Protein Engineering vol.5 no.7 pp.637–645, 1992

Potential of genetic algorithms in protein folding and protein engineering simulations

Thomas Dandekar and Patrick Argos[1]

European Molecular Biology Laboratory, Postfach 10.2209, D-6900 Heidelberg, Germany

[1]To whom correspondence should be addressed

Genetic algorithms are very efficient search mechanisms which mutate, recombine and select amongst tentative solutions to a problem until a near optimal one is achieved. We introduce them as a new tool to study proteins. The identification and motivation for different fitness functions is discussed. The evolution of the zinc finger sequence motif from a random start is modelled. User specified changes of the λ repressor structure were simulated and critical sites and exchanges for mutagenesis identified. Vast conformational spaces are efficiently searched as illustrated by the *ab initio* folding of a model protein of a four β strand bundle. The genetic algorithm simulation which mimicked important folding constraints as overall hydrophobic packaging and a propensity of the betaphilic residues for *trans* positions achieved a unique fold. Cooperativity in the β strand regions and a length of 3–5 for the interconnecting loops was critical. Specific interaction sites were considerably less effective in driving the fold.

Key words: evolution/mutation/optimization/protein folding/protein structure

Introduction

Genetic algorithms are a well known tool for optimization tasks. The search routines use the mechanisms of natural selection and genetics. A tentative solution to a given problem or query is encoded as a long string of characters ('nucleotides') in a 'genome'. A population comprises many individual strings with their respective genomes. Individuals from a start population have only random strings. The probability of becoming a parent for the next generation rises according to the fitness of the solution encoded. There is a predetermined amount of mutation and crossing over between the code strings of selected parents for the next generation. After several generations of fitness selection, mutation and crossing over, individuals arise close to an optimal solution (Goldberg, 1989).

Such a search procedure can cover vast solution spaces typical of proteins and their attributes reliably and quickly. An example is given where a potential zinc finger primary structure is found from 20^{30} possible sequences with little computer effort on a small workstation. Another example shows how the algorithm can be used to search for solutions in a protein engineering problem. An initial wild-type sequence is altered by the genetic algorithm to optimize effectively and simultaneously several criteria such as greater helix stability and preserved core volume. Each of the new mutated structures identified by the algorithm optimizes the total of the engineering parameters. Thus the important starting features of the wild-type primary structure can be conserved while the desired features for engineering are implemented.

A final example was used to investigate the potential of genetic algorithm simulations to fold a protein *ab initio*. In a model four β strand structure the genetic algorithm reached smoothly a unique protein fold. Since many individuals (and even more schemata, Goldberg, 1989) can be processed in parallel, the genetic approach offers a quicker optimization potential than conventional techniques. Furthermore the relative importance of various physicochemical forces to achieve the β strand-rich fold can be explored. Different forces (e.g. electrostatic interactions and hydrophobicity) were used in the fitness function for selection and tested for their ability to encourage the properly folded protein state. Though the present models and goals are simplistic, the potential of the method for more complex tasks is clearly suggested.

Materials and methods

The genetic algorithm

The simulations were carried out on a VAX 3200 workstation. Programs for this study were written (T.Dandekar) in Pascal utilizing modified versions of the simple genetic algorithm as described by Goldberg (1989). In each of our tests the genetic algorithm started with a population of several random bit strings (30–500). The strings had to be decoded in a subprogram depending on the problem. They were translated into amino acid sequences according to the genetic code for two examples modelling zinc finger evolution and λ repressor engineering or they were interpreted as internal coordinates of a model protein in folding trials. The fitness of the decoded bit string solution was calculated according to certain parameters. In the first example, the amino acid matches to a zinc finger consensus (Gibson *et al.*, 1988), the differences in amino acid composition to the average found in zinc finger sequences, and the number of stop codons were multiplied by specific weight values and added to yield the total fitness value of a given bit string. Stop codons which interrupt the zinc finger peptides were heavily selected against and received a large negative weight. Parameters and weights were carefully chosen and empirically tested (see in Discussion an extensive description for the motivation of the different fitness functions) to model a particular problem properly.

A dice is rolled to pick individuals to become parents for the next generation. The probability of an individual being picked increases directly with its fitness value. A selected individual is either directly copied to the next generation or undergoes recombination (the chance for this was set to be 0.2 per individual) at a random crossover site with another selected individual, exchanging the bit string after the recombination site with that from the other individual, resulting in a new generation. Low frequency random bit mutations were also incorporated during crossing over and copying. The mutational level was set to be just below or equal to one mutated bit per individual, allowing for quick evolution in the simulations. Dice selection, mutational copying and crossover are continued until a whole new population, a new 'generation' is achieved. The bit strings of these individuals are decoded and fitness values calculated. Each new

maximum of fitness is reported by the computer program and then the selection for the following generation starts. After sufficient trials (100 or more generations), the individuals encoding near optimal solutions emerge. A further refinement of the simple genetic algorithm (Goldberg, 1989) involved the collection of the fittest individuals from several selection runs and their use in a final competition run, again against a random background population. Each run was called an 'epoch' and consisted of 100 or 120 generations of evolution and selection. Individual experiments ranged from 0 (one selection run only with no final competition) to 24 epochs (24 selection runs and one final competition). Run times varied between 10 min and 8 h on a VAX station 3200 in batch mode.

Simulation of protein motifs

Each string was decoded according to the genetic code translating it in groups of six bits and reading only the frame starting with the first bit such that a start codon was not required. In the zinc finger sequence simulation, the fitness function used for selection was:

$$\text{fitness} = \text{Aadiff} - (300 \times \text{Aastop}) + (100 \times \text{Consensus matches}).$$

Aadiff is the sum over all amino acid types of the squared difference between the amino acid composition of the evaluated sequence and the average amino acid content for the zinc finger motifs according to Gibson *et al.* (1988). Aastop was incremented by one for each stop codon found. Consensus matches were calculated according to amino acid types allowed at certain positions in the zinc finger consensus (Gibson *et al.*, 1988). The completely conserved cysteines and histidines which coordinate the zinc finger cation were upweighted by a factor of two.

```
alignment position : 1, 2,3, 4,6,9,11,13,17,19,22,24,26,27,28
allowed amino acids: E,KR,P,YF,C,C,KR, F, S, L, H,KR, H, T, G
position weight    : 1, 1,1, 1,2,2, 1, 1, 1, 1, 2, 1, 2, 1, 1
```

Other sites were allowed any amino acid type.

