

Molecular structure matching by simulated annealing. III. The incorporation of null correspondences into the matching problem

M.T. Barakat and P.M. Dean*

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

Received 20 August 1990

Accepted 12 September 1990

Key words: Simulated annealing; Molecular matching; Null correspondences

SUMMARY

This paper extends an application of the method of simulated annealing for molecular matching so that the best common subsets of atom positions can be identified. Null correspondences are introduced into the difference distance matrix to enable poorly matched positions to be ignored in minimizing the objective function. The efficiency of the algorithm in finding correct subsets is rigorously tested.

INTRODUCTION

Most molecular matching routines, that proceed via atom coordinate fitting, require an explicit assignment of atom correspondences. However, the central weakness in this approach is a derivation of the initial assignments that is free from preconceptions about the eventual superimposition of the molecules. Unbiased molecular matching is a key problem in drug design, particularly where similarities are sought between ostensibly dissimilar molecules. If there exists a set of dissimilar molecules which bind to the same region of a receptor site, can one determine a subset of atoms which show the best positional correspondences for the set of molecules? It may be inferred, and is highly likely if the molecules are dissimilar, that a common subset of atom positions have different bonding networks linking them. It is this rich variety of chemical structures, incorporating similar atom positions in space, which provides the wealth of possible drug patents.

Distance matrix methods can be used to search for correspondences between atom positions in two molecules. The distance matrix is invariant under rotational or translational transformations. The matrix is symmetric and composed of all interatomic distances with diagonal elements of

* To whom correspondence should be addressed.

zero; only the subdiagonal matrix needs to be computed. Subtraction of two distance matrices gives the difference-distance matrix. The sum of the elements of the difference-distance matrix can be used as an objective function to be minimized by rearranging the order of either one of the distance matrices combinatorially. Deterministic exchange algorithms cannot be used for combinatorial optimization since they are heavily dependent on the initial starting position. Simulated annealing circumvents this problem by incorporating a probabilistic element which permits some bad exchanges to be included as a feasible intermediate solution as the algorithm progresses.

In two previous papers we have presented a method of unbiased molecular matching by simulated annealing which provides a heuristic for overcoming the combinatorial problems associated with brute-force searching [1,2]. Consider the comparison between two molecules A and B, each containing N_A and N_B atoms respectively; atom positional correspondences are to be sought. Four situations can be categorised:

- (a) when $N_B = N_A$ and the whole of B is to be compared with the whole of A;
- (b) when $N_B > N_A$ and the structure of B is searched for a region that resembles the whole of A;
- (c) when $N_B \geq N_A$ and the structure of B is searched for a region that resembles only a defined region of A;
- (d) when $N_B \geq N_A$ and the structure of B is searched for a region that resembles the best unspecified region of A, where the size of this unspecified region is $< N_A$.

A solution for categories a, b and c has been presented [1,2]; here we consider a method for solving problem d.

In categories a, b and c the order of atoms in A is fixed, only the order of B has to be permuted to give the minimum in the sum of the elements of the difference-distance matrix. In case a, the number of permutations required is $N_B!$. If $N_B > N_A$ then the number of permutations of B is $N_B!/(N_B - N_A)!$ for case b. Case c is a related problem to b. These three cases are characterised by exchanges only in the order of B; consequently the objective function converges rapidly. Case d is a more complicated problem to solve because null correspondences have to be inserted into A. If k null correspondences are to be used, then the number, n_A , of unspecified atoms in A that has to be matched is $N_A - k$. There are $N_A!/[(N_A - n_A)!n_A!]$ combinations of n_A atoms in A that need to be considered as well as $N_B!/(N_B - n_A)!$ permutations for B. The convergence of the objective function is frustrated by the constant need to select different random combinations in A whilst re-ordering B. A schedule for exchanging A and B needs to be derived. The exchanges, randomly made to A or B, must not inadvertently weight the optimization trajectory along a particular pathway into a local, but non-global, minimum.

The question being addressed here is: if we have two molecules, what is the corresponding unspecified subset of a fixed number of atoms from each molecule which shows the best match? In this paper we outline an algorithm for examining the null correspondences problem and test the procedure with randomly generated data with a known solution. The following paper [3] makes use of this algorithm for deriving a subset of atom correspondences in a set of 6 molecular structures; atom positional similarities are analysed using classification theory.

