molecular recognition* is governed by the interplay of a number of factors such as steric, electrostatic and hydrophobic interactions. The sum of the free energy of these interactions is termed the *binding affinity* of the molecule for the receptor and is governed, in part, by the geometry of the ligand–receptor complex. The early stages of drug discovery, usually termed *lead generation*, could be significantly expedited if there existed a method whereby the geometry of the ligand–receptor complex and the binding affinity of a given molecule for a receptor of known structure could be reliably estimated without resorting to the experimental techniques of synthesis, co-crystallisation and assay. In computer-aided molecular design (CAMD), the search for methods for the ab initio prediction by computer of the binding geometry and binding affinity of two molecules is termed the 'docking problem'. Because of its potential application in CAMD, the docking problem has received much attention over the years and the progress made has been reviewed in a number of recent articles [1–6].

The earliest docking programs considered only the translational and orientational degrees of freedom of the ligand with respect to the receptor, e.g. Ref. 7. However, more recently, with advances in the power of the available computer hardware and increasingly sophisticated software algorithms, it has become possible to take into account routinely the internal conformational flexibility of the ligand ('flexible' docking) [8–23]. Limited conformational flexibility of the receptor is also being permitted in some approaches [11,16]. Clearly, as the number of degrees of freedom being explored in the docking problem increases, the size of the search space rapidly becomes enormous; Gehlhaar et al. [14] estimate that, in one of their examples, the search space comprised at least 10^{30} solutions. Faced with such a situation, it is obvious that

*Present address: EMBL Outstation, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, U.K.
**Present address: Dagenham Research Centre, Rhône-Poulenc Rorer Ltd., Rainham Road South, Dagenham, Essex RM10 7XS, U.K.
***To whom correspondence should be addressed.

Abbreviations: GA, genetic algorithm; EP, evolutionary programming; SA, simulated annealing; TS, tabu search; RS, random search; COM, centre of mass; rms, root-mean square; ICM, internal coordinate modelling; CPU, central processing unit; DHFR, dihydrofolate reductase; MTX, methotrexate; DANA, 2,3-dehydro-2-deoxy-*N*-acetylneuraminic acid; NAPAP, N^{α}-(2-naphthyl-sulphonyl-glycyl)-DL-*p*-amidinophenylalanyl-piperidine; FIFO, first-in, first-out.

fast and effective search algorithms are of crucial importance if the docking problem is to be tackled successfully.

The purpose of this paper is to compare the performance of four heuristic search algorithms when applied to molecular docking. These four are simulated annealing (SA), a genetic algorithm (GA), evolutionary programming (EP) and tabu search (TS). While SA, GA and EP have each been applied to the docking problem in the existing literature, to our knowledge there has been no previous attempt to use the TS algorithm. This work thus serves to introduce this algorithm to the field and to compare it with algorithms already in use. The algorithms are compared, over a number of test cases, according to their ability to locate optima of an objective/energy function designed for use in the docking problem.

In developing a successful docking method, considerations regarding the energy function are at least as important as those pertaining to the search algorithm. The minimum value(s) of this function should correspond to the preferred binding mode(s) of the ligand, and the ultimate goal should be a correlation between the values of the function and the binding affinity of the ligand. For this work, since our aim is the comparison of search algorithms, we have studied a single energy function. This was chosen to be the function recently described by Gehlhaar et al. [14,15], because it is very fast to evaluate computationally, and because it has been demonstrated to be successful in a number of docking applications. The choice to compare algorithms using a single energy function simplifies our study considerably, and, since potential functions used in docking tend to share similar characteristics, there are good reasons to believe that conclusions drawn about algorithmic performance using this energy function will probably apply when the algorithms are used with different functions. A study comparing various docking energy functions in terms of their ability to predict binding mode and affinity would be of great interest and is planned for the future.

Comparison of heuristic algorithms is difficult for two main reasons. First, each category of algorithm covers many different possibilities for implementation, each of which will perform differently for a given optimisation problem. For instance, within the GA category, it is possible to implement a one-point crossover, a two-point crossover, or something more complicated, and there are many other possible sources of variation. Second, the performance of each algorithm depends on a set of adjustable operational parameters, and the quality of the results depends on the extent to which these are optimal for a given test case.

Our approach to the first of these problems has simply been to seek to implement all the algorithms without bias and with no preconceived idea as to which algorithm should be determined the 'best'. We have also sought to implement each of the algorithms in a fairly 'standard' manner. For instance, maintaining the GA example, our algorithm employs just a simple one-point crossover and random mutations. It is possible, indeed likely, that a more sophisticated GA would perform better than our simple one. However, the more sophisticated the algorithms, the more difficult the comparison, since sophistication brings with it more adjustable parameters to optimise and a difficult choice of which of the many possibilities for increased sophistication should be chosen. It is our belief that the best algorithm for the docking problem is probably a hybrid of various types of algorithm. It is hoped that by comparing fairly simple implementations of each algorithm, our study will point to desirable algorithmic characteristics for use in hybrid algorithms.

The second problem, that of optimising operational parameters (the so-called 'meta-optimisation' problem), is also not straightforward. It is almost impossible to guarantee that any given set of parameters chosen is truly 'optimal', particularly if the parameters are coupled in some way. In this work, a set of parameters for each algorithm was sought which performed well on all the test cases and extensive 'tuning' experiments were carried out to this end. Our experience has shown that it is important to tune parameters over a number of test cases because it is possible to over-optimise the performance on one test case at the expense of the others if only one example is used. Clearly, it is a very desirable characteristic of a docking algorithm that parameters be transferable between test cases, and algorithms for which this is not true will be penalised in this type of study.

Our comparison of algorithms is carried out over five test cases as specified in Table 1. This number was judged to be sufficient to allow general conclusions about algorithmic performance to be drawn, while retaining the possibility of a detailed discussion of each test case. The

TABLE 1
TEST CASES USED IN THE PRESENT STUDY

Enzyme	Ligand	No. of rotatable bonds	PDB code	Reference
Dihydrofolate reductase	Methotrexate	7	3DFR	56
Influenza virus neuraminidase	DANA	4	1NSD	57
HIV-1 protease	XK263	8	1HVR	58
Thrombin	NAPAP	6	1ETS	59
Thrombin	Argatroban	7	1ETR	59

The structures of the ligands and the definition of their rotatable bonds are given in Figs. 7a–e.

test cases were chosen to be relevant problems in CAMD, to be of varying difficulty as docking problems, and to reflect different aspects of molecular recognition. Some of the problems emphasise the formation of hydrogen bonds in molecular recognition, others emphasise steric fit to the active site, and others contain a more equal mixture of the two.

In addition to the comparison of algorithms, a method is described whereby the docking problem can be set up in a very general way, including rigid body and rotatable bond degrees of freedom in the ligand, rotatable bonds in the active site of the receptor, and the variable presence or absence of crystallographic water molecules. In this paper, however, only the ligand degrees of freedom are used, the others being left for a later publication. It is also illustrated how, within this method, each of the search algorithms can be implemented with minimal effort, and with much important code in common. The resulting software will be referred to as PRO_LEADS (ligand evaluation by automatic docking studies) and forms part of our Prometheus system for molecular design and simulation.

Methods

Problem representation

Degrees of freedom

In order for solution of the docking problem, as stated in the Introduction section, to be feasible using currently available methods and computational resources, it is necessary to make certain simplifications. First, neither receptor nor ligand can be considered to be fully flexible. The receptor must be considered rigid except perhaps for some limited flexibility in the active site. Since the ligand is typically much smaller than the receptor, more flexibility can be considered, although most methods only allow ligand flexibility through rotations about rotatable bonds. Second, an active site for the receptor must be defined in order to restrict the region of space in which solutions are sought. Only with these simplifications is the solution space of the problem sufficiently small that a good heuristic algorithm could be expected to find optimal solutions with a reasonable success rate.

The docking methods implemented in PRO_LEADS allow the following degrees of freedom:

(1) Ligand translation – the ligand is free to move within a user-specified box defining the active site; if the centroid of the ligand moves outside the box, a user-specified penalty is added to the energy value.

(2) Ligand orientation – the ligand has full orientational freedom.

(3) Ligand flexibility – the ligand is considered flexible through a list of rotatable bonds. The rotatable bonds can be specified by the user or assigned automatically.

(4) Receptor active site residues – these can be considered flexible through their rotatable bonds. The user can choose the degree of flexibility. The available options are all rotatable bonds in the side chain or just the terminal rotatable bond.

(5) Crystallographic water molecules – if water molecules are present in the X-ray structure of the receptor these can be defined to be 'variable'. In this case the docking algorithms search for solutions in which the water molecules may be either present or absent.

In this article we explain how our approach deals with all the above degrees of freedom. For the results presented, only the first three degrees of freedom on the above list were considered variable, investigation of the others being left until a later publication.

Docking variables

The ICM tree [24] provides a complete internal coordinate description of an assembly of molecules, their internal conformations and relative positions and orientations. This makes it an ideal basis for the choice of the variables for use in docking and thus a very similar scheme has been implemented in this work. In PRO_LEADS, the docking variables representing the relative position of ligand and receptor and their internal conformations are a subset of the variables from the internal coordinate tree. This is illustrated in Fig. 1. The receptor is considered fixed in space, and the relative positions of the receptor and ligand are governed by the rigid body variables attached to the ligand. With the notation of Fig. 1 these are {B1,V1,T1,V2,T2,T3}, and can be interpreted as bond lengths, and valence and torsion angles with respect to the fixed triplet of virtual atoms at the root of the tree. Variables representing flexibility of ligand and receptor are torsion angles taken from the internal coordinate tree.

The docking variables, as manipulated by docking algorithms, are stored as a string of real numbers. The first six are the rigid body variables and after these follow the variables for ligand and receptor flexibility. At the end of the variable string are stored variables controlling the presence or absence of variable crystallographic waters using a method similar to that suggested independently by Read et al. [25].