Sequence engineering of the N-terminal half of the λ repressor involved the following fitness function:

$$\text{fitness} = w_1 \times 50 \times \text{Aasolvent} + w_2 \times 50 \times \text{Aaturn} + w_3 \times 50 \times \text{Aahelix} + w_4 \times 50 \times [50 - \text{abs}\ (549\text{-core})] + w_5 \times 100 \times \text{Aadiff}.$$

This example was used to reproduce a hypothetically more stable sequence from a wild-type start where substituted residues in secondary structures showed a greater preference for the structural type and yet important folding constraints were maintained as hydrophilic residues at the protein surface or a near constant total volume for the hydrophobic core side chains. Aasolvent depends on the solvent accessibility of the residues in the N-terminal half of the λ repressor tertiary structure (Pabo and Lewis, 1982). The residues were divided into three categories: <20 (buried), $20-60$ (neutral) and >60 Å^2 (exposed) accessible surface determined by the routine of Kabsch and Sander (1983). Amino acid preferences to be buried (G,A,L,I,V,M,C,F), neutral (S,T,P,W,H,Y) or exposed (R,K,E,Q,D,N) were chosen according to Janin *et al.* (1978). If in the decoded trial protein an amino acid with a buried preference occurred at a position known to be buried in the tertiary protein structure, Aasolvent was increased by one. An exposed amino acid in this place diminished Aasolvent by one while an intermediate or neutral amino acid did not prompt a change in Aasolvent. Exposed positions were treated analogously.

Regions of turn and helix preference were determined according to their observed secondary structure as determined by the routine of Kabsch and Sander (1983). Amino acid preferences for turns and helices were taken from Chou and Fasman (1978). Turn formers (G,S,D,N,P) in a loop position increased Aaturn by three while turn breakers decreased it by two (A,L,M,H,V,I,F). Aahelix was calculated similarly, but there were five classes: strong formers (A,L,M,E) increasing Aahelix by +4; weak formers (+2: K,F,Q,W,I,V); indifferent formers (+1: D,H); weak breakers (−4: Y,N); and strong breakers (−12:P,G). The respective weights were chosen to have roughly equal probability for a change to lower or higher helicity; similarly the overall values of the parameters were chosen to be about equal. The weights $w_1 - w_5$ can be modified according to the user's needs. In the example shown, w_4 and w_5 were set to 4 and the others to 1.

Aadiff counts the number of times amino acids as preferred by the user occur in the trial protein. The preference chosen here was for charged amino acids (D,E,R,N,Q,H,K). The term 'core' involves the residues L18, V36, M40, V47, F51, L57 and L65 which constitute the λ repressor structural interior. Observed mutations of these residues such that the repressor protein maintained function showed that their total volume must be between 493 and 586 Å^3 (Lim and Sauer, 1989). The total core volume of the trial protein is calculated allowing for all amino acid types at this position. A positive fitness value is maintained if total core volume differs by <50 Å^3 from that of the wild-type (549 Å^3). Charged core residues are punished as their presence did not maintain functionality (Lim and Sauer, 1989). This was achieved by setting their volume to zero, resulting in a large deviation from the allowed volumes. More parameters and more sophisticated calculations can easily be introduced in the fitness function as additive terms or subterms.

Protein folding trials

Bit strings were interpreted as internal coordinates on a tetrahedral lattice. The direction from one C_α atom to the next of the model protein (four possible directions from the C_α atom at the centre of a tetrahedral lattice unit to the next chain point) was determined by decoding two bits of the string. Backwalks in the chain trace leading to clashes were allowed but heavily selected against (see below). A further simplification was to assume equal distances between the residues. Side chains were not modelled explicitly.

The fitness function for the first selection (specific attractive forces) was calculated as:

$$\text{fitness} = c_1 + \text{clashes} + \beta\text{strand} + \text{CNinteraction}.$$

The term c_1 is a positive number used to maintain positive fitness, especially important in the earliest generations which displayed considerable clashes or atomic overlaps. It was typically set to 1500 × length (in bits) of one individual. The number of clashes was incremented by +2 to allow easy removal of the final and only clash typical of later generations. The clash count was multiplied by a large negative weight, set to −2000 in the trials shown. Only the number of those positions in the protein assigned as β strand residues constituted the value of the β strand term. A strand position $i+1$, *trans* relative to the residue i, was modelled by selecting both a similar direction in the two vectors defined respectively by the C_α atom pairs $(i+1, i)$ and $(i-1,$

$i-2$) and different directions involving residue pairs ($i+1$, i) and (i, $i-1$), resulting in an extended zig-zag pattern. For each such residue found the β strand value was increased by betarow. Betarow, starting with a value of one, was increased by +2 for each such parallel direction event found within a strand. However, at the terminal position of each strand, it was set to one. This rule proved superior to others tested as it favoured in a simple way growth of betaphilic start nuclei within the strand and yet allowed modelling of the β strand breakers at the strand termini. The β strand value was then multiplied by its weight (400 for the trials in the results). The CNinteraction was used to model an antiparallel strand bundle (Figure 1). The strands were numbered 1 to 4. To promote the bundle, lattic distances between the N-terminus of an even strand and the C-terminus of an uneven strand (and vice versa) were added for all possible pairs. To increase parallelism of the strands the difference between the $C_{even}-N_{uneven}$ and $C_{uneven}-N_{even}$ distances for each strand pair were also added. The sum was multiplied by a high negative weight (-600 in the results). If the maximum CNdistance found was larger than the length of a strand (eight residues), indicating no tight packing of the protein, then this difference was multiplied by a negative weight of -5000 and added to reduce further the fitness of the particular string coded solution.

For the second folding example, the hydrophobic selection trial, all terms and weights are as previously described except that CNinteraction was replaced by a scatter function:

$$\text{fitness} = c_1 + \text{clashes} + \beta\text{strand} + \text{scatter}.$$

The scatter is simply the calculated sum of the distances of each point or atom in the model protein from its centre of mass. The sum was multiplied by a negative weight (-900 in the results). This function modelled the effect of an overall hydrophobic contraction in promoting secondary and higher protein structure.

Protein structures calculated in the folding simulations were further analysed and drawn using the protein visual characterization system developed by G.Chelvanayagam (unpublished). Labelled residues of the chain trace are shown as glycines (Figures 2–4) since different side chains were only implicit. Only two types of side chains were utilized: those preferring β configuration and those without such a preference.