METHODS

An algorithm for molecular matching by simulated annealing has been developed fully in Ref. 1. The objective function, E , to be minimized is

$$E = \sum_{i=2}^{N_A} \sum_{j=1}^{i-1} \Delta d_{ij}^{AB} \quad (1)$$

where

$$\Delta d_{ij}^{AB} = |d_{ij}^A - d_{ij}^B| \quad (2)$$

d_{ij}^A and d_{ij}^B are elements of the distance matrix for A and B, respectively. A control parameter, T, called the temperature because of its analogy with statistical mechanics, is initialized together with the length and number of Markov chains. At each step in a Markov chain a perturbation from a state, s, is made to some of the elements Δd_{ij}^{AB} to yield a state s'. Thus the objective function changes by an amount ΔE

$$\Delta E = E_{s'} - E_s \quad (3)$$

If the difference in the objective function, ΔE , is negative, the new state, s', is accepted unconditionally; if ΔE is positive the new state is accepted only if it satisfies the Metropolis condition [4]: $\exp(-\Delta E/T) > \text{random number from a uniform distribution } [0,1]$. This test allows uphill excursions in the objective function. When the length of the Markov chain is reached the control parameter, T, is decremented and the cycle repeated. This iteration continues until the change in the objective function during a Markov chain is acceptably small.

Test data

The input coordinates have been randomly generated for set B to lie in a unit cube. In order to test the algorithms exactly, so that the value of the objective function is zero at a perfect match, the coordinates for A have been set so that $N_A - k$ points are a subset of B. The k null points are non-repeatable random data. The order of B is then randomized to initialize the test for matching before the start of annealing. All experimental tests have been performed on 10 sets of non-repeatable random data.

Configuration exchanges

A scheme for the exchanges incorporating three null correspondences is shown in Fig. 1 where $N_A = 6$ and $N_B = 10$. The numbering of points in A is initially ordered whereas points in B are assigned a random initial ordering to reduce unnecessary configuration landscape problems [2]. The nulls are randomly assigned to A and their point numbers indexed in an array IA; corresponding points in B are also denoted to be nulls. Since $N_B > N_A$ there are some points in B that have no initial correspondence with any points in A; they are called the outer subset of B and their numbers are held at the end of an index array IB containing the atoms in B. Atoms in A that are assigned to be nulls together with their corresponding points in the inner subset of B are excluded from the calculation of the difference-distance matrix. Both A and B are perturbed separately by a pairwise exchange. An exchange of nulls in A can only be achieved by swapping one null point with a non-null point. This procedure is monitored by updating the index array IA when an exchange produces an accepted state, s'. The order of A is maintained; only the position of the nulls is varied. In contrast the order of molecule B is changed, if the proposed change is acceptable, by swapping a

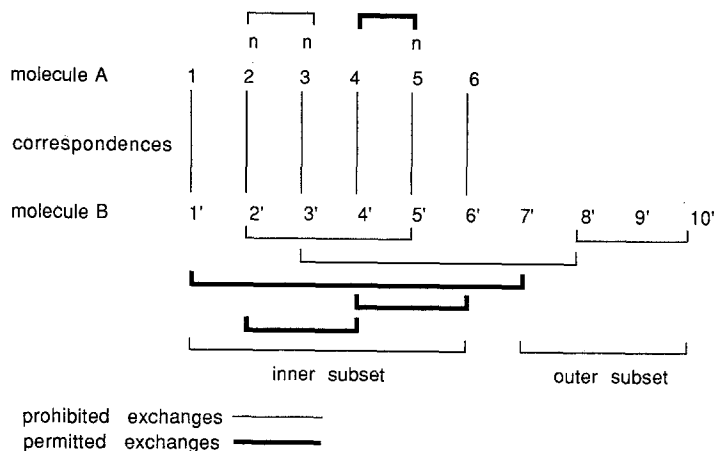


Fig. 1. A scheme for incorporating null correspondences (n) into the simulated annealing algorithm for molecular matching.

pair of points. Excluded exchanges in B are: two points from the outer subset; two null corresponding points from the inner subset; or, a point from the outer subset and a null corresponding point from the inner subset.

Algorithm

An outline of the algorithm for molecular matching using null correspondences is given in pseudo FORTRAN (specific variations in the algorithm are detailed later):

C read in coordinates for A and B

.

C read in number of nulls

.

C rescale data

.

C calculate distance matrices for A and B

.

C randomly allocate nulls in A

C initialize array IA

.

C randomly reorder B

C initialize array IB

.

C calculate E_s

.

