Protein Engineering vol.8 no.12 pp.1205–1213, 1995

Designing amino acid sequences to fold with good hydrophobic cores

Shaojian Sun[1], Rachel Brem[2], Hue Sun Chan and Ken A.Dill

Department of Pharmaceutical Chemistry, Box 1204 and [2]Graduate Group in Biophysics, Box 0448, University of California, San Francisco, CA 94143-1204, USA

[1]Present address: NCI-FCRF, Frederick Biomedical Supercomputing Center, PO Box B, Frederick, MD 21702, USA

We present two methods for designing amino acid sequences of proteins that will fold to have good hydrophobic cores. Given the coordinates of the desired target protein or polymer structure, the methods generate sequences of hydrophobic (H) and polar (P) monomers that are intended to fold to these structures. One method designs hydrophobic inside, polar outside; the other minimizes an energy function in a sequence evolution process. The sequences generated by these methods agree at the level of 60–80% of the sequence positions in 20 proteins in the Protein Data Bank. A major challenge in protein design is to create sequences that can fold uniquely, i.e. to a single conformation rather than to many. While an earlier lattice-based sequence evolution method was shown not to design unique folders, our method generates unique folders in lattice model tests. These methods may also be useful in designing other types of foldable polymer not based on amino acids.

Keywords: evolutionary search method/HP sequence/inverse folding

Introduction

We have developed inverse folding algorithms with the aim of designing amino acid sequences which fold to desired 'target' structures. Recently, Kamtekar *et al.* (1993) have shown that a simple binary sequence code of hydrophobic (H) and polar (P) amino acids may define native chain topologies in proteins. Kamtekar *et al.* (1993) started with the native coordinates of a known biological four-helix bundle, and then designed sequences to fold to it. Their design strategy was simply to bury H monomers and expose P monomers. Here, we introduce two automated procedures for performing such H/P designs. The first is a direct transcription of the procedure of Kamtekar *et al.* (1993): it assigns an H monomer to any solvent-inaccessible position in the given structure, and P otherwise. We have called it the Burial algorithm. It simply codifies the standard lore that hydrophobics should be buried, and serves just as a reference for our second method. The second method is a variant of the 'evolutionary design' procedure of Shakhnovich and Gutin (1993). Following statistical mechanical terminology, we call our second method the Grand Canonical Sequence Evolution (GCSE) method, in contrast to the canonical method of Shakhnovich and Gutin (1993).

We have performed both types of design on the X-ray structures of 20 known proteins from the Protein Data Bank (PDB). Then we tested the GCSE method in a simplified lattice representation of the proteins to demonstrate why we believe that our designed sequences will fold to target structures and to no others. When we compared the sequences designed by these algorithms with the known native sequences of biological proteins, the agreement ranged from 57 to 81%. Our methods may be useful tools for designing monomer sequences that will fold to specific target conformations of proteins and other polymers.

Inverse folding (Drexler, 1981; Ponder and Richards, 1987; Bowie *et al.*, 1991; Lee and Levitt, 1991; Yue and Dill, 1992; Shakhnovich and Gutin, 1993; Shakhnovich, 1994; Gutin *et al.*, 1995) can be thought of as a 'bead coloring' problem. Start with the desired coordinates of a chain of 'colorless beads', i.e. generic amino acids that do not yet have side-chain structural identities. The design process then 'paints' each bead a 'color', representing each of the 20 amino acids. Any such design must solve two problems (Yue and Dill, 1992): (i) positive design — it must generate sequences that will fold to the given structure; and (ii) negative design — it must design sequences that will not fold to other low energy structures. Here we have shown that one of our algorithms provides a good negative design in lattice model tests.

Materials and methods

Our algorithms were given the coordinates of a desired target structure. Because our aim was to test our methods, the coordinates we used were those of known proteins from the PDB. Thus we 'invented' sequences for some known protein structures and compared them with the natural sequences. Of course, it would have defeated the purpose if we had input the all-atom side-chain coordinates for the target structure because then any good sequence design method would have just recreated the natural sequence. Instead we asked 'Can we design just the hydrophobic and polar parts of the sequence?' To solve such a problem, when we input the coordinates of the target structure we had to represent every side chain identically, so that they were 'colorless'. Hence we input coordinates for the target structure we in the following way. We retained all backbone atoms but replaced side chains with a centroid, or spherical virtual atom. Side-chain centroids were placed along the native C_α–C_β bond vector at a uniform distance of 3.0 Å from the C_α, and had a radius of 2.0 Å. For residue positions occupied by glycine in the target structure, no native C_α–C_β bond vector was available, so we used a standard rotamer direction (bond angle 107.63°, dihedral angle 239.11°) for placing our centroid. Our aim was to design HP sequences that would fold to roughly these target conformations.