Results

The first example models the evolution of the zinc finger motif. The total sequence space for this motif (30 residue motif, 20 naturally occurring amino acids) is 20^{30} such that an exhaustive search of each of the possible combinations is virtually impossible. The example provides a test of whether the random mutation and recombination events of the genetic algorithm can search a vast space of residue combinations quickly. The genetic algorithm driven search with a random sequence start population quickly (700 generations, needing only 19 min on a small VAX workstation) leads to zinc finger-like sequences (Table I). A population of 100 random 180-bit strings represents different genomes decoded according to the genetic code. Individuals encoding peptides closer to the zinc finger concensus (Gibson *et al.*, 1988) by overall composition and alignment position received higher fitness scores (see Materials and methods) and were preferentially selected to be parents for the next generation. Further, stop codons were selected against. The algorithm optimizes all three parameters simultaneously and, though it starts with reshuffling random sequences, it reliably finds the region in sequence space with the highest possible similarity score (19,

Table I. Evolution of a zinc finger sequence

0[a]	TLSIWRWDSVAPKEVYCQRVPNCCPANLTR	(first individual[b])
1	EVFVTWRNSSFCEAPMRARKVYANSVVTLI	
2	FKGILFLSVWLYATAGDDLRTTSIPTEFGF	
4	HRQLGDYVPCVLKTLISSLAAGRRKLAAKH	
5	HRRFGDYAPCVLKTLISPLAAGPRKLAAKH	
6	HRRFGDYVPSVLKALISPLAAGLRKSTTTI	
7	HRNRGCALVLPTRKDSSPLPAGPRKSTTTV	
8	SYPFACKLVLSTRNDCSPLRPGLRKSTITI	
9	SYPFACKLVDRTRIGSSPLVPKPRRATTMF	
10	SHPFACKLALLTRRGDSPLGAGRREHTTNI	(generation 51)
11	SHPFACKLAARTRSGDSPLGVGRREHTTNI	
12	SRPFACKLAPRTHTRDSPLKVGRREHTTYI	
13	TRPFACKLVLRMGTHDSPLVLRGRQHTGYI	
13	SRPFECSLTVRACAHDSPLVLRGRQHTGYI	(best of epoch 1)
14	SRPFTLDARSRTFWEASLLYHHPRSHTGPI	(best of epoch 2)
16	ERPYDCLFCKRPTWQNSGEVTHYRAHKGGT	(best of epoch 3)
13	ERPYMCGLCKRHGILVSMLRTSFRLTTSPQ	(best of epoch 4)
17	ERPTVCEGCSRNFSRGSVLSIHMRWHVGGE	(best of epoch 5)
12	FRMAACWQCLRSRSHPSVLRQTITLHTGVG	(best of epoch 6)
17	ERPTVCEGCSRNFSRGSVLSIHMRWHVGGQ	Final selection start
18	ERPTVCEGCSRNFSDASLLYHHPRSHTGPI	(Generation 10)
19	ERPYDCEPCSRNFSRGSVLSTHVRSHTGPN	(Generation 20)
19	ERPYKCGKCFKYFARPSELGTHSRSHTGRN	(Generation 101)
	* * * *	conservation[c]
Negative Controls[d]:		
17	DKPYLCEECPRIQLVNSFLSDHARIHTGRR	(mutation rate 0.03)
14	RRPESCGTCLRSFSRGGSTSIHPKPHVGPL	(mutation rate 0.00)

[a]The number of matches to the consensus is given on the left, with 19 the maximum possible.
[b]The start population consisted of 100 random 180 bit strings decoded according to the genetic code (including STOP codons). The fitness function involved similarity (matches and overall composition, see Materials and methods) to the zinc finger consensus (Gibson *et al.*, 1988).
[c]Asterisks indicate the fully conserved zinc liganding residues.
[d]All other conditions are identical; the final result is given.

see Materials and methods) to the zinc finger consensus. The algorithm models at the same time the evolution of random peptides to a functional protein. This example also shows that the algorithm can reach the consensus in the simulation with optimized evolution parameters. That evolution is mimicked can be seen by setting the mutation rate higher (three per 100 bits, Table I) such that the consensus sequence is not attained, albeit using identical fitness functions and processing conditions. Mutation rates above the threshold (in our simulations higher than 1 bit per each individual of 180 bits) slow down evolution. Similarly, relying in each new generation of individuals on recombination alone is also not able to reach the concensus (Table I, mutation rate 0.00).

A potential application of the genetic algorithm involves protein sequence engineering tasks. We now do not start with random sequences but want to improve a known wild-type sequence to get an optimized structure. It is easy to select analytically good sites for mutations which fulfil one criterion, for example helix stability. The advantage of genetic algorithms in this type of protein engineering is their ability to select mutations which simultaneously fulfil many weighted criteria, typical in applied protein engineering. The data shown in Table II illustrate the design of new mutations in λ repressor. The starting population consists of 40 612-bit strings all encoding the 102 residues of the N-terminal part of λ repressor (Lim and Sauer, 1989) but each containing random bit permutations which do not alter the 102 wild-type amino acid residues. The program shown optimizes the following parameters (see Materials and methods): helix- and turn-preferences for residues found in such secondary structures, amino acid type preferences at a given sequence position according to the solvent accessibility of the side chain in the known