C M = number of Markov chains

C L = length of Markov chain

C T = control parameter

C V = frequency of changes to A

C set M, L, T, V and stop criterion

.

```

C start annealing
  DO 30 I = 1,M
    DO 20 J = 1,L
      C decide whether to perturb A or B
        IF(RND[0,1).LT.V)THEN
          ITAG=0
        C swap a null within A randomly using IA
          .
        C s' = new possible configuration
          .
        C calculate Es'
          .
          GO TO 10
        ENDIF
      C swap a random pair within B, check for valid exchange with IA and IB
        .
      C s' = new possible configuration
        .
      C calculate Es'
        .
        ITAG=1
    10   ΔE = Es' - Es
    C Metropolis test
      IF(ΔE.LE.0.OR.EXP(-ΔE/T).GT.RND[0,1))THEN
        Es = Es'
        s = s'
      IF(ITAG.EQ.0)THEN
        C update IA
          .
          ELSE
        C update IB
          .
          ENDIF
        ENDIF
    20   CONTINUE
    C if stop criterion satisfied go to 40
      .
    C compute ΔT
      T = T - ΔT
    30   CONTINUE
    40   CONTINUE
    C print matched correspondences
      .
      STOP
    END

```

Scaling

We have previously shown [1] that in order to standardize the algorithm for matching molecules of different sizes, the objective function E can be made comparable by co-scaling A and B within a unit cube. Furthermore, the difference in the objective function, ΔE , between two proposed states shows a distribution with a mean of approximately zero. The distribution of ΔE can be standardized using the standard deviation of ΔE , $\sigma_{\Delta E}$

$$\Delta E_{\text{scaled}} = \Delta E \cdot C / (3\sigma_{\Delta E}) \quad (4)$$

where C is a constant, set here to 4, which controls the initial acceptance frequency from the Boltzmann probability distribution (see Ref. 1 for a detailed explanation). An appropriate scaling constant circumvents the need to choose an arbitrary initial temperature.

Length of Markov chains

In the previous study [1], in the absence of null correspondences, where changes were made only to the configuration of B , the maximum length, l_m , of the Markov chain was set to

$$l_m = 2N_A(N_B - 1) \quad (5)$$

Exploratory tests showed that this value is too small for the null correspondences problem; it does not take into account the larger number of exchanges that are needed to reach equilibrium at the end of the chain in the case where the configuration of A is frequently varied. The number of combinations of A is given by the binomial coefficient

$$C(N_A, n_A) = N_A! / [(N_A - n_A)! n_A!] \quad (6)$$

In this work, where $n_A < N_A$, the length of the Markov chain has been initially set to

$$l_m = \min\{2N_A(N_B - 1) \ln C(N_A, n_A), 50000\} \quad (7)$$

to take into account the frequency of changes to A as well as B . If during a Markov chain, the number of accepted exchanges exceeds $l_m/2$, then the chain is terminated on the basis that equilibrium has been reached.

Temperature decrement

A starting temperature of $T = 2$ was chosen as before [1] to give an initial acceptance probability close to 1. A dynamic temperature factor, f_{dyn} , has been used to decrease the temperature at the end of each Markov chain

$$f_{\text{dyn}} = \{1 + [T_i \langle E \rangle_{T_i} \ln(1 + \delta)] / 3\sigma_E\}^{-1} \quad (8)$$

where $\langle E \rangle_{T_i}$ is the mean value of E at a particular temperature T_i , σ_E is the standard deviation of E at T_i , and δ is a small positive number; $\delta = 0.1$ was found to be adequate.

Stop criterion

The maximum number of Markov chains is set to 500. This value is an overestimate and the value of the stop criterion effectively determines the end of the annealing process. When a perfect match is obtained with identical subsets of points in A and B, the difference-distance matrix elements are all zero and the algorithm stops. If the algorithm does not reach zero, two stop criteria have been used here: (a) when the acceptance ratio η is smaller than a specified value ($\eta = 0.008$), or (b) when the difference between the maximum accepted E and the minimum accepted E is equal to the maximum accepted ΔE [1,5].

Re-annealing

Where incorrect results arise in rigorous tests for simulated annealing, it appears that two distinct factors are responsible for the discrepancies. First, there is the trivial case where the great majority of point assignments are correctly matched and a few are mismatched. This error is probably due to the stop criterion η being too large and not allowing enough time in the later stages for equilibrium to be reached. Second, more serious mismatches occur when the configurational landscape contains a number of deep crevasse-like local minima into which the algorithm gets trapped [2]. In this case the order of the molecular matches is often reversed.