Most of the energy functions for use in docking require Cartesian coordinates for the ligand and receptor molecules. It is necessary therefore to carry out interconversions between the docking variable string and Cartesian coordinates. This is accomplished by a single software module of PRO_LEADS which is called the coding module. The interface of this module to the outside world comprises three routines: InitialCode, Code and Decode. InitialCode is called once for each ligand/receptor system; this sets up an initial variable string consistent with Cartesian coordinates of the molecules and user options related to flexibility. It also sets up private module data

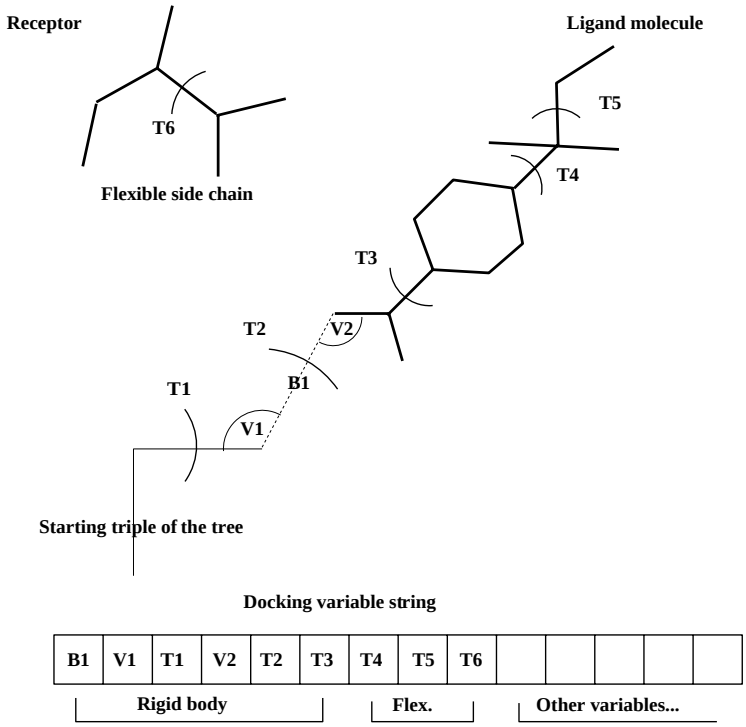


Fig. 1. Derivation of the docking variable string from variables present in the molecular internal coordinate tree. B denotes a bond length, V a valence angle and T a torsion angle. The variable string is composed of real numbers and only the derivation of conformational variables is illustrated. Variables denoting the presence or absence of crystallographic waters are appended to the string after the conformational variables, and other types of variable may also be added.

containing the mapping of the docking variables into the receptor and ligand structures and information about the types of each variable. After the call to InitialCode, calls to Code and Decode convert ligand and receptor Cartesian coordinates into docking variables and vice versa.

Within this design, implementation of most docking algorithms is straightforward. The algorithm manipulates the string of docking variables, simply calling Decode prior to any energy evaluation. The fact that each algorithm is able to share the coding facility has allowed us to implement a variety of algorithms quickly. Similarly, the energy evaluation called by each algorithm is performed by the same routine, which at the outer level is simply a switch over user options. This allows the addition of new energy functions with minimal changes to existing code.

It is worth noting at this stage that the efficiency of an algorithm often depends upon some knowledge of the nature of each docking variable. For instance, the distance moved by the ligand for a change in V1 is proportional to B1, and for a change in T1 it is proportional to (B1)sin(V1). When an algorithm changes these variables, such effects should be accounted for, in this case by using an appropriate scaling factor. A related point is that the software chooses an atom close to the centroid of the molecule as the first real atom in the tree. If an atom near one end of a long ligand were chosen, small changes in some rigid body variables would result in very large movements at the other end of the molecule.

Energy function

The energy function chosen for this work is that due to Gehlhaar et al. [14,15]. This function, which is specifically designed for fast docking applications, comprises four terms: the non-bonded interaction between ligand and receptor, a ligand internal energy associated with torsion angles, a term penalising internal clashes within the ligand, and a term penalising solutions which lie outside a box defining the active site.

The non-bonded interaction term is a pairwise sum over ligand and protein heavy atoms, each term taking the piecewise linear form shown in Fig. 2. The function has six parameters {A–F}, and the force field employs two sets, one representing a hydrogen bond (HB) interaction {2.3,2.6,3.1,3.4,–2.0,20.0} and the other a steric (S) interaction {3.4,3.6,4.5,5.5,–0.4,20.0}. Atom typing assigns to each heavy atom a type which is either hydrogen bond acceptor (nitrogen and oxygen with no attached hydrogens), hydrogen bond donor (primary and secondary amines, amides), hydrogen bond donor and acceptor (hydroxyl groups and crystallographic waters), or nonpolar (carbon and sulphur). The type of interaction between each atom pair is given in Table 2. For computational speed, the non-bonded interaction term is stored on four grids covering the active site. The extent of the grid is calculated by adding a proportion of the radius of the molecule to each dimension of a bounding box encom-

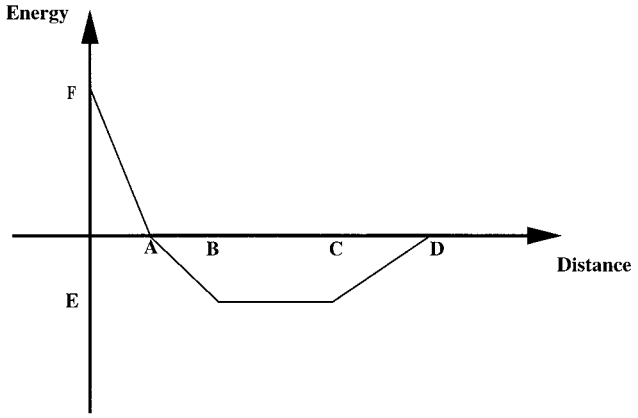


Fig. 2. The non-bonded interaction function of the docking energy function of Gehlhaar et al. [14].

passing the molecule. In general, it has been found that adding half the radius creates a grid of sufficient size to encompass the movement of the molecule during docking while remaining within reasonable memory limits. In all experiments reported in this paper, a grid resolution of 0.2 Å was used. An additional option available with this energy function is to scale down repulsive components in the interaction energy by a user-specified factor; many docking algorithms make use of this option in order to facilitate searching (by allowing the ligand to penetrate the receptor) in the early stages of a docking run, but it was not used in any of the examples quoted in this paper. This was because for two of the algorithms (TS and RS), it would have been difficult to implement the scaling in a meaningful way. However, it is likely that, for certain cases, such scaling may prove beneficial to the docking process. Recently, Verkhivker et al. [26] have suggested that better searching characteristics are obtained when the F value of the energy function is set equal to 4.0; however, all the results in this paper were generated using the function in its original formulation.

The internal energy of the ligand is the sum of torsional and internal clash terms. The latter term is a penalty of 10 000 when the distance between two non-bonded atoms becomes less than 2.35 Å. The former term has the form

$$E = A(1 - \cos(n\phi - \phi_0)) \quad (1)$$

where ϕ is the torsion angle, for sp^3–sp^3 bonds $A = 3.0$, $n = 3$, $\phi_0 = \pi$, and for sp^3–sp^2 bonds $A = 1.5$, $n = 6$, $\phi_0 = 0$. Other types of bonds cannot be considered rotatable.

The final term in the energy function is a penalty for leaving the box defining the active site. Two options are available: the first attaches the penalty to solutions with the centroid of the ligand outside the box, and the second attaches the penalty to each ligand atom which falls outside the box. In this work, the former option has been preferred because it constrains position while allowing full orientational freedom.

Search algorithms

The different search algorithms used are briefly described below and Figs. 3–6 give the schemes used for SA, EP, TS and GA, respectively. In the case of tabu search, more description is given since this is a novel approach to the docking problem. For each search algorithm, the parameters have been chosen so that the total number of energy evaluations per docking run was approximately 200 000.

Initial conformations

For the purposes of the test cases used in this paper, the starting position and orientation of the ligand were randomised within the box defining the active site. All torsion angle docking variables were also randomised. Some algorithms require only one such starting position (SA and TS), others (the GA and EP) require a population of them. The software also provides the option of a user-specified starting conformation.

Simulated annealing

The simulated annealing algorithm [27,28] follows the scheme illustrated in Fig. 3.

Within our implementation, the perturbations required to generate new solutions are random numbers drawn from either the Gaussian or Cauchy distribution, at the choice of the user. The use of the Cauchy distribution was motivated by the fast simulated annealing algorithm of Szu and Hartley [29]. Both options were tried for a number of test cases, and, since no particular advantage was found for the Cauchy distribution, the Gaussian distribution was used for the examples cited in this paper. Perturbations to angular variables were forced to lie in an appropriate domain ($[-\pi,\pi]$ for torsion angles and $[0,\pi]$ for valence angles) by translating any out-of-domain values through multiples of the domain size. The size of the perturbations generated depends on the width of the generation distribution (standard deviation for the Gaussian, semi-interquartile range for the Cauchy). Within the algorithm this is set as a proportion of the size of the allowed domain for each variable (e.g. a proportion of 2π for torsion angles in the ligand), and this proportion is the same for any variable. For the variables B1, T1 and

TABLE 2
PAIRWISE INTERACTIONS IN THE DOCKING POTENTIAL FUNCTION

	Donor	Acceptor	Both	Non-polar
Donor	S	HB	HB	S
Acceptor	HB	S	HB	S
Both	HB	HB	HB	S
Non-polar	S	S	S	S

HB: hydrogen bond interaction; S: steric interaction. Protein atom types are listed horizontally and ligand atom types vertically.

1. **Generate starting point** either the same as the input solution or by randomising the docking variables

2. Begin **loop over** the number of **temperatures**
   (a) Set current solution to the best from previous temperature
   (b) Scale down width of generation distribution according to temperature if option is chosen
   (c) Begin **loop over** the number of **trials**
      (i) Generate new solution by **perturb**ing current **solution**
      (ii) Evaluate energy of new solution
      (iii) Decide whether or not to **accept** the new **solution using Metropolis criterion**
      (iv) If accepted, set current solution to new solution
      (v) Update best solution at this temperature if necessary
   (d) Update best solution overall if necessary

3. **Output best** solution found

Fig. 3. Simulated annealing algorithm.

V1, the width was set with respect to a maximum-allowed translation equal to the length of the longest side of the bounding box, rather than the size of the allowed domain, using scaling for the angular variables as described in the Docking variables section. The initial width of the generation distribution is a user input (a value of 0.05 was used for the examples cited in this paper). An option exists to scale this width down linearly with temperature, resulting in smaller perturbations being used at lower temperatures; however, this was found to provide no significant advantage, and so was not used.

The algorithm is driven by a user-specified cooling schedule comprising a set of monotonically decreasing temperatures and a number of trials at each temperature. The effectiveness of the algorithm in optimising the system energy is strongly dependent on the cooling schedule. The examples cited in this paper all used the same cooling schedule. This was $\{T_1 = 40\,000;\ T_{i+1} = 0.8801T_i,\ i = 2,...,20\}$, with 10 000 trials at each temperature. The value of the Boltzmann constant used was 0.01986 kcal/(mol K). The temperatures could only be considered 'real' if the unit of the energy function were kcal/mol, which is not the case for the energy function used in this paper.

Evolutionary programming

The EP algorithm follows the framework given in Fig. 4.

Each individual in the population is represented by a pair of real-valued vectors. One vector stores the docking variables described in the earlier section and the other holds parameters guiding self-adaptive mutation (vide infra). In EP, offspring are created from parents by mutation. Traditionally, a mutation operator based upon Gaussian random numbers has been used, but recent work by Yao and Liu [31] suggests that more rapid convergence can be obtained by using Cauchy random numbers instead. Furthermore, good results have been found using self-adaptive mutation parameters [32], which allows the mutation to mould itself to the search as it proceeds [30]. In PRO_LEADS, self-adaptive mutation is always used and Cauchy random numbers have been investigated as an alternative to the traditional Gaussian operator.