Table II. Engineering trial of λ repressor

```
r.Fit[a] n[b]  amino pref    core          helix       turn      accesibility

1.182  0  6.08000E+04  4.00000E+04      6560        400       10400[c]
     STKKKPLTQEQLEDARRLKAIYEKKKNELGLSQESVADKMGMGQSGVGALFNGINALNAYNAALLAKILKVSVEEFSPSIAREIYEMYEAVSMQPSLRSEYE[d]

1.182 ......................................................................................................[e]

1.192 .........................E..........V...................................E.............................
1.195 ...........H........V....................................D............................................[f]
1.196 ................................................V...................................C.I.............NQ
1.200 .............................D..............................................L.........................* 2[g]
1.233 ............K.....I.............H....H.V.K...RL.....DNY.I.G..V.............I........H...K....R........
1.234 .SE.........K...S.I.....R....EI.H....H.V.KS..RL.....DNY.I.D..V....K........I........H...K....R.....V..
1.248 ..N...SA..EHI...S.IVT...R.....I.H.C..H.VRK...RL.G...DKI.I.DC..V...RT.NAWL.N.TRW.......K.......S.RKL..Q
1.248 44  7.04000E+04  4.00000E+04      6240        200       8000

1.185 ....................P...R...E..............................................L..........................
1.188 .....................F................................................................................* 1
1.197 .S..........H.......................................................M...............HQ...A............
1.198 ............................Q........................N...........................S.....K..............
1.204 ......S..K...........F........I..W...........E....................V.....................G.............
1.208 .....................F..............................................................................H.* 2
1.210 ............................Q........................N.............H.............S.....K..............
1.214 .........G..........M.......................H.R..S.KS..............VR........R.G....H.........S....K.K
1.220 T...S.........D.K...M......G....QT..........R.R..R.KS..............VR..G....RF.G.K.C...V...C....Q..K.K
1.221 T...S.........D.K....C.....G....QT..........R.R..R.KS..............VR..G....RF.G.K.C.I.V..C.....Q..K.K
1.221 27  6.56000E+04  4.00000E+04      6960        400       9200

1.202 ..........................................S..R...........................KY..................H........
1.209 ..........R..G.....S......................R..A...Q....................L.....L.......H...........M.....
1.210 ............H..............KH..G........RL..D................V.D.TRV.....K......P......KT........GRQ..
1.224 .I.........................KH..R.K......RL..........S..........D.TRV....IK......P......KT..I..R.......
1.233 ...E........H..............KH..G.Q......RS...V.......F.......D.D.TRV.....K......P......KT..K.....GRQ..
1.274 .I.........................KH..R.K......RLR.........RN.........D.TRV....IK......R......KT........D.Q..
1.290 .I.EN..R...............Q...KH..RHKP.....RLR.R....F..RN..K.F....D.TRVF...IK.LT.N.R....I.RT..R..P..D...V
1.290 38  7.84000E+04  4.00000E+04      1360       -800       10000

1.189 .R.........Q..................................................T..........G............................
1.199 ....R.................................................................................................* 1
1.200 ...........Q........V.....................................G...H...............P.............L.........
1.201 ...N................M.....................D................................S...........K...C......C..D
1.205 .......S...Q.H......V.....................................G...H..M............P.............LE........
1.211 P...R.......................................................C..S.....L....G.......................N...
1.222 .......S...Q.H......V...R....................R........T...G...H..M.L..........P..........Q..LE..S.....
1.259 PR..R................................N......................C..S.....L....G..........N..D..K......N...
1.303 R.....PS...Q.H......V.N.R..A..................R..ET...T...G...H..M..LN.K..........NC..H........N.P..FD
1.303 26  7.36000E+04  4.00000E+04      6480        400       9800

1.192 ...........................K...............E........................Q................Q................
1.209 ..............................................................................................R.......* 1
1.225 .M.........................D.D........E......R......D.....K........EM..................I.....K.......V
1.254 ................W........................RK..R............................Q...K...........D.I.....N...
1.261 .A.N.A...R..HV........K....KSR.......V...L.E.E.RT..SE..VI..H..Q.D.A...P.DQ..A......V.D.HR.I.....P.....
1.261 34  7.04000E+04  4.00000E+04      6880        400       8400

(final competition[h]:)
1.334 28  7.52000E+04  4.00000E+04      8560        400       9600
      R.....PS...Q.H.....VV.N.R..A..................RR.ET...T...G...H..M..LH.K..........NC..H........N.L..FD
1.363 .I.EN..R..............Q.R..KHD.RHKP.....RLR.R....F..RN..K.F....D.TRVF...IK.LTRW.......K.......S.RKL..Q
1.374 .I.QD.SR..............Q.R..KHD.RHKP.....RLR.R....F..RN..K.F....D.TRVF...IK.LTRW.......K...E...S.RKLD.Q
      .INED.SR...Q..P.......Q.R..KHD.RHKR.....RLR.R....F..RK..K.F....D.TRVF...IK.LTRW..K....K...E...S.RKL..Q
1.375 45  8.32000E+04  4.00000E+04      5280       -800       9800
```

[a]The relative fitness of each individual. The selection for new mutations conserves core packaging, turns, helices and solvent accessibility in the same regions as in wild-type λ repressor. The user specified engineering parameters were a mutation preference for charged residues (D,E,R,N,Q,H,K) and parameter weights for helix, turn and accessibility of 1 and for amino acid preference and core packaging of 4 in the fitness function used for selection.
[b]The number of mutations.
[c]The absolute values calculated for each parameter are listed for the wild-type and for the end of each selection epoch after which a new selection trial is started.
[d]The N-terminal 102 residues of wild-type λ repressor which are engineered by the genetic algorithm using a start population of 40 612-bit strings, five epochs with 11 generations and a CPU time of 10 min on a VAX 3200 workstation.
[e]Wild-type residues are indicated by a dot and amino acid mutations by capital letters.
[f]Only the individuals which have a higher fitness than all individuals tested previously in that epoch are shown.
[g]High fitness individuals having critical substitution sites are marked.
[h]Final competition between the fittest from each epoch.