Attempts have been made here to minimize the effect of both problems by a systematic re-annealing strategy. In the first problem, where a few mismatches are encountered, the difference distance matrix is examined by summing the elements in each row. The rows are ranked to find the points giving the largest 25% of the row sums. This subset of points is then re-annealed from a starting temperature of $T = 1.5$ and is termed the first re-anneal step. Minor mismatches are frequently rectified by this procedure. The second problem can often be corrected by reversing the order of the points and using this as input to a subsequent re-annealing step starting at $T = 1.5$. The reverse re-anneal can then be improved further by re-annealing the worst 25% of the matches. The minimum in these four annealing steps is taken to be the global minimum.

RESULTS

Frequency of exchanges in A or B

The introduction of null correspondences in the simulated annealing procedure for molecular matching entails making exchanges to the distance matrix of both A and B. Combinations of points in A are matched by permutations of points in B. What strategy should be adopted for making the exchanges?

Changes in the objective function could be made firstly by varying either A or B separately, or secondly a combined change could be made to A and B simultaneously. The disadvantage of the second alternative is that two random pair exchanges have to be correct in the final assignment if a perfect match is to be found. The algorithm would not be minimizing through single perturbations to the objective function. For this reason the second alternative has been rejected here. The question that now has to be resolved is: how often must changes in A be made relative to B to ensure that perfect matches are found efficiently?

Consider the problem where $N_A = N_B = 18$ and k is allowed to vary from 1 to 16, then the number of combinations of points in A is given by Eq. 6 and the number of permutations in B

$$P(N_B, n_A) = N_B!(N_B - n_A)! \quad (9)$$

TABLE 1
COMBINATORIAL CHANGES INTRODUCED BY k NULL CORRESPONDENCES IN THE PROBLEM FOR $N_A = N_B = 18$

k	Combinations (A)	Permutations (B)	Product
1	1.8×10	6.4×10^{15}	1.1×10^{17}
2	1.5×10^2	3.2×10^{15}	4.9×10^{17}
3	8.1×10^2	1.1×10^{15}	8.7×10^{17}
4	3.1×10^3	2.7×10^{14}	8.2×10^{17}
5	8.6×10^3	5.3×10^{13}	4.6×10^{17}
6	1.8×10^4	8.9×10^{12}	1.7×10^{17}
7	3.2×10^4	1.3×10^{12}	4.0×10^{16}
8	4.4×10^4	1.6×10^{11}	6.9×10^{15}
9	4.9×10^4	1.8×10^{10}	8.6×10^{14}
10	4.4×10^4	1.8×10^9	7.7×10^{13}
11	3.2×10^4	1.6×10^8	5.1×10^{12}
12	1.8×10^4	1.3×10^7	2.5×10^{11}
13	8.6×10^3	1.0×10^6	8.8×10^9
14	3.1×10^3	7.3×10^4	2.2×10^8
15	8.1×10^2	4.9×10^3	4.0×10^6
16	1.5×10^2	3.1×10^2	4.7×10^4

Table 1 lists a number of these values; for a fixed value of k , the number of combinations in A is less than the number of permutations in B. The number of combinations increases to a maximum at $k = N_A/2$ but the number of permutations decreases as the number of nulls increases. The total number of non-repeatable arrangements is the product of the number of combinations in A and the number of permutations in B; in this case a peak number is found at $k = 3$.

The decision whether to modify the combination of A, for each iteration, is taken if a randomly generated number $[0,1)$ is less than the parameter V , otherwise a permutation exchange in B is performed. The formula

$$V = k/(N_A + N_B) \quad (10)$$

allows V to vary linearly between 0 and 0.5 with k . With small numbers of nulls changes are made predominantly to B; whereas with large numbers of nulls changes are made with nearly equal frequency to A or B. Table 2 shows the effect of different values for V on solving the problem for $N_A = N_B = 18$ and with $k = 3, 9$ or 15 . When V is dependent on k , perfect matches were found for $k = 3$; when $k = 9$ only one pair of assignments from one of the 10 runs gave an incorrect match; with $k = 15$, 8 out of 10 attempts gave the correct match but two runs gave completely wrong matches. If V is not allowed to vary with the size of the combinatorial problem then the percentage of correct assignments and the fraction of perfect runs is lower. However, the results with $V = 0.01$ are in all cases 80% correct, or better, and the time taken to reach the solution is similar to that with a variable V . Similarly, when there is an equal frequency of exchanges to A and B ($V = 0.5$) the optimization performs well. In all cases, with large numbers of nulls ($k = 15$) some bad mismatches were produced irrespective of the value for V .