The Burial algorithm

In this method, we used the given coordinates of the target structure to determine the degree of burial of each amino acid, and then assigned H to the buried positions and P to the exposed sites. Each amino acid in the target structure was defined as exposed or buried according to whether its side-

chain bead had > or <10 Å^2 of exposed surface area, respectively. A residue was 'painted' H if it was buried by this definition, and was painted P otherwise. This algorithm was fast because it depended only linearly on the chain length of the target structure. We designed sequences for 20 proteins using the Burial algorithm. Table I shows the fraction of sequence positions for which the Burial algorithm assignment of H or P agreed with the polar/nonpolar identity of the residue in the true native structure, where hydrophobic = A, C, I, L, M, F, W, Y and V, and polar = R, N, D, E, Q, G, H, K, P, S, and T in the one-letter code of amino acids. The Burial algorithm designed sequences that were identical to the natural sequences (in terms of hydrophobic and polar assignment) at ~60–80% of the residue positions. This was consistent with the many earlier studies indicating that the cores of globular proteins are predominantly hydrophobic and the surfaces are extensively polar (Chothia, 1976; Wertz and Scheraga, 1978; Janin, 1979; Meirovitch and Scheraga, 1980; Guy, 1985).

The Sequence Evolution algorithm

We also developed another bead painting algorithm. Following earlier work on the inverse folding problem (Ponder and Richards, 1987; Lee and Levitt, 1991; Shakhnovich and Gutin, 1993; Shakhnovich, 1994; Gutin *et al.*, 1995), our second method performed positive design by the minimization of a conformational energy function through sequence mutations. Our approach differed from earlier works in the following respects. (i) Rather than using 20 amino acid types (Ponder and Richards, 1987; Lee and Levitt, 1991; Shakhnovich, 1994), we considered just the two monomer types, H and P. In this way we could explore the design of other possible foldable polymers, in addition to proteins. We tested the method on lattice model examples, for which the true global minimum in free energy was known in both sequence and conformational spaces. (ii) We did not fix the amino acid composition. In analogy with statistical mechanics, we called our approach a 'grand canonical' method, which allowed a variable composition, in contrast to the 'canonical' method of Shakhnovich and Gutin (1993) which requires a fixed composition. Shakhnovich and Gutin (1993) recognized that when using the HP potential alone, the sequence evolution methods converged to the homopolymer of all hydrophobic units, because that gave the lowest achievable energy. To avoid this problem, they fixed the sequence composition. Theirs is a 'bead swapping' process which iteratively reassigns H and P monomers to different positions. The total number of H monomers and the total number of P monomers remain unchanged in their method. In the canonical method, it is necessary to know in advance the correct H and P compositions. The use of the wrong HP composition can lead to sequence designs that do not fold uniquely (Yue *et al.*, 1995). In contrast to their canonical bead swapping method, ours involves bead painting and explores all possible HP compositions. It does not converge to homopolymer sequences because of a term we have added to the energy function, justified below. We show below that our method often produces sequences that fold uniquely to the desired native structure, when tested in lattice models. (iii) Rather than a single sequence evolution Monte Carlo search method of the type used by Shakhnovich and co-workers (Shakhnovich and Gutin, 1993; Shakhnovich, 1994; Gutin *et al.*, 1995), we used a genetic algorithm to search sequence space which created many different sequences simultaneously. This provided useful statistics. (iv) Unlike the lattice model methods (Yue and Dill, 1992; Shakhnovich, 1994; Gutin *et al.*, 1995), our method allowed real-space coordinates and so could be designed and tested on real molecules.