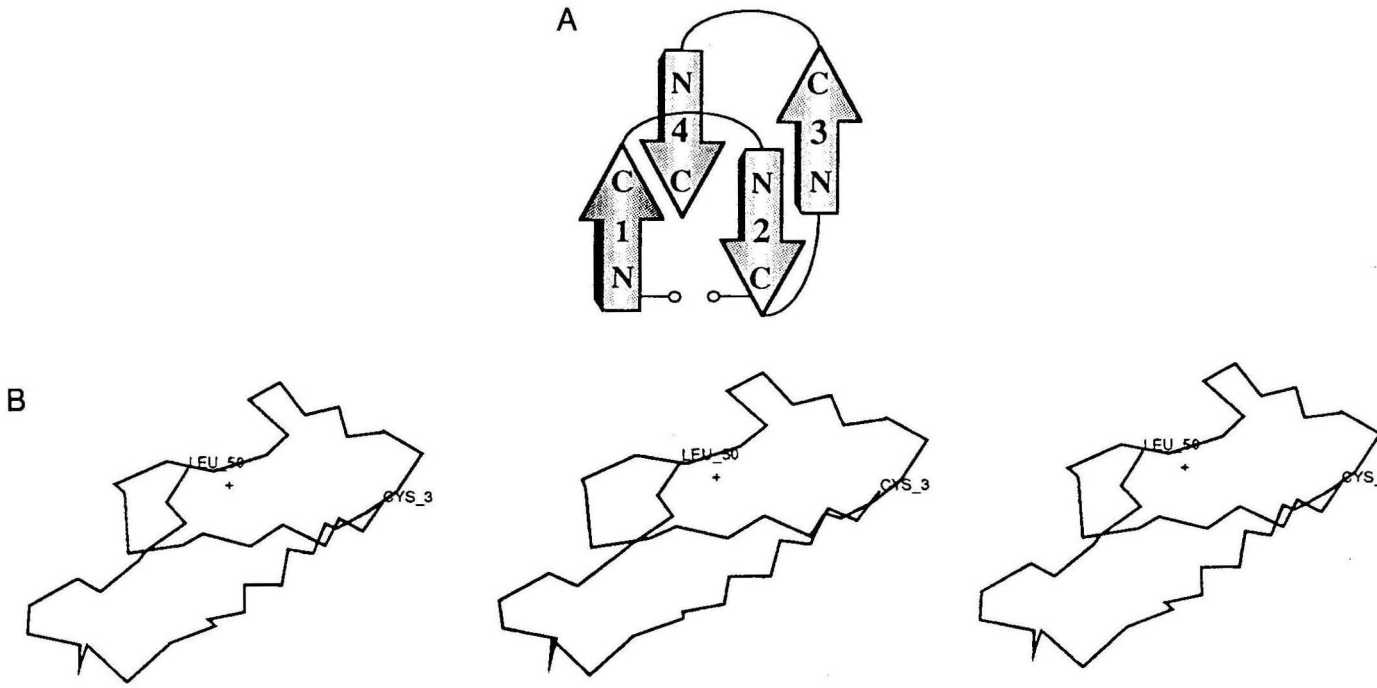

Fig. 1. (**A**) The four-stranded bundle investigated. Four units (1–4) of secondary structure (β strands) are connected at their termini (C,N) by three loops. There is no crossing over in the fold. An exemplary side chain interaction is sketched between the N-terminus of structural unit 1 and the C-terminus of unit 2. In the text, this would be designated as a $N_{uneven}C_{even}$ attractive force. (**B**) Three image stereo picture of the C_α trace for the first 50 N-terminal residues of azurin (Adman and Jensen, 1981; no coordinates for residues 1 and 2). The experimental structure is similar to that of the model shown in (**A**).

tertiary structure, constraints on the amino acid type and total side chain volume of seven critical core residues in the protein fold, and preferences for new amino acid exchanges defined by the user. For an engineering trial in designing a primary sequence with certain structural characteristics, the user may assign equal or different weights in the fitness function to these parameters and the amino acid preferences for new mutations. For instance, a higher weight may be given to the conserved core packaging or charged residues could be favoured as new mutations to investigate their influence on the function of λ repressor. The course of the genetic algorithm run starting from the wild-type λ repressor sequence is shown in Table II. Individuals with few mutation sites yet high fitness (marked in Table II) are of particular interest in actual mutagenesis trials; also mutations leading to drastic changes in fitness can also be tested experimentally. Selection by a fitness function where the sign of some of the fitness subterms has been changed (see Materials and methods) identifies particularly disruptive mutations (e.g. disrupting a helix and the core packaging) which are most useful to confirm protein structure interactions if ony partial information is available. As all other criteria of the fitness function remain optimized, the algorithm identifies test mutations for the presumed interaction. Additional parameters can easily be added to the fitness function as in other applications of genetic algorithms (e.g. Bickel and Bickel, 1990).

The next part of our study investigated the power of the genetic algorithm in a far more complex problem of *ab initio* protein folding from random conformations. For this the complex protein structure has again to be encoded in a string representing the 'genome' of an individual. This was achieved in a simplified way by using internal coordinates on a tetrahedral lattice (see Materials and methods). A fitness function judging the quality of the folded structure was established and should then mimic the important factors governing protein folding.

The ability of the genetic algorithm to discriminate between different forces in protein folding is critical for the evaluation of results by others on protein folding rules in their respective models (Skolnick *et al.*, 1989; Chan and Dill, 1991). Introducing proper folding parameters should result in a unique and plausible structure while unrealistic parameters would lead either to random, not observable structures or converge to different structures in different runs with different starts.

A simple model protein fold provided the trial (Figure 1A). It consisted of stretches of hydrophobic residues in β strand (B) configuration (zig-zag pattern) with interspersed loop regions (L) formed by residues with no distinct structural preference. No specific hydrogen bonding scheme was considered for the model. The hydrophobic residues should have a preference to be in a *trans* position in contrast to *cis*. The total protein had the formula $b_8L_3b_8L_3b_8L_3b_8$, resembling a β strand-rich protein. An approximation of the ideal fold can also be found in known tertiary protein structures; e.g. the first 50 residues of azurin (Adman and Jensen, 1981) as shown in Figure 1b or parts of a β roll in gene activator protein (Weber and Steitz, 1987). The model was also chosen as it is similar to those first investigated by the *ab initio* Monte Carlo simulations of Skolnick *et al.* (1988) and Chan and Dill (1990), and to provide a convenient and simple model to study folding of a protein in general.

How does this primary sequence fold into a three-dimensional structure? Are specific interactions to be included in the folding model or are, in contrast, more global, general forces important for the overall fold? Different fitness functions in our simulations tested this. A first fitness function (see Materials and methods) tried to enhance the probability of formation of a uniquely folded four-member antiparallel β bundle (Figure 1) by assuming important attractive (e.g. electrostatic) interactions between the strand ends such that the C-terminus of an even numbered β strand should be attracted to the N-terminus of an