TABLE 2
THE EFFECT OF VARYING THE PARAMETER V IN THE MATCHING OF PROBLEMS WITH $N_A = N_B = 18$
AND FOR DIFFERENT VALUES FOR k NULLS

V	k	Mean initial E (with nulls)	Mean final E (with nulls)	C.P.U. (s)	% correct assignments	Fraction of perfect runs
0.01	3	0.2357	0.0107	4.8	98	9/10
	9	0.2396	0.0164	19.1	82	5/10
	15	0.2210	0.0099	65.0	80	8/10
0.5	3	0.2415	0.0000	2.4	100	10/10
	9	0.2360	0.0085	16.5	80	8/10
	15	0.2147	0.0040	63.0	80	8/10
$V = k/(N_A + N_B)$						
0.0833	3	0.2485	0.0000	2.3	100	10/10
0.25	9	0.2312	0.0027	15.9	98.9	9/10
0.4166	15	0.2060	0.0056	65.4	80	8/10

Statistical data are taken from 10 complete runs using randomly generated coordinates. E is the value of the objective function.

Different problem sizes

Table 3 illustrates the ability of the algorithm to minimize problems with $N_A = N_B = 25, 50$ and 70 respectively; the combinatorial difficulties associated with each value of k are also given. The trend in all three tables shows that the amount of time required to optimize a problem increases with the increased number of nulls and with all problem sizes. This increase in computer time is observed even though the combinatorial difficulty is less. This fact is clearly displayed in Table 3 where $N_A = N_B = 25$ and $k = 5$ and 20 ; the number of combinations is identical but the number of permutations is reduced from 10^{23} to 10^6 , yet the average computer time increases from 14.3 s to 228.1 s. This trend is also observed in Table 2. The computing time required is largely independent of the value used for V since very similar times are found with V set to 0.01 or 0.5 .

With each of the three different values for N_A the quality of the solution appears to deteriorate when $k \geq N_A/2$. This decline in the performance of the algorithm occurs despite the enormous decrease in the number of possible assignments. For example, with $N_A = N_B = 25$ and $k = 20$ only 24% of the assignments are correct with 10^{11} possible arrangements, whereas with $k = 5$, 90% of the assignments are correct even though there are 10^{27} possible arrangements. Similarly, for $N_A = N_B = 70$ with $k = 10$ the number of arrangements is 10^{105} , but with $k = 40$ the number of possible arrangements falls to 10^{71} and yet the percentage of correct assignments is reduced by half as the number of nulls is increased.

DISCUSSION

Unbiased molecular matching, using atom positions, is a combinatorial problem that can only be solved unambiguously by brute-force searching methods. However, these methods are not practicable for problems involving large numbers of atoms because of the massive number of per-

TABLE 3
PERFORMANCE OF THE SIMULATED ANNEALING PROGRAM FOR MATCHING THREE-DIMENSIONAL COORDINATE DATA

k	% correct assignments	Mean C.P.U. (s)	Combinations (A)	Permutations (B)	Product
$N_A = N_B = 25$					
5	90.0	14.3	5.3×10^4	1.3×10^{23}	6.7×10^{27}
10	100.0	27.8	3.3×10^6	4.3×10^{18}	1.4×10^{25}
15	79.0	40.9	3.3×10^6	1.2×10^{13}	3.9×10^{19}
20	24.0	228.1	5.3×10^4	6.4×10^6	3.4×10^{11}
$N_A = N_B = 50$					
5	91.8	50.7	2.1×10^6	2.5×10^{62}	5.5×10^{68}
10	83.7	93.3	1.0×10^{10}	8.4×10^{57}	8.6×10^{67}
15	87.4	114.6	2.3×10^{12}	2.3×10^{52}	5.2×10^{64}
20	91.8	109.8	4.7×10^{13}	1.3×10^{46}	5.9×10^{59}
25	66.0	186.0	1.3×10^{14}	2.0×10^{39}	2.5×10^{53}
30	72.5	186.4	4.7×10^{13}	1.1×10^{32}	5.4×10^{45}
$N_A = N_B = 70$					
10	81.8	130.2	4.0×10^{11}	3.3×10^{93}	1.3×10^{105}
20	80.6	167.9	1.6×10^{17}	4.9×10^{81}	8.0×10^{98}
30	89.7	217.0	5.5×10^{19}	4.5×10^{67}	2.5×10^{87}
40	41.0	318.2	5.5×10^{19}	1.5×10^{52}	8.1×10^{71}