Table I. PDB protein names

Protein	Chain length	Percent identity (%) with natural sequences of HP sequences designed by:	
		Burial algorithm (cut-off = 10 Å)	GCSE algorithm
1aaj	105	66	66
1aba	87	81	81
1aps	98	76	72
1arr monomer	53	60	62
1arr dimer	106	70	73
1bba	36	61	58
1bbl	37	62	68
1bov	69	67	74
1brq	174	71	68
1cis	66	67	64
1cmb monomer	104	59	62
1cmb dimer	208	68	70
1hel	129	78	74
1ifb	131	69	76
1kba monomer	66	71	72
1kba dimer	132	80	73
2gbl	56	77	80
2hpr	87	78	78
2il8	71	73	77
256b	106	79	81
3cln	143	68	62
3rn3	124	71	81
3trx	105	77	80
Average		72	73

'Monomer' and 'dimer' refer to sequence design calculations performed using the monomer structure alone as the target and the full dimer structure as the target, respectively.

We assumed a very simple fitness function. The 20 amino acids were divided into two classes: H, for hydrophobic, and P for polar or other. The sequence fitness energy function is:

$$E = \varepsilon \sum_{i<j}^{N} h_{ij} + \sigma \sum_{j}^{N} s_j, \qquad (1)$$

where $\varepsilon = -2$ and $\sigma = +1$. Here i and j are indices for the positions in the sequence of two monomers. The first term is negative (favorable) if both the ith residue and the jth ($\neq i$) are hydrophobic and are in nonlocal contact (here nonlocal means $|i-j| > 2$). In full-space studies, we define the distance-dependent function

$$h_{ij} = h_{ij}(r_{ij}) = \frac{1.0}{1.0 + e^{(r_{ij} - 6.5)/1.0}}, \qquad (2)$$

with the inter-residue distance r_{ij} between hydrophobics in Å. This function is sigmoidal, representing more softness in the potential than a step function. The cut-off distance of the sigmoidal function is 6.5 Å. In the lattice model test of our methods, we set $h_{ij} = 1$ when two contacting lattice residues are hydrophobic, and $h_{ij} = 0$ otherwise. The second term in Eq. (1) is the hydrophobic residue-solvent interaction, representing an additional tendency for avoidance of contacts between solvent and the hydrophobic monomers. It is positive (unfavorable) if residue j is hydrophobic and contacts a solvent site, in which case $s_j = 1$; otherwise $s_j = 0$. In the full-space model, the sum in the second term of Eq. (1) is over solvent

(A) 1hel

```
1                                                                              80
KVFGRCELAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGSTDYGILQINSRWWCNDGRTPGSRNLCNIPC
HPHPHHPHHPHHHPHPHHPHPPHHHHHHHHHHHPHHHPHPHPHPPPHPHPHHHHHHHHHHHHPHHPHPPHPPHPHPHPHPH
HPHPHPPHHPHHPPHPHPPHPPHHHHHHHHHHHPHHHPHPHPHPPPHPHPHHHHHHHHHHHHPHHPHPPHPPHPHPHPHPH

81
SALLSSDITASVNCAKKIVSDGNGMNAWVAWRNRCKGTDVQAWIRGCRL
PHHHPPPHPHHHPHHHPHPPPPPHPHHPHHPPPHPPHPHPPHHPPHPH
PHHHPPPHPHHHPHHHPHPPPPPHPHHPHHPPPHPPHPHPPHHPPHPH
```

(B) 3rn3

```
1                                                                              80
KETAAAKFERQHMDSSTSAASSSNYCNQMMKSRNLTKDRCKPVNTFVHESLADVQAVCSQKNVACKNGQTNCYQSYSTMS
PPPPHPPHPPHHHHPPHPPHPPPPHHHPHHPPHPHHPPPHHPPHHHHHHPHPPHHPHHPHPPHPHPHPHPHHHHHPPPHH
PPPPHPPHPPHHHHPPHPPHPPPPHHHPHHPPHPHHPPPHHPPHHHHHHPHPPHHPHHPHPPHPHPHPHPHHHHHPPPHH

81
ITDCRETGSSKYPNCAYKTTQANKHIIVACEGNPYVPVHFDASV
HHHHPHHPPHPPPPPHHHPHPHHPHHHHHHHHPPPHHHHPHHHHH
HHHHPHHPPHPPPPPHHHPHPHHPHHHHHHHHPPPPPHHHPHHHHH
```