Following Saravanan et al. [32], self-adaptive mutation of a parent (x,σ) to an offspring (x',σ') can be formulated thus:

$$\sigma_i' = \sigma_i \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (2)$$

and

$$x_i' = x_i + \sigma_i' N_i(0,1) \quad (3)$$

where x is the vector of docking variables and σ is the associated vector of mutation parameters. N(0,1) is a normally distributed random number with a mean of 0 and a standard deviation of 1. The parameters τ and τ' are commonly set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$, respectively, where n is the length of vector x.

For fast evolutionary programming, Yao and Liu [31] suggest that Eq. 3 be replaced by

$$x_i' = x_i + \sigma_i' \delta_i \quad (4)$$

1. **Create** an **initial population** of solutions

2. Evaluate the fitness of each population member using the energy function

3. **Create offspring** from all parents without selection **using mutation** operator

4. **Evaluate fitness** of offspring

5. **Loop over** all **offspring** for tournament selection
   (a) Randomly **choose** N other offspring as **opponents for tournament**
   (b) Score a win each time the chosen offspring is more fit than its opponent
   (c) **Rank** this offspring **by number of wins** in its tournament

6. **Select top-ranking** solutions as new population

7. If user defined **number of generations** is **exceeded**, **stop**. Else **goto 3**

Fig. 4. Evolutionary programming algorithm.

where δ_i is a Cauchy random number variable. Following these authors, the Gaussian perturbation of the mutation parameters has been maintained. It may be that there are better schemes for use with Cauchy mutation; this is being investigated [31].

For all the EP experiments described in this paper, a population size of 2000 individuals was used and the evolutionary search took place over 50 generations. In each generation, every parent gave rise to two children and the number of competitors in the selection tournaments was set to five. Initial tests showed that using Cauchy rather than Gaussian random numbers for mutation gave superior results and so all EP runs used the former distribution. The initial value of the mutation parameter σ was set at 0.075 for all the docking variables except for two of the rigid body variables, for which it was scaled as described in the Docking variables section.

Tabu search

The modern form of tabu (or taboo) search is due to Glover [33,34] and was originally applied to problems in the field of operations research. More recently, however, tabu search has begun to attract attention as an effective heuristic search procedure for combinatorial optimisation problems in molecular design, such as the evaluation of the chemical distance between two molecules [35]. Other workers in the molecular design field have employed related concepts [36–38], but, to our knowledge, this paper reports the first application of tabu search to the docking problem.

As its name suggests, tabu search is concerned with imposing restrictions to enable a search process to negotiate otherwise difficult regions [33]. These restrictions take the form primarily of a *tabu list* which stores a number of previously visited solutions or regions of space. By preventing the search from revisiting these regions (except under special conditions, vide infra), the exploration of the search space can be encouraged. Our implementation of tabu search for molecular docking is presented in Fig. 5.

Tabu search maintains only one current solution during the course of a search and the initial solution is chosen (vide supra) at the start of the run. From this current solution, a user-defined number of 'moves' is generated by a mutation-like procedure in which Gaussian or Cauchy random variables are added to each of the docking variables in the current solution. Each of these moves is then scored using the energy function and they are then ranked in order, with the best move at the head of the list. The moves are examined in rank order. Moves are considered 'tabu' if they generate solutions which are not sufficiently different from those solutions in the tabu list. The threshold measure used in this work to determine the tabu status or otherwise of potential moves is a root-mean square (rms) (measured over heavy atoms) of 0.75 Å or less between the two solutions being compared. The highest ranking move (tabu or not) is always accepted if its energy is lower than the lowest energy so far. Otherwise the algorithm chooses the best non-tabu move. If neither of these criteria can be met, the algorithm terminates.

If a new current solution can be found, it is added to the tabu list. Early in the search, solutions are simply added to the end of the list until it is full. Thereafter, the current solution must replace an existing solution stored in the tabu list. In PRO_LEADS, the tabu list is managed in a 'first-in, first-out' (FIFO) manner with the current solution replacing the tabu solution having the longest residence in the list. We have also experimented with an energy-based updating criterion in which the current solution replaces the solution of lowest energy in the tabu list, but tests have shown that it offers no particular advantage over the traditional FIFO updating procedure.

Once the new current solution has been identified and stored, a new set of moves is generated from it and the search procedure continues with the next iteration. A further mechanism which helps search exploration has also been implemented: if, after a number of iterations of the above procedure, it is observed that the best solution has not changed, then the tabu search is randomly restarted at a new position in the search space. While this

1. **Create initial solution** as specified or at random. Make this the current solution
2. **Evaluate current solution. If** the current solution is the **best** so far, **record** it
3. **Update Tabu list**
   (a) If tabu list is not full, add current solution to list
   (b) Else, replace oldest member of list with current solution
4. **Generate** and evaluate **N possible moves** from the current solution
5. **Rank** N **possible moves** in ascending order of energy
6. **Examine the moves in rank order**
   (a) If move has lower energy than **best so far, accept** it and go to 7
   (b) If move is **not Tabu**, **accept** it and go to 7
   (c) If **no acceptable moves** are located, **terminate** algorithm
7. If the **iteration limit** has been reached, **exit** with the best solution found. If the best solution so far has not changed for a given number of iterations, restart the whole procedure (go to 1). Otherwise, go to 2

Fig. 5. Tabu search algorithm.

1. **Generate** an **initial population** of solutions

2. Begin **loop over** number of **genetic operations**
   (a) **Calculate** the **fitness of** each solution in the **population**
   (b) If population has converged or maximum number of genetic operations has been exceeded, finish
   (c) **Select** two **parent** solutions **by roulette wheel** procedure
   (d) **If (Crossover)**, produce two children by one point **crossover**
      (i) Choose random position in docking variables
      (ii) Divide parents at this point
      (iii) Obtain children by taking combining first piece of one parent with second piece of other parent
      Otherwise, **copy** parents to children
   (e) Loop over children and **if (Mutate) apply random mutation**
      (i) Choose docking variable at random
      (ii) Add random number from Gaussian distribution to docking variable. Width of distribution is 0.1 of the domain size for the variable
   (f) **Replace least fit** population member if child's energy is lower

3. Go to 2(a)

Fig. 6. Genetic algorithm (the conditions crossover, mutate and accept are explained in the text).

multiple restart procedure is not part of the classical tabu search, it has been shown to help the search escape from local minima in our studies. The tabu search continues for a user-defined number of iterations. At the end of this time, it terminates and returns the best solution found during the search.

In all the tabu search experiments described in this paper, the search was allowed to proceed for 2000 iterations. At each iteration, 100 moves were generated using Cauchy mutation with a fixed σ value of 0.075. The length of the tabu list was 25 and the random restart was initiated if the best solution had not changed after 100 iterations.

Genetic algorithm

A detailed account of genetic algorithms can be found in Goldberg [39]. The general framework for our genetic algorithm is illustrated in Fig. 6. The algorithm is implemented in 'steady-state' form, i.e. the same population of solutions is continually updated, and there is no concept of a generation.

Although many genetic algorithms are implemented with a binary encoding (i.e. the variables are encoded in a bit string), it was decided to allow the genetic operators to act directly upon the string of real docking variables described in the Docking variables section. The algorithm makes use of two operators, *crossover* and *mutation*. Crossover acts upon two *parent* solutions and produces two new solutions called *children*. The mutation operator acts on one solution. The probability of the operations occurring is controlled by the user. For the examples in this paper, the crossover probability is 0.5 and the mutation probability is 0.5. However, in PRO_LEADS, mutation will always occur if there has been no crossover so as to increase the genetic diversity of the population.

Selection of parents at each step follows the roulette wheel method [39]. Each population member is assigned a raw *fitness* value, F_r, which is given by the difference in energy between the solution energy and the solution of maximum energy within the population. This raw value is then scaled linearly, $F = aF_r + b$, so that the average fitness is preserved and the maximum fitness is MaxScaleParam times greater than the average. If this scheme ever results in negative scaled fitness values, the latter criterion is dropped and the lowest fitness is set to zero. When the fitness values have been calculated, each solution is assigned a section of a roulette wheel of size proportional to its fitness, and this wheel is spun to select parents. The point of scaling the fitness values is to vary the selection pressure used by the algorithm. With a large value of MaxScaleParam, selection pressure is very strong and the fittest individuals have a very high probability of selection. Since this tends to lead to low genetic diversity of the population and subsequent trapping in a local minimum, moderate values of MaxScaleParam are usually used (1.2 is used for the applications in this paper).

Random search

The random search (RS) procedure simply generates a random conformation and orientation of the ligand subject to the constraint that the ligand's centre of mass (COM) lies within the bounding box specified by the user. In each of the RS docking runs, this was repeated 200 000 times and the algorithm terminated returning the lowest energy solution found during the search.

Local minimisation

A local minimiser which uses the Powell algorithm [40] has been implemented in PRO_LEADS. This is a non-derivative algorithm designed to move the solution to the nearest local minimum. It can be used optionally as a final stage minimisation of the lowest energy conformation found after the operation of any of the search algorithms.

Basis of comparison between algorithms

In this work, the primary concern is with examining the relative performance of the various algorithms. All

the heuristic algorithms contain a stochastic element, and so produce different results depending on the starting value of a random number seed. It is necessary therefore to assess performance statistically over a sufficiently large number of independent trials. To ensure a fair comparison between algorithms, each one was limited to a maximum of 200 000 (±1%) function evaluations per docking. This number was chosen to be large enough for most algorithms to achieve a reasonable success rate, while leading to a CPU time requirement short enough to permit many independent trials to be made.

The first and most straightforward criteria we use in the comparison of the algorithms are the characteristics of the energy distribution of the results, generated from the above trials. These characteristics are the average energy of a solution, and the width of the distribution around this average value. For a simple case, in which the energy surface has a single minimum which is much deeper than any other minimum, an ideal algorithm would be expected to produce an average energy close to the value of this minimum with a narrow distribution of results around this value, reflecting the fact that most trials lead to this deep minimum being located.

The energy surface is rarely as simple as the ideal case outlined above, and frequently there are a number of deep minima of very similar energy ('competing minima'). In such a case, while the characteristics of the energy distribution are still useful quantities, they do not always reveal all the differences between the algorithms which are present in the results and can sometimes even be misleading. When there are competing minima, it is useful to classify solutions produced by the algorithms according to the minimum to which they correspond, and to study how solutions are distributed amongst the various minima for each of the algorithms. A useful quantity to aid this analysis is the rms distance of the docked ligand conformation from the conformation it adopts in the crystal structure (i.e. the distance from the 'correct' answer). A scatter plot of rms against energy for all the solutions produced usually reveals a number of clusters of solutions, each cluster corresponding to a given binding mode of the ligand, and identified with a single broad minimum in the energy function. An examination of such scatter plots often reveals interesting algorithmic characteristics which would not be apparent from a study of the energy distribution alone, as will become clear in the Results section.

For many cases, using the energy function chosen for this study, we find that the deepest minimum located by any of the algorithms is one corresponding to the crystal structure. With this in mind, we also compare algorithms according to their 'success rate', that is, following Gehlhaar et al. [14,15] the proportion of the trials which find a solution within 1.5 Å rms (heavy atoms only) of the crystallographic ligand conformation.