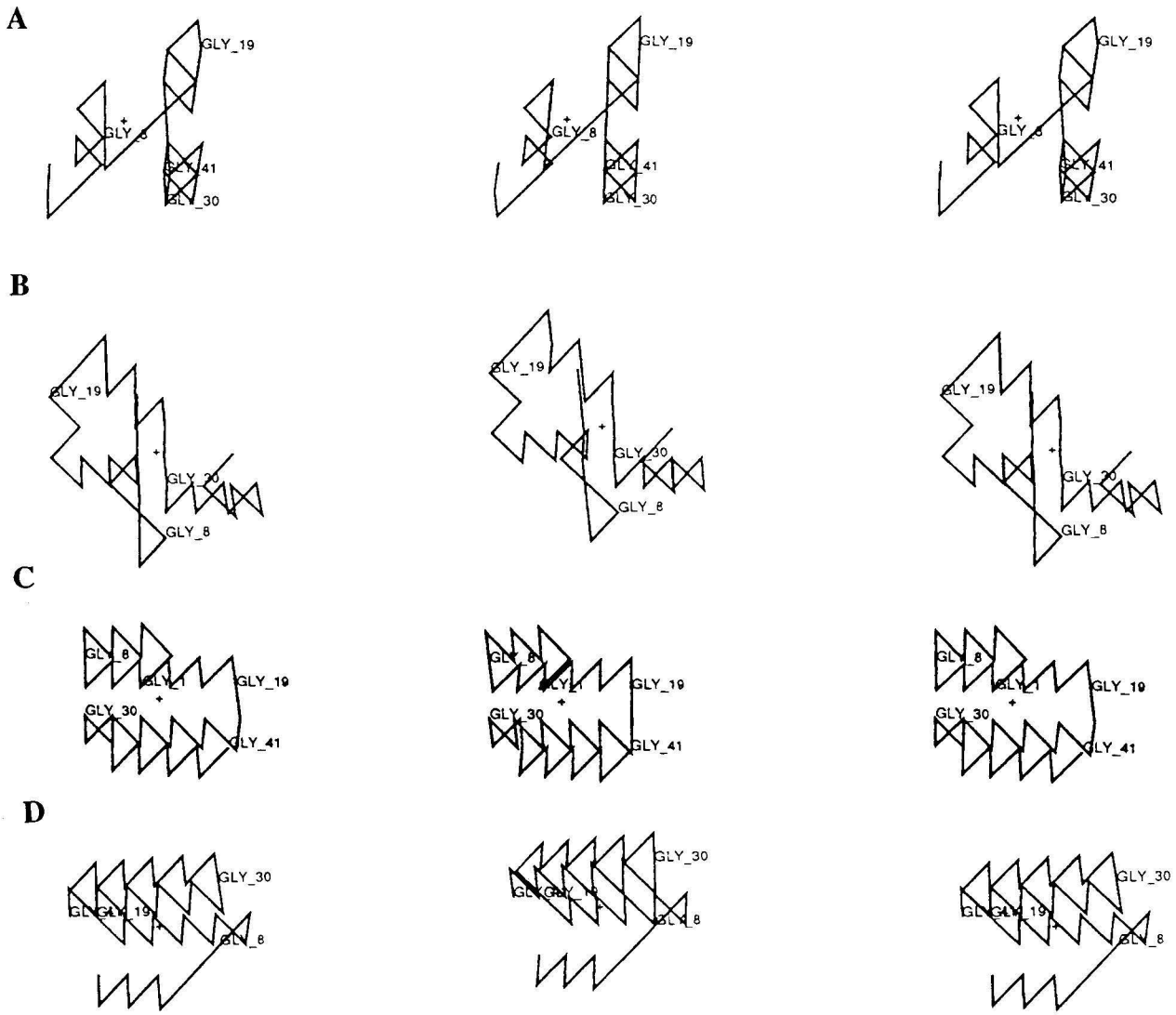

Fig. 2. Three image stereo picture of the *ab initio* folding of a four β strand-rich protein by the genetic algorithm. The program starts from random conformations (**A**) and more ordered structures arise by recombination, mutation and selection (**B**). The fitness function is based upon three residue loops (L3), eight-residue β strands (b8) with C_α atoms in a *trans* position and following a zig-zag pattern, and upon attractive forces between adjacent N- and C-termini. The last two structures (**C**) and (**D**) have adopted the secondary structure $b_8L_3b_8L_3b_8L_3b_8$ but have different topologies and represent some of the fittest individuals from various runs with different starting configurations. Residues are labelled as a glycine for visual facility in tracing the fold (Figure 1).

uneven numbered β strand and similarly the C-terminus of an uneven strand by the N-terminus of an even strand (Figure 1A). The simulation terminates in highly ordered structures and each of the β strands is completely formed with all the residues assuming *trans* configuration; however, the final fold observed is not always unique, a phenomenon which becomes even more prominent as the loop length is increased (Figure 2). Furthermore, in no case was the expected optimal fold (Figure 1) achieved.

Other parameters for governing the fold were then investigated (see also Discussion). The importance of site specific interactions were replaced by a different and more general condition: namely, the overall globularity of the protein. The scatter of the atom positions around their centre of mass was minimized in the fitness function. The scatter mimicks the preference of hydrophobic residues to be buried in the core of the protein fold. Driving the selection with the aid of this parameter leads in fact to a unique fold resembling a four-membered β bundle (Figure 1), the expected optimal solution. This fold (loop length = 3) is reproducible in different runs, independent of the random start configurations (Figure 3). As the final conformation is independent of the start it may be concluded that the complete conformational space is effectively searched (8 h run time on a VAX 3200 workstation in batch mode for the complete simulation).

The short loops which were not constrained structurally but were participants in the scatter fitness value were also significant in achieving the fold. For loop lengths 3–5 the bundle fold was maintained and was independent of the starting configuration; loop length 6 still achieved a compact fold. At loop length 7, not only was the bundle and considerable compactness lost, but different starts resulted in different final folds (Figure 4). No selection pressure was applied for or against the *cis* or *trans* position in the loop residues. For the larger loop lengths, which are close to those of the secondary structural elements, the non-ordered loop residues present more and more effort for the core forces and challenge the symmetry of the protein structure. In other trials, the scatter part of the fitness function was applied only to the β strand residues as the structureless loop residues

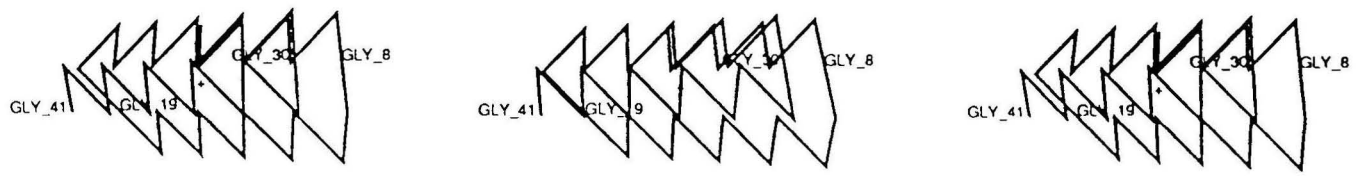

Fig. 3. Stereo picture (three images) of a typical fit individual for a four β strand bundle structure (Figure 1) using a fitness function based upon a three residue loop structure, eight-residue β strands ($b_8L_3b_8L_3b_8L_3b_8$), and a minimum distance of all atoms from the molecular centre of mass, the latter mimicking hydrophobic packing forces.