(C) 256b

```
1                                                                              80
ADLEDNMETLNDNLKVIEKADNAAQVKDALTKMRAAALDAQKATPPKLEDKSPDSPEMKDFRHGFDILVGQIDDALKLAN
HPHPPHHPHHPPHHPPHPPHPPHPPHHPHHPPHHPHHPHHHPHPHPPHPPPPPPPHPPHPPHPPHHPPHHPHHPPHHPHHP
HPHPPHHPHHPPHHPPHPPHPPHPPHHPHHPPHHPHHPHHPPHPHPPHPPPPPPPHPPHPPHPPHHPPHHPHHPPHHPHHP

81
EGKVKEAQAAAEQLKTTRNAYHQKYR
PPPHPPHHPHHPPHPPHPPPHPPPHP
PPPHPPHHPHHPPHPPHPPPHPPPHP
```

(D) 1aba

```
1                                                                              80
MFKVYGYDSNIHKCGPCDNAKRLLTVKKQPFEFINIMPEKGVFDDEKIAELLTKLGRDTQIGLTMPQVFAPDGSHIGGFD
PHPHHHHPHPHPPHPPHPPHHPHHPPPPPPHPHPHHHPPPPPPPPPPPPPPHPPHPPPPPPPPHPHHHHHHPPPPHHPPHP
PHPHHHHPHPHPPHPPHPPHHPHHPPPPPPHPHPHHHPPPPPPPPPPPPPPHPPHPPPPPPPPHPHHHHHHPPPPHHPPHP

81
QLREYFK
PHPPPHP
PHPPHHP
```

Fig. 1. Natural sequences and the two best (lowest energy) GCSE-designed sequences for (**A**) lysozyme (PDB identifier 1hel), (**B**) RNase A (3rn3), (**C**) cytochrome *b*562 (256b) and (**D**) a glutaredoxin mutant (1aba). H = hydrophobic; P = polar.

contacts. We used the program ACCESS (Richmond, 1984) to define solvent accessibilities of the amino acids. We assumed one solvent contact for every 3Å^2 of surface area exposed of a sidechain bead, and we used the default 1.4 Å probe. In the lattice representation, the sum in the second term of Eq. (1) was over all hydrophobic residue faces not involved in contacts.

We explored sequence space with a genetic algorithm similar to that used in folding algorithms (Dandekar and Argos, 1992; Sun, 1993). It comprises several steps (Sun, 1992). (i) Randomly create a population of HP sequences of a given length. (ii) Apply genetic operations to this initial sequence population to generate ~10-fold more sequences. (iii) Use the fitness function to select the fittest (lowest energy) among these new sequences. (iv) Iterate through steps (ii) and (iii) until the energy function reaches a minimum and sequences stop improving. This differs from the Monte Carlo sequence design method (Shakhnovich and Gutin, 1993; Shakhnovich, 1994; Gutin *et al.*, 1995) in which the search operates only on a single evolving sequence. The genetic operations used were the following.

Replication. Copy all p fittest sequences of the last generation into the next generation. Thus p represented the replication population.

Mutation. Each of the p fittest sequences of the last generation was mutated and then copied into the next generation. The mutation sites were chosen randomly within a sequence, and we allowed a maximum of three point mutations, i.e. H to P or P to H, per sequence. In the mutation operation, we also allowed two monomers to exchange their sequence positions (permutation). The mutation population was $2p$.

Crossover. There are different ways to perform crossover. Here we used random monogamy: the crossover operation randomly selects two sequences and cuts them at a random site, then crosses over and pastes in the parts to generate two new sequences for the next generation. The crossover sequences

were created randomly between the replication and mutation sequences. Therefore the total number of new sequences generated by the crossover operation was $2p$. Although crossover could be allowed at multiple sites, we used only one crossover site for any pair of sequences.

Shuffle. The shuffle operation interchanges two pieces of the chain from one place in the sequence to another. It conserves composition. We randomly chose the lengths (up to a maximum of eight residues) and starting positions of the shuffled pieces. To shuffle segments at the ends of sequences, we used a periodic boundary condition. Hence if a segment four residues long was to be shuffled, and it began at the protein's last residue, the final segment to be shuffled would include residues 1, 2 and 3. The shuffled sequences were generated from the p fittest sequences of the last generation. The shuffle population was p.

Reverse shuffle. Reverse shuffle is identical to the shuffle operation, except that the sequence directionality is changed. Sequence piece *ABCD...* becomes *...DCBA* at a different sequence position. The reverse shuffle population was p. Reverse genetic shuffles are not biological, but for computer algorithms like ours, this was not important.