A preliminary test was carried out in order to decide on appropriate statistical methods with which to assess the results. This revealed that, in general, the distribution of results deviates significantly from the normal distribution, as might have been predicted from the expected form of the energy surface. With this in mind, the median and semi-interquartile range were preferred as descriptive statistics to the more common mean and standard deviation. When comparing the heuristic algorithms, the main quantity considered was the median energy of the distribution of best energies obtained over 500 independent trials. This number of trials was chosen because it was found to provide a very good estimate of the median energy and a good estimate of the more variable semi-interquartile range. The minimum energy solution found over the 500 trials is also reported for each of the heuristic algorithms. Note that because the RS procedure is only intended as a control, statistics for this algorithm were gathered over 100 independent trials. In view of the deviations from the normal distribution, a non-parametric method was chosen to assess the statistical significance of these comparisons. The method of Gardner and Altman [41] was used to compute a 95% confidence interval for the difference between the two medians, and a significant difference between two algorithms was supposed to exist if zero fell outside this interval. As with all significance tests, this simply tells us if an observed difference may be considered 'real', i.e. unlikely to have occurred by chance. Of course, it is possible for small differences to be statistically significant and yet be of no practical significance.

In general, the local minimiser is used to refine the best solution at the end of each docking run. However, for one of the test cases (1HVR), it was decided to examine the relative performances of the algorithms both with and without this final refinement. This enables an assessment of the benefit to each algorithm of the final stage of local minimisation.

Test cases

The test cases chosen for the paper are given in Table 1. All are topical test cases in CAMD, for which inhibitors of the associated enzyme are on the market or in clinical trials as therapeutic agents for important diseases. Dihydrofolate reductase–methotrexate (DHFR–MTX) was chosen because in recent years it has become a standard test case for docking algorithms. It thus serves as a useful benchmark for our results. Influenza virus neuraminidase–DANA was chosen as a test case in which electrostatic/hydrogen bond effects are thought to dominate recognition, and HIV-1 protease–XK263 was chosen because in this case lipophilic interactions are particularly important for good binding (steric fit). The two thrombin examples were chosen because the inhibitors make strong lipophilic and strong electrostatic interactions in different,

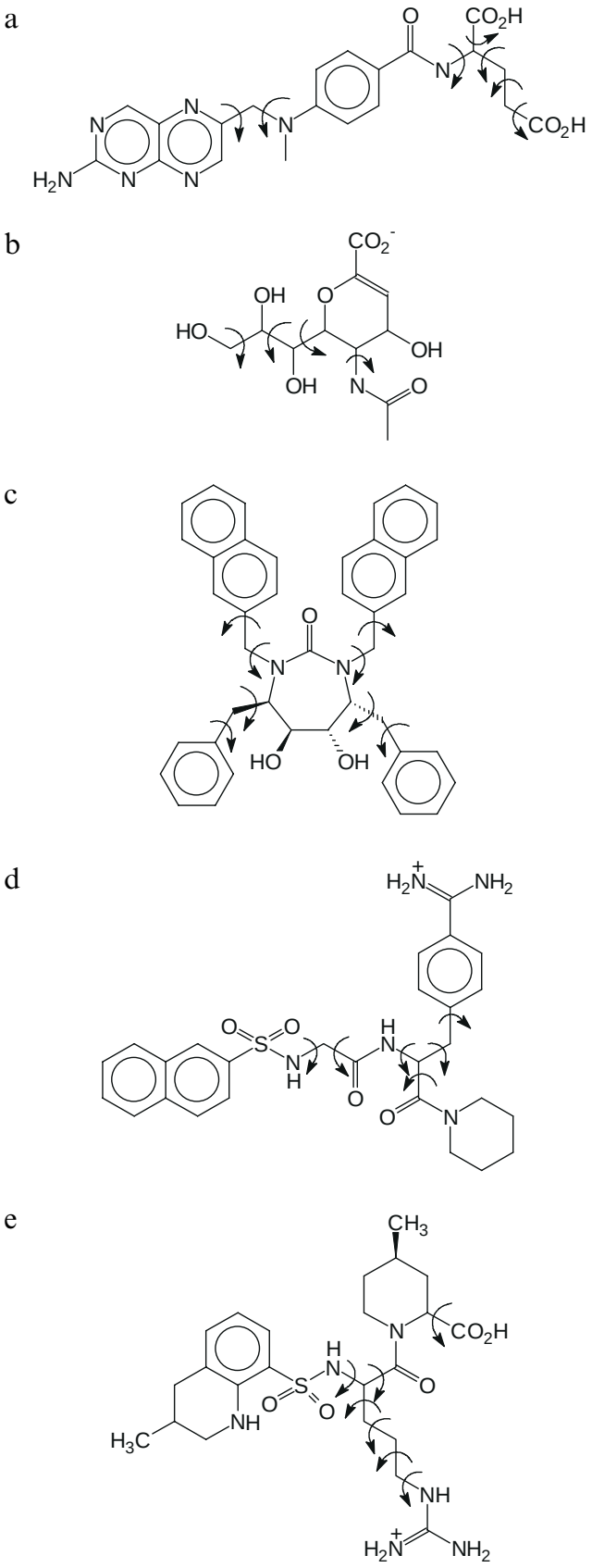


Fig. 7. Ligands used in docking studies in this paper with rotatable bonds indicated: (a) methotrexate; (b) DANA; (c) XK263; (d) NAPAP; (e) argatroban.

but similarly sized, binding pockets. Thus, the thrombin examples provide a good test of the ability of the docking potential function to differentiate between the two types of interaction.

All of the test cases were prepared in a similar fashion. The crystal structures were extracted from the Brookhaven Databank [42] and hydrogen atoms were added using the InsightII/Discover software [43]. The ligand structures were minimised prior to docking using the CVFF force field [43]. Since the potential function used in this work does not require accurate hydrogen atom positions, no minimisation of the receptor was performed.

In all cases, a bounding box defining the active site was specified by permitting the ligand's COM to move up to 2.0 Å along each axis from its crystallographic location. The total volume of the bounding box constraining the COM was thus 64 $Å^3$.

The treatment of crystallographic water molecules in docking and other molecular design studies is not a simple problem and has recently been the subject of a detailed computational study [44–46]. One approach is to remove crystallographic water molecules before attempting to dock the ligand; see for example Refs. 13, 16 and 18. These groups reported that they were still able to obtain correct docked conformations in their test cases, although other workers have found that removal of the crystallographic waters has necessitated the inclusion of a continuum solvation model before good results could be obtained [17,47]. In our test cases, we have found that the removal of crystallographic waters has a significantly deleterious effect upon the ability of the docking algorithm to locate the crystallographic binding mode for 1NSD, 1ETR and 3DFR. For instance, with a 'dry' active site for 3DFR, the best success rate observed was about 50%; on inclusion of all the water molecules, this rose to over 80%. Clearly, water molecules in the active site help to stabilise the crystallographic conformation by the imposition of additional steric and electrostatic constraints. In the light of these findings, and in view of the nature of the work in this paper which is primarily concerned with investigating algorithmic performance, we have retained all the crystallographic waters present in our test cases.

Results

Figure 7 shows the structures of the ligands used in the study together with the bonds which are considered rotatable. Table 3 gives the results for the different search algorithms on the test cases. The most obvious result is that, for all the algorithms, RS performs very poorly. The high median energy and low success rate produced by RS reflects the fact that RS is ineffectual in a search space of this size, and indicates the advantage of the more 'intelli-

TABLE 3
DOCKING RESULTS FOR THE TEST CASES GIVEN IN TABLE 1

PDB code	Algorithm	Minimum energy	Median energy	Semi-interquartile range	Success rate (%)
3DFR	SA	−163.75	−151.62	5.94	90
3DFR	EP	−164.61	−152.13	8.08	76
3DFR	TS	−164.69	−150.13	6.73	93
3DFR	GA	−167.64	−157.96	7.54	76
3DFR	RS	−137.07	−82.15	22.63	9
1NSD	SA	−103.88	−93.04	3.12	40
1NSD	EP	−105.60	−98.35	2.37	64
1NSD	TS	−104.71	−96.78	2.56	88
1NSD	GA	−105.43	−98.75	1.84	57
1NSD	RS	−92.75	−73.18	6.95	6
1HVR	SA	−177.31	−158.40	5.59	65
1HVR	EP	−175.24	−155.02	8.82	54
1HVR	TS	−176.33	−156.56	7.66	58
1HVR	GA	−176.52	−156.44	9.43	59
1HVR	RS	−154.86	−75.41	24.77	2
1HVR[a]	SA	−159.58	−137.40	5.05	61
1HVR[a]	EP	−168.55	−143.99	9.31	48
1HVR[a]	TS	−157.66	−135.74	7.19	60
1HVR[a]	GA	−173.29	−152.93	10.06	57
1HVR[a]	RS	−63.50	21.92	26.09	0
1ETS	SA	−138.01	−115.76	5.56	3
1ETS	EP	−139.76	−117.07	5.41	9
1ETS	TS	−139.67	−120.13	5.21	8
1ETS	GA	−144.00	−118.39	4.72	11
1ETS	RS	−112.68	−71.53	17.13	2
1ETR	SA	−138.46	−88.86	16.97	30
1ETR	EP	−140.52	−87.52	8.98	21
1ETR	TS	−138.68	−97.23	14.01	39
1ETR	GA	−140.85	−88.60	8.90	13
1ETR	RS	−101.45	−52.38	9.23	3

Energies are in arbitrary units and statistics are derived from 500 independent docking attempts for each algorithm, except RS (100 attempts). For comparison, the energies of the unrelaxed crystal conformations are −141.76 (3DFR), −100.04 (1NSD), −149.55 (1HVR), −132.48 (1ETS) and −98.89 (1ETR).

[a] These results were obtained *without* using local minimisation of the best solution.

gent' algorithms. The results for the other algorithms on the individual test cases are considered below.

Dihydrofolate reductase–methotrexate

The results in Table 3 show that the best performance for 3DFR in terms of median energy is produced by the GA. The differences in median energy between EP, SA and TS are not statistically significant. The success rate is not well correlated with the median energy; the most successful algorithm in terms of energy (the GA) is actually the joint worst of the four 'intelligent' algorithms in terms of success rate. This may indicate that the GA is more prone to becoming trapped in local energy minima, which represent conformations more than 1.5 Å rms from the crystal conformation, than SA or TS, both of which perform very well on this test case. It is worth noting that this test case was used in parameterisation of the energy function [14], and that therefore it might be expected that this function should have a single deep minimum near the crystal structure and yield good success rates. The scatter plot of energy versus rms from the crystal structure (Fig. 8) for the GA adds some weight to this hypothesis, showing that the majority of solutions found by the GA with energy less than −150 are indeed close to the crystal structure. Nonetheless, the plot clearly shows some solutions of low energy with rms values of more than 2.5 Å representing suboptimal minima on the energy surface.