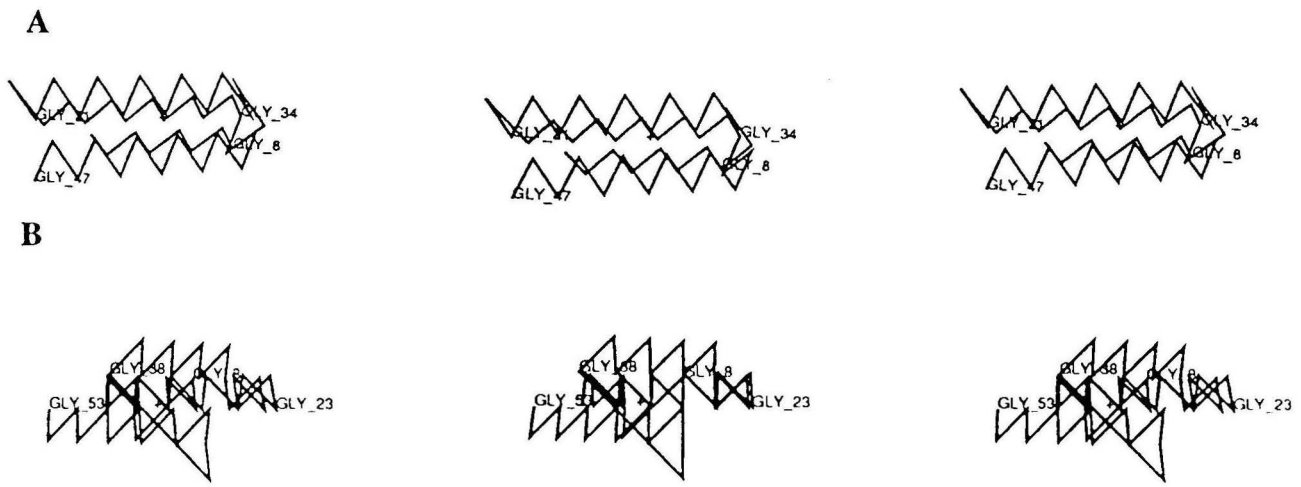

Fig. 4. As in Figure 3 but using different loop lengths L: (**A**) $b_8L_5b_8L_5b_8L_5b_8$, (**B**) $b_8L_7b_8L_7b_8L_1b_8$. The loop residues were included in the scatter fitness function. It is clear that the topology of the antiparallel four-strand bundle (Figure 1) is not maintained if the loop length increases above 5.

may have interfered with convergence to optimality, especially as the loops became larger. However, similar results were found as in the cases above where loop atoms were included in the compactness term.

In all of the protein folding experiments for a given set of parameter constraints in the fitness function, the best individuals from 24 'epochs' of different start populations with random lattice coordinates competed in a final selection run against a random background. The number of epochs was chosen as it resulted in the expected optimal fold under at least one set of fitness conditions. For each fitness function at least 10 complete simulation runs of 24 epochs each were tested.

Discussion

Finding the fitness function

The selection driven genetic algorithms are very efficient search tools covering vast conformational space. They can be used to optimize simultaneously several parameters. However, the direction of the selection is critical for achieving a good and realistic model. The fitness functions for different simulations must be carefully chosen and tested given the parameters to be modelled and their respective weights. A good guideline is to keep the model as simple as possible, introduce and change only one parameter at a time and maintain a minimum number of fitness parameters. The various weights for the parameters were adjusted in the beginning with the most significant features similarly weighted. However, if a fitness characteristic was crucial in the early generations, its weight was increased decisively. For instance, a high negative clash weight was essential to build realistic structures without any atom overlap.

Genetic algorithms offer the specific advantage that they are robust search routines. Especially the introduction of crossover allows high fitness islands to be reached which would not be attainable by using mutation alone (Spears and De Jong, 1991). The fitness parameters and weights for the examples given here were determined empirically by test runs as the problem space including all possible parameter conditions is rather large (Goldberg, 1989). Thus the advantage of genetic algorithms in competition with other methods (e.g. Monte Carlo searches) depends on how easily an effective fitness function can be tailored for the application.

The ratios between the different parameter weights determine the selection outcome. Thus in the first simulation the very high weight on Aastop allowed counterselection against stop codons in the earliest generations. The higher weight on consensus matches than amino acid differences leads to two stages of evolution in the example and was explicitly chosen. First, the general zinc finger binding consensus is approached and heavily selected for. Only when this is nearly reached is the amino acid composition optimized according to the known average. A two stage selection procedure by genetic algorithms allows the possibility that the initial high-weight parameter is the first to reach a near optimal state. This evolution simulation tested the ability of the genetic algorithm to search efficiently a vast combinatorial space and delineate quickly the expected answer, namely, the zinc finger consensus. Negative controls illustrated conditions where the algorithm failed showing the example to be non-trivial. In contrast, the engineering example developed a known wild-type sequence further. The engineering direction is open for the user by placing weights on preferred parameters.

Conclusions from the folding simulation

In more complex simulations like protein folding, the choice of a proper model (and corresponding string coding) in which the

genetic algorithm performs well is more complex. For instance, simulations on the tetrahedral lattice proved to be superior to those based on a cubic lattice (three bits per chain atom). Chan and Dill (1991) in *ab initio* protein folding trials conclude from their exhaustive configurational enumerations on small lattices that formation of secondary structure is inherently probable but that the unique native fold is rare and difficult to find. The genetic algorithm selection used here found the optimal fold and avoided an exhaustive enumeration of all states which are of the order of 10^{24}. Though the model had only the secondary structural propensities of its residues predefined, the ability of the algorithm to find the unique three-dimensional fold and the effectiveness of various forces in folding (e.g. hydrophobicity) or structural stabilization (e.g. cooperativity) were tested in the simulations.