We used an initial sequence population size of $p = 500$. All the initial sequences were created randomly: at each position, H or P was initially chosen with equal probability. After the replication, mutation, crossover, shuffle and reverse shuffle operations, $7p$ sequences were created from the fittest p sequences of the last generation, and their fitness values were computed according to the energy function. To avoid possible premature convergence in the sequence space, a share mechanism (Goldberg, 1989; Sun, 1993) was applied in the sequence selection process. The number of identical sequences in the selected population was restricted to no more than two. All newly created sequences were sorted according to their energy values. The energies of the sequences usually converged in ~120–180 generations. Converged sequences were ~95% identical among themselves. For targets >~60 residues, we often found that >50% of the final sequences had fitness values within 1–5% of the lowest energy.

Results and discussion

Designing protein sequences

We used 20 proteins from the PDB as target structures. We designed an ensemble of HP sequences for each target structure and then compared our invented sequences with the natural sequences, translated to the HP code. Table I compares our designs from the GCSE and Burial algorithms with the natural sequences. Figure 1 gives the lowest energy sequences from the GCSE algorithm for several example proteins. Table II correlates the average success and failure at a given residue with its exposure to solvent. In sequences designed by either the Burial or the GCSE algorithm, when Ps are assigned to exposed positions, 75–80% agree with the native sequence at that position, and 67–69% of Hs assigned to buried positions agree with the native sequence. These data illustrate the extent to which the 'hydrophobic inside, polar outside' principle is borne out in our test set of proteins. Table II also shows that while the GCSE algorithm seldom assigns Ps to buried positions (only 2.8% of all assigned Ps are buried), it often assigns Hs to exposed positions (25.4% of all assigned Hs are exposed). Hence, the physics described by GCSE is different from that of the Burial algorithm. Using GCSE, the success rates varied from 58 to 81%, averaging ~73% among the biologically relevant forms of the proteins tested. The highest success rates were, not surprisingly, for the most globular proteins having the most buried hydrophobic cores.

Should we expect better? It is not clear. Presumably nature designs sequences not just for structure and stability, but also for function and perhaps to optimize folding kinetics. Hence, even if we designed proteins using nature's true potential function, we might not achieve much better similarities to natural proteins. Moreover, natural proteins may not be optimal. For example, stability can be improved when buried polar groups are replaced by hydrophobic residues (Schildbach *et al.*, 1995) or when exposed hydrophobic residues are made polar (Pakula and Sauer, 1990). Hence, some buried polar groups in proteins and some exposed nonpolar groups may be accidental. It is worth noting the possibility that the GCSE method may cover sequence space more broadly, and may design sequences that are more stable in their native structures, than nature does.

Comparing designed and natural sequences. Figure 2 compares the native and designed sequences for six proteins, threaded on to their corresponding structures. The GCSE algorithm designs good hydrophobic cores and polar surfaces. Figure 2A–F shows that the discrepancies with natural sequences are caused by exposed hydrophobic and buried polar residues. Figure 2G and H shows two examples of more global errors in the GCSE assignments. In a glutaredoxin mutant (1aba), the algorithm creates a hydrophobic 'core' even at a protrusion near the surface of a protein, where there are multiple residue contacts. Despite these errors, the designed sequence for 1aba was 81% identical to the native sequence (Table I).

Table II. Breakdown of average sequence assignment by degree of exposure and by residue type

	Buried position		Exposed position	
	assignment correct (%)	assignment incorrect (%)	assignment correct (%)	assignment incorrect (%)
GCSE-designed				
Total assigned Hs	51.6	23.0	12.8	12.6
Total assigned Ps	2.1	0.7	77.9	19.3
Burial-designed				
Total assigned Hs	67.4	32.6	0.0	0.0
Total assigned Ps	0.0	0.0	75.1	24.9

Results were averaged over all 20 naturally occurring proteins used in this study (see Table I). Here, 'Buried position' indicates that either an H or a P residue is assigned at a position in the structure with <10 Å^2 of exposed surface area. Likewise, exposed residues are defined here to have >10 Å^2 of exposed surface area. An assignment is 'correct' if it agrees with the native sequence, i.e. if the assigned residue is H, the native sequence has an H at that position.