Neuraminidase–DANA

The results for 1NSD show that the GA again performs best in terms of median energy, although the difference between EP and the GA is not statistically significant. The success rates are somewhat lower than those observed with DHFR–MTX, except for TS which still continues to perform very well. This points to a more complicated energy surface for this case, possibly with

more than one minimum of similar depth to that corresponding to the crystal structure. This suspicion is confirmed by the scatter plot shown in Fig. 9 in which it can be clearly seen that there is no simple linear relationship between energy and rms from the crystal structure. The figure shows that TS finds two dominant clusters of low-energy solutions, one close to the crystal structure having energies in the range −87 to −105 units, and the other with rms values in the range 4.5–5.0 Å with energies ranging from −88 to −99 units. The lower success rates can be attributed in part to the existence of this latter minimum in which each algorithm becomes entrapped in some proportion of its docking attempts. The results in this case seem to suggest that TS is less susceptible to this entrapment and, thus, may be carrying out a more effective global search than the other algorithms.

As an aside, it is interesting to consider the two observed binding modes in more detail. In the crystallographic conformation, the carboxylate group of DANA forms a salt bridge with Arg^{373} and also hydrogen bonds with Arg^{115} and Arg^{291} – clearly a very strong and specific interaction. The carbonyl of DANA's acetylamino group forms a hydrogen bond with Arg^{149} and the methyl moiety of the acetylamino group makes hydrophobic contact with Trp^{176} and Arg^{222}. The alternative binding mode discovered by the docking algorithms is almost inverted with respect to the crystallographic conformation. The salt bridge is not formed; instead, the carbonyl of DANA's acetylamino group forms a hydrogen bond with Arg^{115}. The majority of the remainder of the interactions in this mode are hydrophobic. It is our belief that this latter binding mode is in fact an artefact of the potential function, which seems to be biased in favour of steric fit and appears not to favour specific interactions like the carboxylate-arginine salt bridge sufficiently. The result is that the energy separation of the two minima is very small (about six units); this leads to a greater tendency for some of the algorithms to be trapped in the higher energy minimum than would exist if the separation were made greater by altering the potential function so that salt bridges were more highly rewarded.

HIV-1 protease–XK263

A point to note concerning this system is that the complex as deposited in the PDB contains no water molecules; the active site water molecule which mediates contact between peptidomimetic HIV-1 protease inhibitors and the enzyme is displaced by the carbonyl group of the urea which interacts directly with the active site residues Ile^{50} and $Ile^{50'}$.

SA performs best according to the median energy criterion, and the difference between it and the second-placed algorithm (the GA) is statistically significant although the differences in median energies between the GA, TS and EP are not. In this test case, in contrast to the two previous examples, a better correlation is observed between success rate and median energy, SA performing best and EP performing worst on both counts. The scatter plot shown in Fig. 10 for the results produced by SA shows this correlation well. It is also noticeable from this figure that a number of low-energy solutions lie

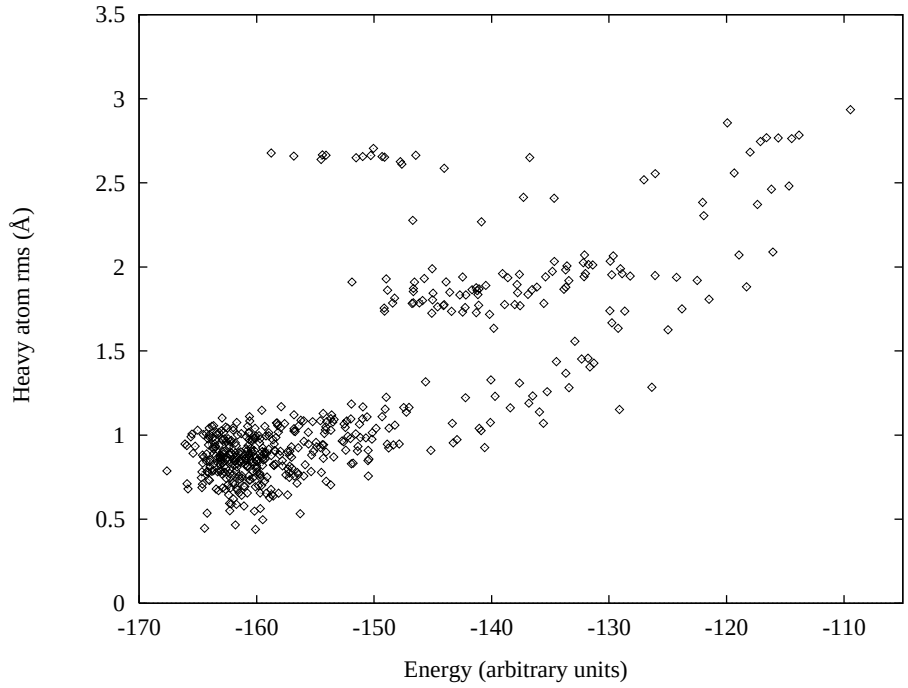


Fig. 8. Scatter plot of heavy atom rms versus energy for 500 docks of methotrexate into DHFR using the GA.

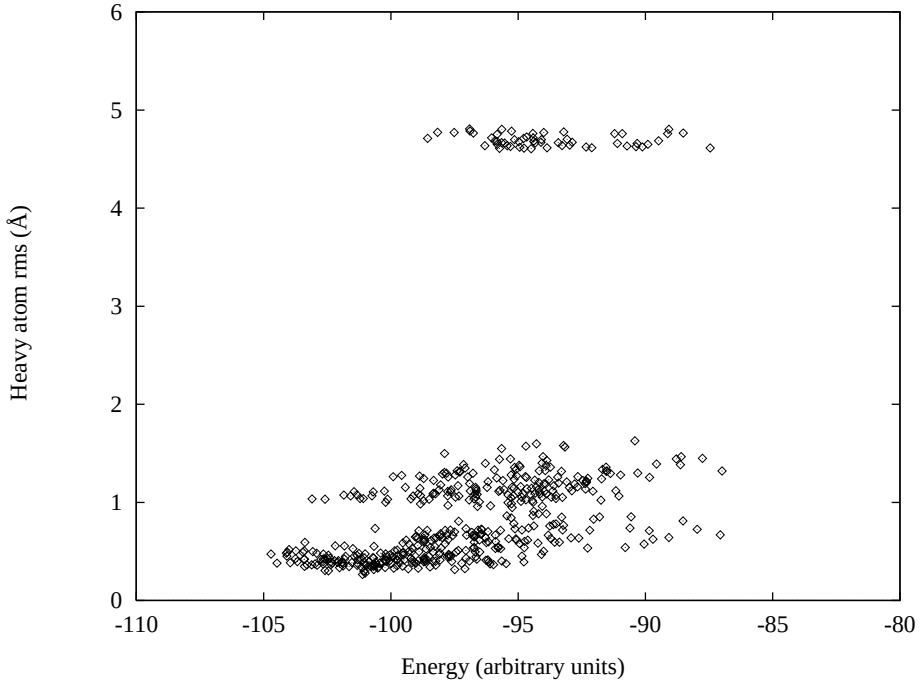


Fig. 9. Scatter plot of heavy atom rms versus energy for 500 docks of DANA into influenza virus neuraminidase using TS.

in the range 1.5–2.0 Å rms. This explains why the success rates for this test case are somewhat lower than those found in DHFR–MTX. The greater spread of good solutions around the crystallographic conformation reflects the less directional nature of binding of parts of this large molecule in an active site dominated by lipophilic contacts. For example, the exact orientation of the naphthyl and phenyl rings has greater effect on the calculated rms than on the value of the energy function.

The performance of the algorithms for 1HVR when the local minimiser is not employed is also given in Table 3. First, in terms of median energy, it can be seen that the results from each algorithm are improved by local minimisation, the size of the improvement varying as RS (97.53) > SA (21.00) > TS (20.82) > EP (11.03) > GA (3.51). This variation can be interpreted as an indication of the effectiveness of the local searching performed by the algorithms. Clearly, the population-based evolutionary algorithms are more effective as local searchers than our implementations of SA or TS. It should be noted that these improvements are both significant and computationally inexpensive (typically the local minimisation requires only a few hundred energy evaluations to be compared with the 200 000 permitted for the main algorithms), making the local minimiser a useful adjunct to the heuristic search algorithms. Second, it is interesting to note that in terms of success rate, the performance of the four algorithms is little affected by the absence of the local minimisation procedure. This again indicates that the crystallographic minimum is broad with a wide range of energies being possible within the 1.5 Å rms cutoff.

Thrombin–NAPAP

TS and the GA produce the lowest median energies for 1ETS, the difference between them not being statistically significant. In terms of successful docking, this example produces the lowest rates of our test set. However, these figures are somewhat misleading since the lowest energy solutions are not considered successful by our criterion, as will now be explained.

The scatter plot of energy versus rms for the TS results is shown in Fig. 11. The figure indicates that there are at least three major clusters of solutions produced by the algorithms. The first cluster is close to the crystallographic minimum, which can be characterised by the piperidine, naphthyl and benzamidine moieties of NAPAP interacting with (respectively) the lipophilic 'P' pocket, the lipophilic 'D' pocket and an aspartate residue at the bottom of the S1 subsite of thrombin. A second cluster of solutions occurs at about 3.5 Å rms from the crystal structure and contains some solutions of slightly lower energy than the first cluster. From the standpoint of minimising the global energy of our scoring function, this is probably the correct solution, and this explains why the tabu search does relatively poorly at docking NAPAP 'successfully'. Tabu search locates the second cluster 16% of the time, more often than all the other algorithms. (The corresponding percentage rates for the GA, EP and SA are 6%, 7% and 9%, respectively.) The second cluster is characterised by the 'incorrect' positioning of the naphthyl moiety, which points into solvent and makes little contribution to the score. This positioning is favourable

since the 'correct' positioning of the naphthyl incurs a significant, and probably unrealistic, torsional energy penalty of about 12 units with this energy function. In addition, the correct placement of the naphthyl causes the hydrophobic piperidine and naphthyl moieties to be in close contact; this favourable intramolecular interaction is given no weight in the scoring function. The third cluster of structures has a slightly higher energy than the first and illustrates another weakness of the scoring function similar to that seen with DANA–neuraminidase. Here, the naphthyl moiety is placed where benzamidine forms a salt bridge in the crystallographic configuration. Analysis of the energy components indicates that this naphthyl positioning contributes about −60 units of energy, which is very similar to the score from a correctly positioned benzamidine. It is tempting to speculate that the minimum corresponding to the third cluster is 'easier to find' than those corresponding to the first two clusters. A possible explanation for this is that correct positioning of the benzamidine moiety may involve much more specific molecular recognition than the incorrect positioning of the naphthyl group which is governed mainly by non-directional steric fit.