The fitness functions described in the results have been optimized from many trial runs. Specifically in the folding example, a very low value of c_1 (positive fitness parameter) had only very few individuals surviving the first generation, significantly reducing variants for further selection. A very high value for c_1 diminished the fitness differences between individuals and slowed evolution. Too large a value for the clash weight prevented formation of occasional turns or disallowed internal movements in the protein structure. The clash weight chosen allowed maximum flexibility of the protein chain as it was the smallest value which just deleted all clashes from later generations. The introduction of the betarow factor in the fitness function, allowing for growth of regions which already had β conformations, proved to be critical in achieving extended structure in β conformation. A very high value left clashes or lead to a loss of an overall compact three-dimensional fold. The β strand start weight and the increment chosen for betarow represented a balance between promoting nucleating regions for β structure and rewarding strand outgrowth.

The basis for secondary structure formation has been suggested to be core packaging rather than hydrogen bonds or specific interactions (cf. Chan and Dill, 1990). This idea was tested in our model. The pairwise packing of the β strands is a result of simple selection for low scattering around the centre of mass, tantamount to attractive hydrophobic forces acting on the core residues. Rather non-specific, global forces may be more important for protein folding than previously anticipated (Hughson *et al.*, 1991; Jeng and Englander, 1991). Moreover, our results show that the reliance on specific interactions at the ends of β strands was not able to achieve a unique fold and the ideal β bundle model. This is also consistent with *ab initio* protein folding studies by Skolnick and coworkers who found that site specific interactions are only involved in fine tuning the fold (Skolnick *et al.*, 1988). In comparison with electrostatic interactions and overall hydrophobicity as governing parameters for the overall fold, other parameters tested proved to be far less effective in the simulations. In particular, neither a strongly selected (already in the earliest generations) hydrophobic collapse of the centres of the β strand regions nor different selections for parallelity or close vicinity of the β strands proved to be effective. In this way different parameters could be inspected in the model for their relative power to drive a protein fold.

Loops may play an important but indirect role in producing a unique native state (Skolnick *et al.*, 1989). The effect of loop length was tested in our model by varying systematically the length of the three interconnecting loops. It was observed that the optimal fold was no longer reached above a certain threshold length which was near the length of the secondary structural elements. Even when the compactness criterion was applied only to the strand residues, the relatively long loops continued to interfere with optimal folding. Though core hydrophobic packing primarily directs the fold, long loops can be destabilizing unless internally stabilized.

Perspectives

In contrast to typical Monte Carlo simulations, the model fold presented here was found by a different type of search and perspective. The calculation requirements of the genetic algorithm (24 epochs of 120 generations for an entire four-strand simulation) compares favourably with a Monte Carlo (MC) simulation (3×10^6 MC cycles per temperature tested; Skolnick *et al.*, 1988). The genetic algorithm folding analysis answered directly how stable and unique protein structures are achieved under an evolutionary type of selection. The general conclusions of Skolnick *et al.* (at least for their start models, 1988, 1989) and Chan and Dill (1990, 1991) were verified by the genetic algorithm approach, which also allowed the study of cooperativity in the formation of β strand regions, the effect of variance in loop length and the significance of specific interactions in the folding process.

We wish to call attention to the wide potential application of genetic algorithms in the study of proteins. Possible areas of interest are illustrated by examples from protein evolution, engineering, design and folding. We intend to explore further important forces for protein folding by using fitness functions with various residue physicochemical characteristics as preferred side chain−side chain interactions, hydrogen bonds, size and shape. Simulations with α helical structures are in progress. Independence of the lattice, allowing general atom positions, will also be explored to avoid any bias (Gregoret and Cohen, 1991). Secondary structure predictions can be exploited to enhance the power of the genetic algorithm in guiding tertiary structure folding. Complementation with other prediction models is also possible where a pregiven set of solutions can be refined by simply including them in the start population (see Materials and methods, 'epochs'). The ability of genetic algorithms to search vast conformational spaces in parallel with a realistic fitness function represents a potential which awaits further exploitation for problems in protein structure.

Acknowledgements

T.D. gratefully recognizes the financial support of Deutsche Forschungsgemeinschaft in the form of a postdoctoral fellowship.

References

Adman,E.T. and Jensen,L.H. (1981) *Isr. J. Chem.*, **21**, 8−12.
Bickel,A.S. and Bickel,R.W. (1990) *Comput. Biol. Med.*, **20**, 1−3.
Chan,H.S. and Dill,K.A. (1991) *Annu. Rev. Biophys. Biophys. Chem.*, **20**, 447−576.
Chan,H.S. and Dill,K.A. (1990) *Proc. Natl Acad. Sci. USA*, **87**, 6388−6392.
Chou,P.Y. and Fasman,G.D. (1978) *Adv. Enzymol.*, **47**, 45−148.
Gregoret,L.M. and Cohen,F.E. (1991) *J. Mol. Biol.*, **219**, 109−122.
Gibson,T.J., Postma,J.P.M., Brown,R.S. and Argos,P. (1988) *Protein Engng*, **2**, 209−218.
Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, Reading, MA.
Hughson,F.M., Barrick,D. and Baldwin,R.L. (1991) *Biochemistry*, **30**, 4113−4118.
Janin,J., Wodak,S., Levitt,M. and Maigret,B. (1978) *J. Mol. Biol.*, **125**, 357−386.
Jeng,M.-F. and Englander,S.W. (1991) *J. Mol. Biol.*, **221**, 1045−1061.
Kabsch,W. and Sander,C. (1983) *Biopolymers*, **22**, 2577−2637.
Lim,W.A. and Sauer,R.T. (1989) *Nature*, **339**, 31−36.

Pabo,C.O. and Lewis,M. (1982) *Nature*, **298**, 443–447.
Skolnick,J., Kolinski,A. and Yaris,R. (1988) *Proc. Natl Acad. Sci. USA*, **85**, 5057–5061.
Skolnick,J., Kolinski,A. and Yaris,R. (1989) *Proc. Natl Acad. Sci. USA*, **86**, 1229–1233.
Spears,W. and De Jong,K.A. (1991) In Belew,R. and Booker,L. (eds), *Proceedings of the Fourth International Conference of Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, pp. 230–236.
Weber,I.T. and Steitz,T.A. (1987) *J. Mol. Biol.*, **198**, 311–326.

Received on May 20, 1992; revised on July 20, 1992; accepted on July 23, 1992