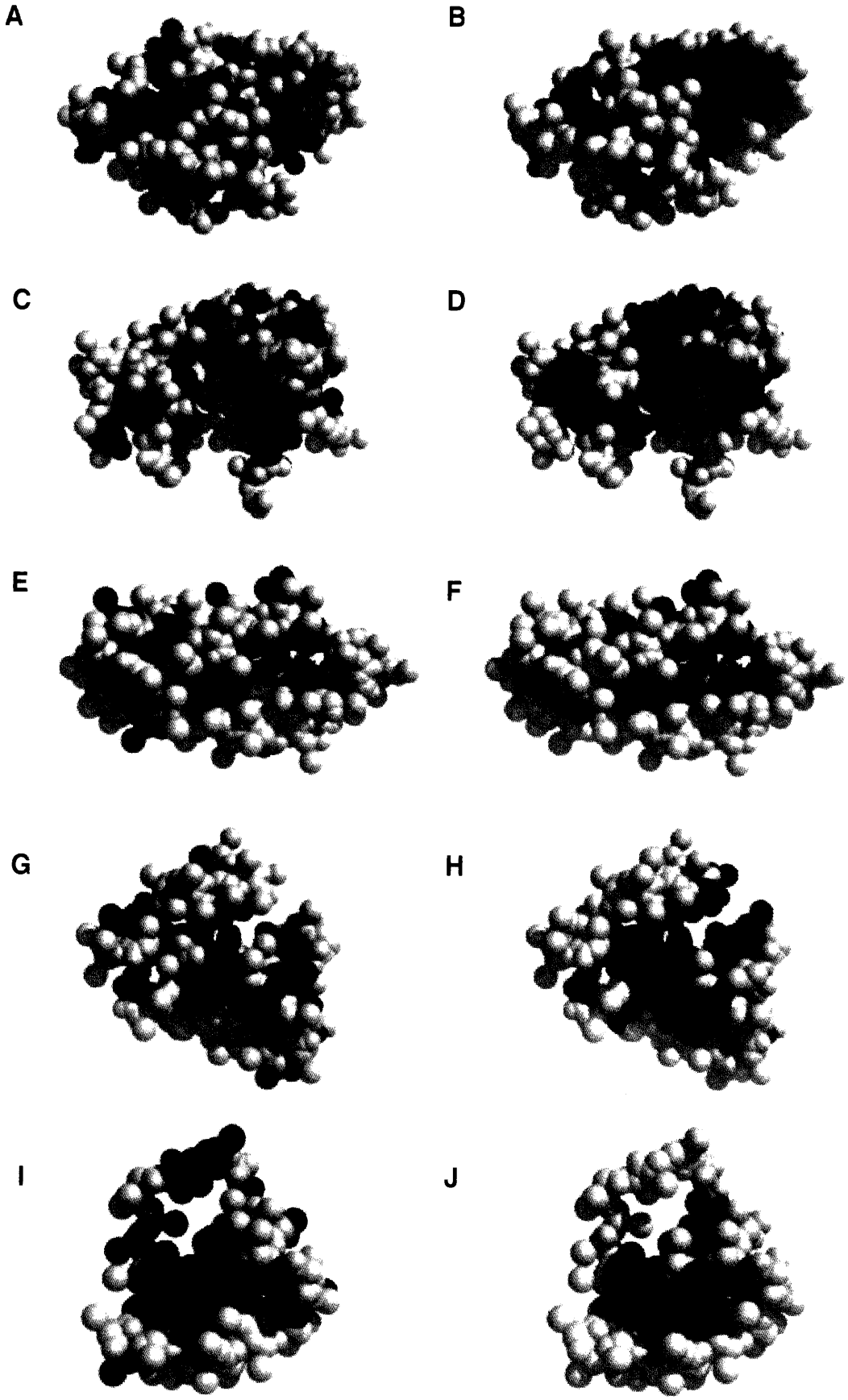

Fig. 2. Native (left) and GCSE-designed sequences (right) superimposed on the native structures. Black, hydrophobic; white, polar. (**A** and **B**) lysozyme (1hel), (**C** and **D**), RNase A (3rn3), (**E** and **F**) cytochrome *b*562 (256b), (**G** and **H**) glutaredoxin mutant (1aba), (**I** and **J**) subtilisin-chymotrypsin inhibitor (1cis).