Thrombin–argatroban

For this test case there is some correlation between success rate and median energy, the best algorithm, TS, producing rather a better success rate than the rest. Once again, the success rates are relatively low when compared to, for example, DHFR–MTX or neuraminidase–DANA. The scatter plot of energy versus rms for the GA, shown in Fig. 12a, indicates that competing minima on the energy surface lie at the root of this problem. A cluster of solutions is observed corresponding to a minimum within 1.5 Å of the crystal structure with energies in the range −105 to −140, and another cluster of higher energy with rms in the range 4.5–7.5 Å. It seems that, despite the higher energy of this latter cluster, there is a significant possibility that the GA produces a solution from within it. This corresponds to the algorithm becoming trapped in a deep local minimum. The scatter plot for TS, shown in Fig. 12b, shows a similar effect, yet for this algorithm it seems that the probability of becoming trapped is rather lower. This suggests that TS is more effective at global searching, perhaps as a consequence of its random restart procedure. The SA and EP algorithms also exhibit a stronger tendency than TS to become trapped in the higher energy minimum, suggesting that this minimum may be 'wider', i.e. easier to find than that corresponding to the crystal conformation. It is possible that the 'narrowness' of the crystallographic minimum is caused by the very specific interactions that need to be made by the arginine moiety of argatroban in the S1 subsite of thrombin. The arginine scores poorly unless the geometry of these interactions is correct. It is interesting to note that this test case is easier for the docking algorithms than NAPAP partly because, in this instance, there is a trapped water molecule in the S1 subsite (its B factor is 12.8 and it has several hydrogen bonds to ligand and enzyme) which probably prevents a large hydrophobe from entering the pocket completely. Removal of the water molecule from this pocket has a markedly deleterious effect on the success rate of the docking.

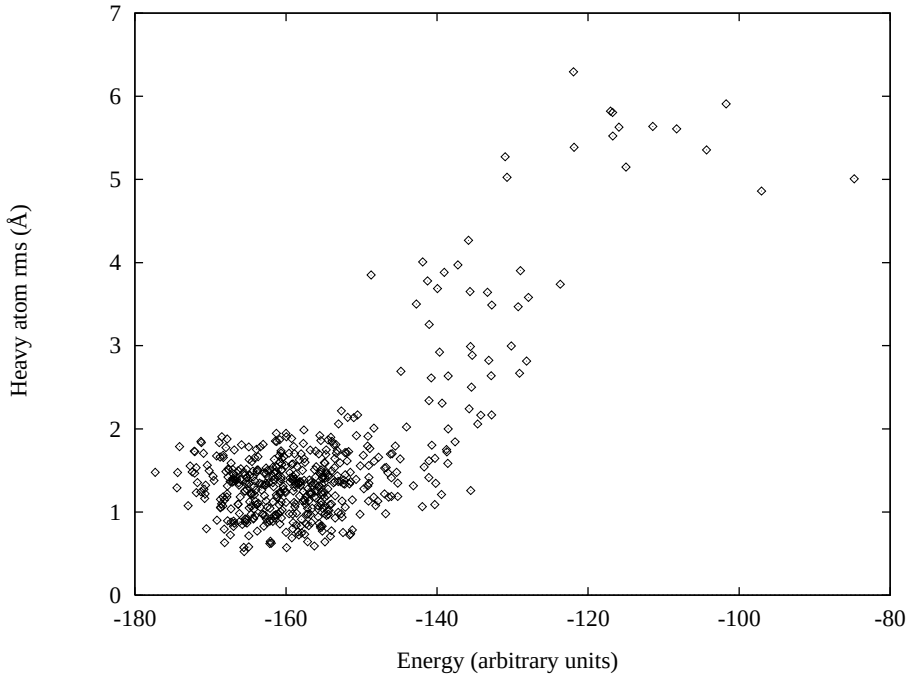


Fig. 10. Scatter plot of heavy atom rms versus energy for 500 docks of XK263 into HIV-1 protease using SA.

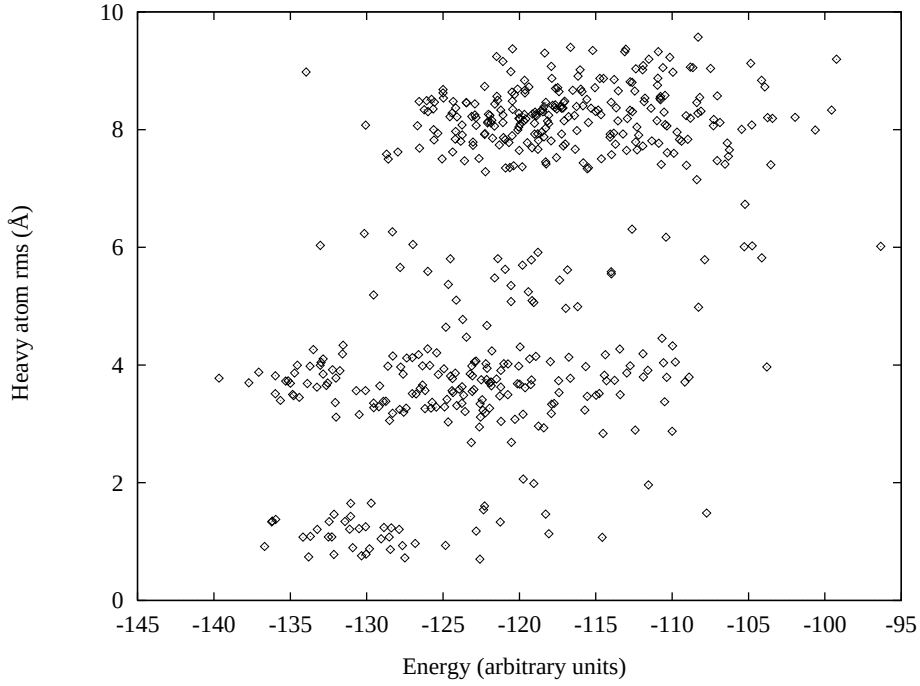


Fig. 11. Scatter plot of heavy atom rms versus energy for 500 docks of NAPAP into thrombin using TS.

Argatroban is a moderately difficult test case for the algorithms because, in spite of there being a good low-energy minimum, it is quite difficult to locate (with the number of function evaluations chosen for this study). For this reason, it was decided to rerun the jobs with all the algorithms increasing the number of rotatable bonds by one each time. Because of the large numbers of experiments involved in this study, only 100 docking attempts were used to derive the statistics. The order in which the rotatable bonds were activated is indicated in Fig. 13a. This order was chosen so as to approximately guarantee the best performance of the GA for a given number of rotatable bonds. The resulting median energies and success rates are shown in Figs. 13b and c. The introduction of the first five rotatable bonds can be seen to have little effect on the performance of the algorithms, except for the random search which is seriously compromised as additional rotatable bonds are activated. This indicates that although the number of rotatable bonds is a reasonable indicator of the size of the search space for the test case, it is a poor indicator of its difficulty. The difficulty is primarily controlled by the presence and character of the competing low-energy minima on the potential energy surface. The addition of bonds 6 and 7 obviously introduces and consolidates at least one more competing minimum. In general, the success rates follow the pattern TS > SA > EP > GA. The relative performance of the algorithms using the median energy criteria varies considerably as the test case changes from an easy test to a difficult one (i.e. on the introduction of competing low-energy minima). The GA does best in terms of median energy when the test case is easy, but not so well when the sixth and seventh rotatable bonds are introduced. We ascribe this to good local searching capabilities but somewhat poorer global searching. EP shows similar but less pronounced characteristics. TS is a poorer local searcher but produces the best median energies for seven rotatable bonds presumably because of its increased ability to sample the global energy minimum relative to the other algorithms. It could be argued that SA shows a similar though less clear-cut effect. The fact that the success rates and median energies at seven rotatable bonds are not exactly the same as the results for 500 runs given in Table 3 underlines the need to carry out large numbers of runs for comparisons of this type.

Discussion

The clearest conclusion that can be drawn from our results is that RS is always out-performed by the other more 'intelligent' algorithms. This was of course to be expected given the size of the search spaces involved; nonetheless, RS does provide a good 'control' for our results. Drawing conclusions about the other four algorithms is more difficult. An immediate observation that can be made is that all benefited, albeit to differing degrees, from hybridisation with the Powell local optimisation algorithm. Turning to the comparison of median energies, in three of the five test cases the GA was in (joint) first place in terms of its median energy and it was never worse than joint second. Thus, on the basis of this criterion, the GA may perhaps be judged the 'best' algo-

rithm. If points are awarded according to ranking by median energy (from 4 for first place to 1 for fourth place), then, based on overall performance across the five test cases, the algorithms may be ranked as follows: GA (18) > EP (15) > TS (14) > SA (12).

In terms of the docking 'success rate', the results are rather different. In the following analysis, the results from the NAPAP example are not included because, as discussed earlier, the global minimum on the energy surface is not judged as a 'success' using our criterion. Interestingly, the best result attained by the GA using this criterion is only second place behind SA (HVR-XK263). It is likely that this result reflects a predisposition on the part of our GA for entrapment in local minima more than 1.5 Å rms from the crystal conformation. Conversely, the success of TS, which produces the best success rate in the remaining three test cases, suggests that it is able to escape from such minima and locate the 'correct' minimum more often. This