Coordinate points were generated randomly. Ten repeat runs were performed for each entry. Problem sizes are for $N_A = N_B = 25, 50$ and 70 with different numbers of k nulls.

mutations. Simulated annealing has proved to be a realistic computational procedure for discovering good solutions to the atom matching problem [1,2]. In this paper we have extended the method to include null correspondences so that an unspecified subset of points, of defined size, from a pair of structures can be sought which gives the best match. In concept, the method is derived from earlier work [6] where hydrogen-bonding atom positions were matched as acceptors with acceptors, or donors with donors, but not acceptors with donors unless one atom of the pair was of mixed function. Algorithmic efficiency has been examined rigorously by using an input consisting of randomly distributed data but with a known solution for each number of nulls being tested. Multiple examples with varying positional data for problems of the same type can be used for statistical testing. This procedure provides an unequivocal assessment of the performance of the method.

The molecular structures of two molecules A and B can be represented by their distance matrices; if $N_A < N_B$ then the distance matrix of A can be compared with the partial distance matrix of B. The order of atoms in B has to be permuted so that the difference-distance matrix is a minimum for an optimum match. Null correspondences are introduced so that the algorithm can ignore badly matched positions which are reflected in certain rows and columns. Different combinations of atoms in A are achieved by randomly altering the positions of the nulls in A; on the other hand

the order of B is changed by permutation. This development in the matching method enables the best subset of matched atoms to be found.

Although the introduction of nulls reduces the total combinatorial problem (Table 1), paradoxically it increases the difficulty of reaching the correct solution (Table 3). This problem is manifest by both an increase in the amount of computing time needed to reach a solution, because of an increase in the number of Markov chains needed, and by the decrease in frequency of obtaining the correct solution (Table 3). The problem does not appear to be directly related to the frequency of changing the combination in A or the permutation in B. What is clear is that the objective function converges slower as the number of nulls increases. The reason for this observation may lie in the fact that when numerous nulls are introduced there may be a greatly increased number of nearly equivalent local minima close to the global minimum. If this state is reached, the algorithm would spend much of its time searching in these regions and eventually become locked into one of them. In other words, the introduction of nulls creates configurational landscape problems [2]. When matching molecules containing up to 70 atoms, it would appear from these experiments that if $k < N_A/2$ then at least 50% of the discovered matches would be close to the global solution. An alternative explanation for the deterioration in convergence of the objective function, by the introduction of null correspondences, may be that the nulls generate an element of frustration similar to that experienced in simulated annealing of spin glasses [7].

The computing method developed in this paper provides a rapid solution to the question of how to select, without bias, a subset of atoms in two molecules that show the best positional matches. We have shown that for structures with up to 70 coordinate positions the best matched subsets containing 50% or more of the points can be found satisfactorily. In the following paper this algorithm is used to identify, and classify, common positional substructures in a small set of molecules [3].

ACKNOWLEDGEMENTS

The authors wish to thank the Wellcome Trust for generous financial support to M.T.B. by a Wellcome Prize Studentship and to P.M.D. through the Principal Research Fellowship Scheme. M.T.B. is grateful to St. John's College Cambridge for a Benefactor's Research Studentship. Part of the work was supported by the SERC Cambridge Centre for Molecular Recognition.

REFERENCES

- 1 Barakat, M.T. and Dean, P.M., *J. Comput.-Aided Mol. Design*, 4 (1990) 295.
- 2 Barakat, M.T. and Dean, P.M., *J. Comput.-Aided Mol. Design*, 4 (1990) 317.
- 3 Papadopoulos, M.C. and Dean, P.M., *J. Comput.-Aided Mol. Design*, 5 (1991) 119.
- 4 Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., *J. Chem. Phys.*, 21(1953)1087.
- 5 Huang, M.D., Romeo, F. and Sangiovanni-Vincentelli, A., An efficient general cooling schedule for simulated annealing. In *Proceedings IEEE International Conference on Computer-Aided Design*, Santa Clara, CA, 1986, pp. 381-384.
- 6 Danziger, D.J. and Dean, P.M., *J. Theor. Biol.*, 116 (1985) 215.
- 7 Toulouse, G., *Commun. Phys.* 2 (1977) 115.