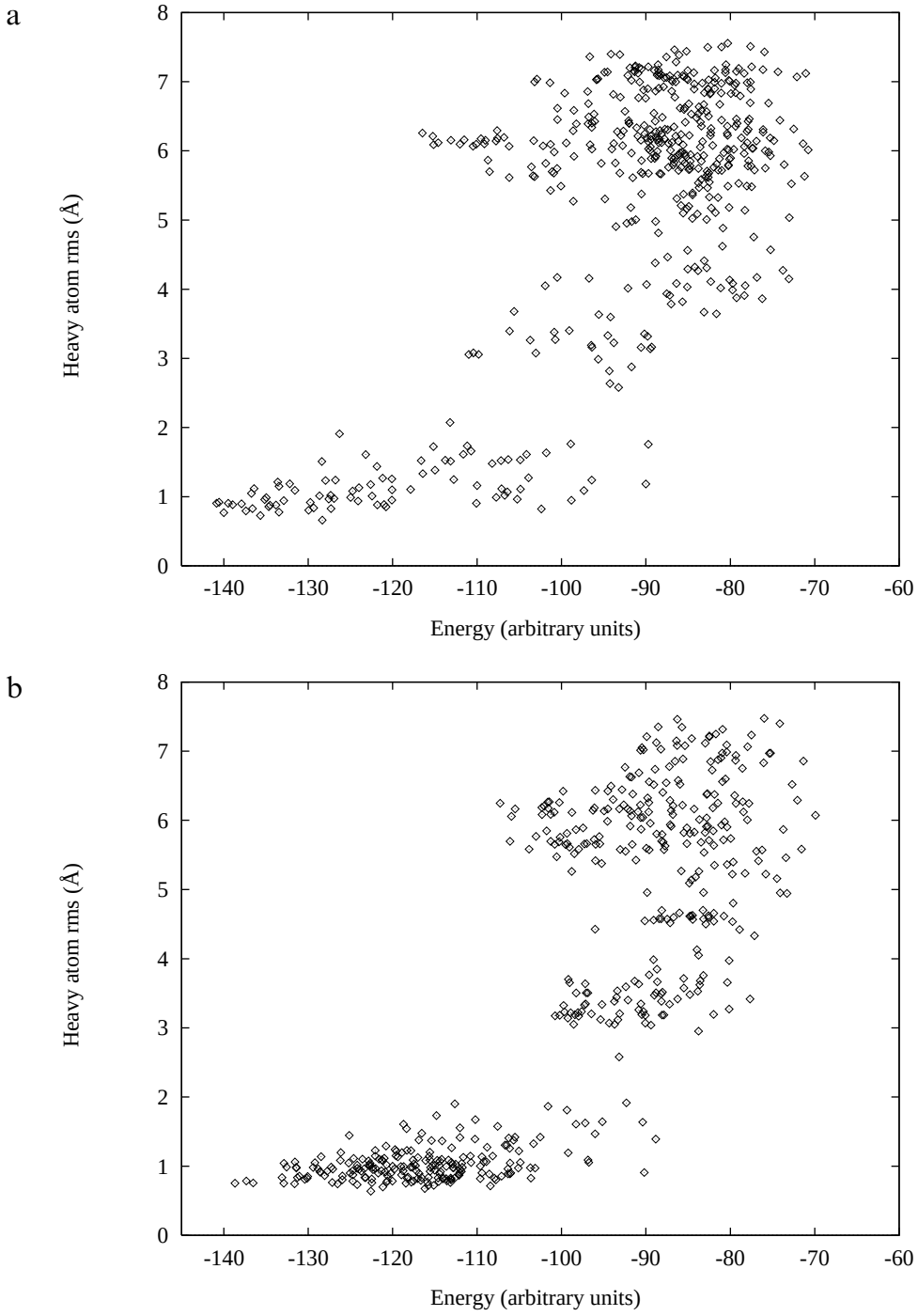
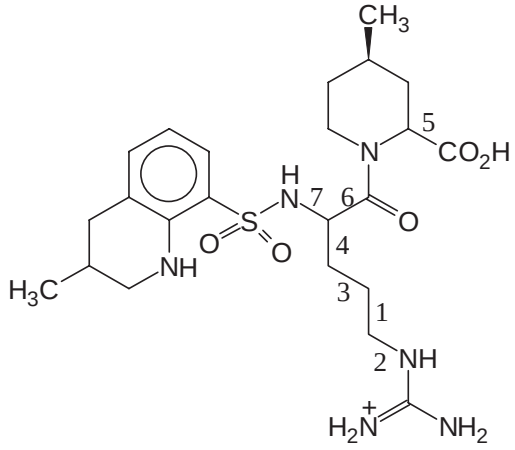
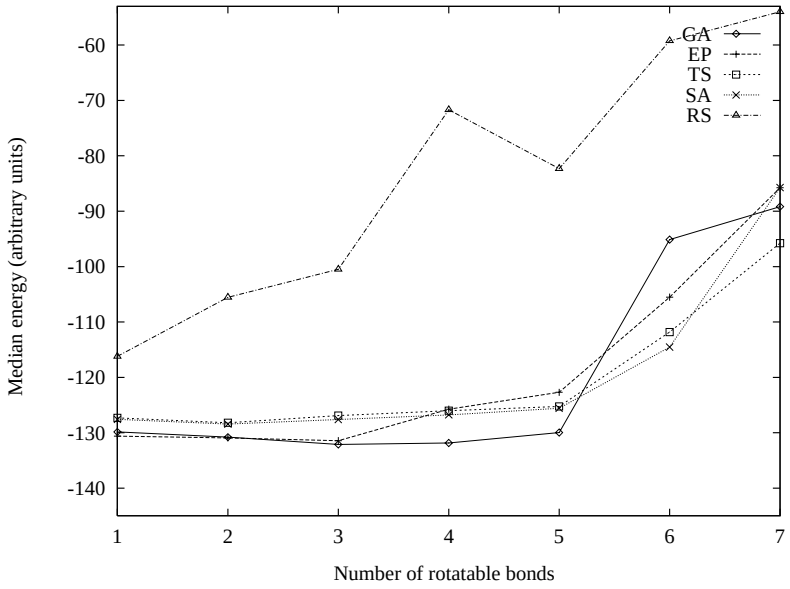


Fig. 12. Scatter plots of heavy atom rms versus energy for 500 docks of argatroban into thrombin using (a) the GA and (b) TS.

a



b



c

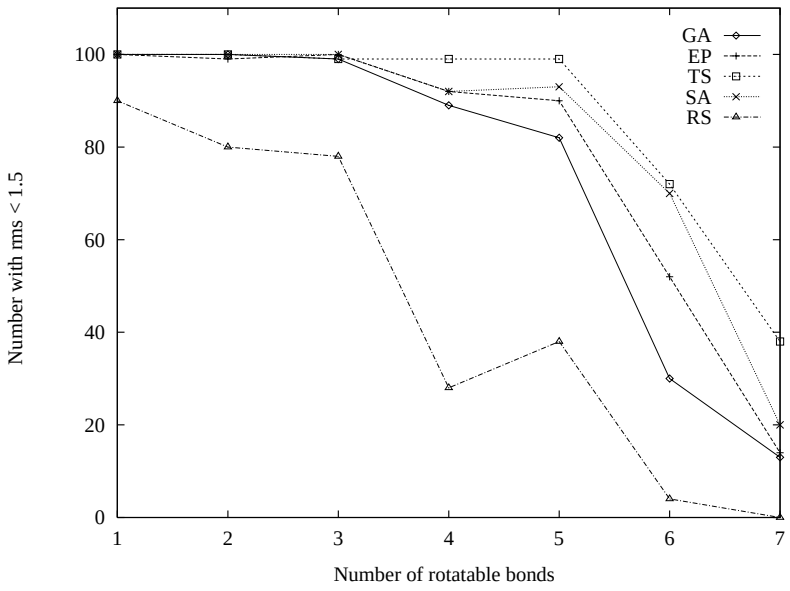


Fig. 13. (a) Argatroban showing the order in which the rotatable bonds are activated. (b) Effect on median energy of incrementally increasing the number of rotatable bonds in argatroban. (c) Effect on success rate of incrementally increasing the number of rotatable bonds in argatroban.

capability is probably a result of both the 'random restart' feature of our TS algorithm and the use of a high tabu threshold which drives the search into new areas of space. To summarise, using the same points scoring scheme as for the median energy and excluding the thrombin–NAPAP example, the ranking of the algorithms according to success rate is TS (14) > SA (11) > GA = EP (8).

In passing, it is also interesting to consider the relative efficiencies of the docking algorithms in terms of CPU time per docking attempt as measured on one node of a Convex Exemplar (HP 735 chip). The use of the grid-based energy evaluation means that the CPU time per energy evaluation varies in a linear fashion with the number of heavy atoms in the ligand. Thus, for TS the fastest docks are with DANA (223 s) and the slowest are with XK263 (535 s). Using this slowest test case as a basis for comparison, the ranking from fastest to slowest is SA (530 s) < TS (535 s) < EP (584 s) < GA (679 s). It is clear that the population-based methods suffer some loss from the overheads required for book-keeping and manipulating a population of a few thousand solutions. However, the slowness of our GA implementation is probably due to a lack of code optimisation rather than any inherent feature of the algorithm.

It is important not to lose sight of wider issues in the midst of all this discussion about the relative qualities of algorithms. We have considered in detail how the algorithms behave on very difficult energy surfaces with competing minima of similar energy. Solution of the docking problem of course depends only in part on the algorithm; the other major component is the energy function. The lack of correlation of docking success rate with the median energy values in some test cases suggests that the energy function used in this work has deficiencies: for each of neuraminidase–DANA, thrombin–argatroban and thrombin–NAPAP, there exists at least one competing minimum of very similar depth to that corresponding to the crystal structure. We have suggested that the relative depth of this minimum reflects the fact that the energy function does not attach sufficient weight to salt bridge formation and is too biased towards steric fitting. If the energy function were improved in some way to make the competing minimum much less favourable, then any algorithm could be expected to produce a much higher docking success rate. Deficiencies in the details of the energy function are also indicated by the fact that the lowest energies emerging from successful docking runs are generally much better than that obtained for the original crystallographic conformation. This is mainly due to the torsional component of the ligand's internal energy, indicating either that the crystal structures contain some errors or, more likely, that the torsional energy term is too large. The results with NAPAP also indicate that the failure of the energy function to reward intramolecular hydrophobic interactions leads to a global energy minimum conformation which is not realistic. The original authors have had similar experience and this has led them to implement a much more complete internal energy representation, in the form of the DREIDING force field [48]. This improvement has yielded better solutions [49].

Our ultimate aim is to use PRO_LEADS to dock ligands for which the bound conformation at the receptor is unknown. In such a situation, one obviously cannot rely on an rms criterion to judge the success of a docking [50]. For such an application, it is essential to have confidence that the global minimum of the energy function corresponds to the bound conformation of the ligand. We have shown that the potential function in use in this work is reliable in most of the cases, but still requires some development before it can be used for 'de novo' docking on a variety of test cases.

Another point to come out of this work relates to what features make a docking test case difficult. It is our experience that increasing the bounding box size degrades the performance of the algorithms. Thus, one must strike a compromise between allowing exploration of the active site and obtaining reliable results. Another important factor is the number of rotatable bonds in the ligand. It is worth noting at this point that all of the intelligent algorithms are capable of virtually 100% successful rigid docking (based on the crystallographic conformation of the ligand) in all the reported test cases and even random search can obtain the correct answer in a non-trivial percentage of attempts! A simple count of the number of rotatable bonds, however, proves to be a rather naive measure of difficulty. It is the *situation* of the rotatable bonds within the molecule that is crucial, as evidenced by the experiments with thrombin–argatroban: a rotatable bond in a 'hinge' position in a molecule will clearly make for a more difficult search problem than one in a more terminal position. Mostly, the difficulty of our test cases is governed predominantly by the form of the energy surface. Our 'easiest' test case (DHFR–MTX) has seven rotatable bonds, but the energy surface seems to have just one deep minimum corresponding to the crystal structure, which the algorithms find with comparative ease. The more difficult test cases (neuraminidase–DANA, thrombin–argatroban and thrombin–NAPAP) have four, seven and six rotatable bonds, respectively, but have energy surfaces complicated by the existence of competing minima.

We have sought to make an unbiased comparison of four heuristic search algorithms with a random search procedure acting as a 'control'. The difficulties of carrying out such a comparison in terms of algorithm implementation and parameter optimisation are manifest. Nonetheless, we have carried out our development and experiments with an open mind and with no particular preconceptions of what the results should be. We have also made every effort to ensure that the adjustable parameters in each of the algorithms have been sufficiently

optimised to allow conclusions to be drawn. Our conclusion from this work is that the GA is the most effective algorithm in terms of the median energy of solutions. From the point of view of docking success rate, however, TS is superior. We believe that these results indicate that the GA is a very effective local searching algorithm but, correspondingly, that it has a tendency to become entrapped in low-energy local minima. Interestingly, a similar observation has been made recently by Meza et al. [51], who found that their GA for conformational search 'relatively quickly exhausts its ability to search globally and instead starts concentrating on small, but promising local regions'. TS seems to perform a much more coarse, global search which enables it to sample the global energy minimum more frequently, although less deeply, than the GA. There is a place for both types of behaviour in docking applications. One goal of the current work is to provide a reasonably objective identification of a ligand's binding mode so that new cycles of design and SAR can be initiated on the basis of the modelled results. In this case, a good global searching algorithm is very important. However, for those concerned with using docking for scanning databases, reliable energies are important so that new potential ligands can be compared on an equal footing with existing active molecules. Our findings suggest a number of possible future directions. First, a more sophisticated GA could yield improved performance. For a GA, trapping in local minima can be discouraged by measures aimed at introducing greater genetic diversity into the population. Possible approaches to this are to use subpopulations or an island model and the implementation niche restriction or sharing mechanisms [39]. Second, TS could be adapted so that towards the end of a search it behaves much more like a local search algorithm. This could be achieved simply by scaling down the tabu threshold during the docking run and concomitantly reducing the length of the tabu list. Thirdly, some form of hybrid search algorithm for docking could be developed. Combinations of heuristic search algorithms have been investigated by workers in other fields [52–55] and have shown promise. The results of our comparison suggest that the combination of the global searching TS with the local searching ability of the GA would be an obvious starting point. Finally, our results bear out the findings of workers in other research areas concerning the effectiveness of tabu search in combinatorial optimisation problems. Ideas related to tabu search have already been successfully applied in conformational searching [36–38] and here it has been shown that it performs extremely well as a molecular docking algorithm. It is possible that tabu search may be equally successful in other search and optimisation procedures associated with molecular design and modelling, for instance, de novo molecular design, the design of diverse combinatorial libraries and protein folding simulations.

Conclusions

We have presented a comparison of four searching algorithms applied to flexible molecular docking. One of the algorithms, tabu search, has not previously been used for this purpose. The algorithms have been applied to five test cases using the same energy function and a similar treatment of variables. The results indicate that all the methods are effective and give satisfactory performance. In particular, the genetic algorithm is generally found to give the lowest median energies, and the tabu search is generally found to locate the assumed global minimum more reliably. However, given the problems of parameter optimisation and differences in our implementation compared to other workers, these conclusions may not necessarily apply in all situations. It is safe to conclude, though, that tabu search is a promising new search algorithm for molecular docking and is worthy of further investigation for other applications in CAMD.

Finally, molecular docking is far from being a solved problem. In the future, we hope to address issues relating to energy/objective functions, the handling of receptor flexibility and the treatment of crystallographic water molecules.

Acknowledgements

The authors would like to thank Drs. David Fogel (Natural Selection Inc.) and Dan Gehlhaar (Agouron Pharmaceuticals Inc.) for helpful discussions concerning evolutionary programming and the docking potential function, Dr. Tim Auton (Proteus Molecular Design Ltd.) for help and advice related to the statistical treatment of the results, and the referees for constructive suggestions on ways to improve the presentation of this work.

References

1 Blaney, J.M. and Dixon, J.S., Perspect. Drug Discov. Design, 1 (1993) 301.
2 Jones, G. and Willett, P., Curr. Opin. Biotechnol., 6 (1995) 652.
3 Lybrand, T.P., Curr. Opin. Struct. Biol., 5 (1995) 224.
4 Rosenfeld, R., Vajda, S. and Delisi, C., Annu. Rev. Biophys., Biomol. Struct., 24 (1995) 677.
5 Gschwend, D.A., Good, A.C. and Kuntz, I.D., J. Mol. Recog., 9 (1996) 175.
6 Lengauer, T. and Rarey, M., Curr. Opin. Struct. Biol., 6 (1996) 402.
7 Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R. and Ferrin, T.E., J. Mol. Biol., 161 (1982) 269.
8 Goodsell, D.S. and Olson, A.J., Proteins Struct. Funct. Genet., 8 (1990) 195.
9 Yamada, M. and Itai, A., Chem. Pharm. Bull., 41 (1993) 1200.
10 Yamada, M. and Itai, A., Chem. Pharm. Bull., 41 (1993) 1203.
11 Leach, A.R., J. Mol. Biol., 235 (1994) 345.
12 Miller, M.D., Kearsley, S.K., Underwood, D.J. and Sheridan, R.P., J. Comput.-Aided Mol. Design, 8 (1994) 153.

13 Clark, K.P. and Ajay, J. Comput. Chem., 16 (1995) 1210.
14 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.
15 Gehlhaar, D.K., Verkhivker, G.M., Rejto, P.A., Fogel, D.B., Fogel, L.J. and Freer, S.T., In McDonnell, J.R., Reynolds, R.G. and Fogel, D.B. (Eds.) Evolutionary Programming IV (Proceedings of the Fourth Annual Conference on Evolutionary Programming), MIT Press, Cambridge, MA, U.S.A., 1995, pp. 615–627.
16 Jones, G., Willett, P. and Glen, R.C., J. Mol. Biol., 245 (1995) 43.
17 Judson, R.S., Tan, Y.T., Mori, E., Melius, C., Jaeger, E.P., Treasurywala, A.M. and Mathiowetz, A., J. Comput. Chem., 16 (1995) 1405.
18 Oshiro, C.M., Kuntz, I.D. and Dixon, J.S., J. Comput.-Aided Mol. Design, 9 (1995) 113.
19 Goodsell, D.S., Morris, G.M. and Olson, A.J., J. Mol. Recog., 9 (1996) 1.
20 Morris, G.M., Goodsell, D.S., Huey, R. and Olson, A.J., J. Comput.-Aided Mol. Design, 10 (1996) 293.
21 Rarey, M., Wefing, S. and Lengauer, T., J. Comput.-Aided Mol. Design, 10 (1996) 41.
22 Rarey, M., Kramer, B., Lengauer, T. and Klebe, G., J. Mol. Biol., 261 (1996) 470.
23 Welch, W., Ruppert, J. and Jain, A.N., Chem. Biol., 3 (1996) 449.
24 Abagyan, R.A., Totrov, M. and Kuznetsov, D., J. Comput. Chem., 15 (1994) 488.
25 Read, R.J., Hart, T.N., Cummings, M.D. and Ness, S.R., Supramol. Chem., 6 (1995) 135.
26 Verkhivker, G.M., Rejto, P.A., Gehlhaar, D.K. and Freer, S.T., Proteins Struct. Funct. Genet., 250 (1996) 342.
27 Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., Science, 220 (1983) 671.
28 Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., J. Chem. Phys., 21 (1953) 1087.
29 Szu, H.H. and Hartley, R.L., Phys. Lett., A122 (1987) 157.
30 Fogel, D., Evolutionary Computation: Towards a New Philosophy of Machine Intelligence, IEEE Press, Piscataway, NJ, U.S.A., 1995, pp. 121–186.
31 Yao, X. and Liu, Y., In Fogel, L.J., Angeline, P.J. and Bäck, T. (Eds.) Evolutionary Programming V (Proceedings of the Fifth Annual Conference on Evolutionary Programming), MIT Press, Cambridge, MA, U.S.A., 1996, pp. 257–266.
32 Saravanan, N., Fogel, D.B. and Nelson, K.M., BioSystems, 36 (1995) 157.
33 Glover, F. and Laguna, M., In Reeves, C.R. (Ed.) Modern Heuristic Techniques for Combinatorial Problems, Blackwell, Oxford, U.K., 1993, pp. 70–150.
34 Cvijovic, D. and Klinowski, J., Science, 267 (1995) 664.
35 Kvasnička, V. and Pospíchal, J., J. Chem. Inf. Comput. Sci., 34 (1994) 1109.
36 Goodman, J.M. and Still, W.C., J. Comput. Chem., 12 (1991) 1110.
37 Huber, T., Torda, A. and Van Gunsteren, W.F., J. Comput.-Aided Mol. Design, 8 (1994) 695.
38 Smellie, A., Teig, S.L. and Towbin, P., J. Comput. Chem., 16 (1995) 171.
39 Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, U.S.A., 1989.
40 Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., Numerical Recipes in C: The Art of Scientific Computing, 2nd ed., Cambridge University Press, Cambridge, U.K., 1992, pp. 412–420.
41 Gardner, M.J. and Altman, D.G., Statistics with Confidence, British Medical Journal, Tavistock Square, London, U.K., 1989, pp. 74–76.
42 Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. and Tasumi, M., J. Mol. Biol., 112 (1977) 535.
43 InsightII/Discover, v. 95.0, Molecular Simulations Inc., San Diego, CA, U.S.A., 1996.
44 Poornima, C.S. and Dean, P.M., J. Comput.-Aided Mol. Design, 9 (1995) 500.
45 Poornima, C.S. and Dean, P.M., J. Comput.-Aided Mol. Design, 9 (1995) 513.
46 Poornima, C.S. and Dean, P.M., J. Comput.-Aided Mol. Design, 9 (1995) 521.
47 Guida, W.C., Bohacek, R.S. and Erion, M.D., J. Comput. Chem., 13 (1992) 214.
48 Mayo, S.L., Olafson, B.D. and Goddard, W.A., J. Phys. Chem., 94 (1990) 8897.
49 Gehlhaar, D.K., private communication, 1996.
50 Gschwend, D.A. and Kuntz, I.D., J. Comput.-Aided Mol. Design, 10 (1996) 123.
51 Meza, J.C., Judson, R.S., Faulkner, T.R. and Treasurywala, A.M., J. Comput. Chem., 17 (1996) 1142.
52 Montoya, F. and Dubois, J.M., Europhys. Lett., 22 (1993) 79.
53 Kido, T., Takagi, K. and Nakanishi, M., Informatica, 18 (1994) 399.
54 Mahfoud, S.W. and Goldberg, D.E., Paral. Comput., 21 (1995) 1.
55 Tang, R.Y., Yang, S.Y., Li, Y., Wen, G. and Mei, T.M., IEEE Trans. Magnetics, 32 (1996) 1326.
56 Bolin, J.T., Filman, D.J., Matthews, D.A., Hamlin, R.C. and Kraut, J., J. Mol. Biol., 257 (1982) 13650.
57 Burmeister, W.P., Henrissat, B., Bosso, C., Cusack, S. and Ruigrok, R.W.H., Structure, 1 (1993) 19.
58 Lam, P.Y.S., Jadhav, P.K., Eyermann, C.J., Hodge, C.N., Ru, Y., Bacheler, L.T., Meek, J.L., Otto, M.J., Rayner, M.M., Wong, Y.N., Chang, C.-H., Weber, P.C., Jackson, D.A., Sharpe, T.R. and Erickson-Viitanen, S., Science, 263 (1994) 380.
59 Brandstetter, H., Turk, D., Hoeffken, H.W., Grosse, D., Stuerzebecher, J., Martin, P.D., Edwards, B.F.P. and Bode, W., J. Mol. Biol., 226 (1992) 1085.