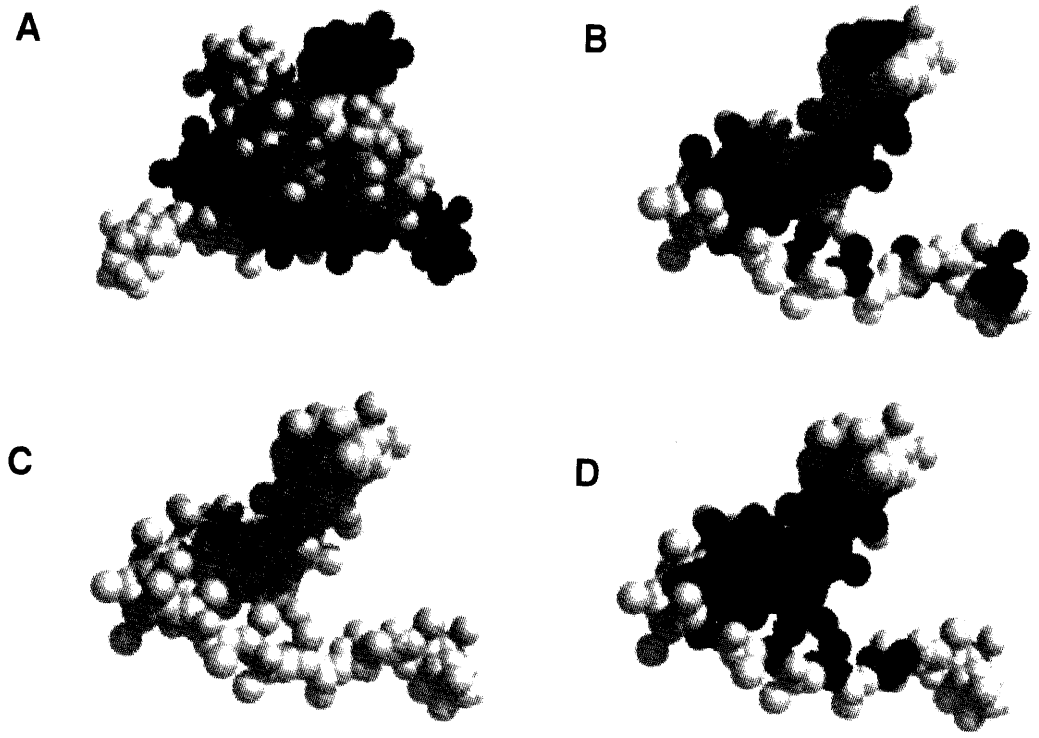

Fig. 3. Designing dimers. (**A**) Illustration of the dimer interface of the ARC repressor as a dimer, with one monomer in black and the other in white. (**B**) Color-coded hydrophobic and polar illustration (black, hydrophobic; white, polar) of the native sequence and (**C**) the GCSE-designed lowest energy sequence when the ARC monomer was used as the target. (**D**) The GCSE-designed sequence when the ARC dimer was used as the target. In (C), all residues appeared to be exposed to the algorithm and were assigned as polar, whereas with the full dimer as the target in (D), the residues now buried are assigned to be hydrophobic.

Table III. Number of encodable $n = 16$ conformations on square lattices by exact enumeration

Topological contacts t	Unique in sHP[a]	Unique in HP[b]	Unique in sHP and HP[c]	Lowest energy sHP[d]
6	1	0	–	–
7	337	163	163	159 (97.5%)
8	884	273	271	261 (95.6%)
9	53	20	20	20 (100.0%)
Total	1275	456	454	440 (96.5%)

[a]Number of conformations with t intrachain contacts that are unique native structures for the sHP potential.
[b]Number of conformations with t intrachain contacts that are unique native structures for the HP potential.
[c]Number of conformations that are encodable by at least one identical sequence for both potentials.
[d]Number of conformations among those in [c] with at least one sequence that encodes the given conformation uniquely for both potentials and has the lowest native energy among all converging sHP sequences.

Table IV. Numbers of $n = 16$ sHP sequences that fold uniquely to HP native structures on square lattices

Topological contacts t	sHP[a]	Lowest energy sHP[b]	Lowest energy sHP and HP[c]
7	887	223	167 (74.9%)
8	3265	367	268 (73.0%)
9	1200	34	26 (76.5%)
Total	5352	624	461 (73.9%)

The number of intrachain contacts is t.
[a]Total number of sHP sequences that encode HP native structures.
[b]Number of sHP sequences that have the lowest energy among all converging sHP sequences that fold to the same HP native structure.
[c]Number of lowest sHP energy sequences among those in [b] that would also fold uniquely to the same native structure for the HP potential.

The method has a few other notable failures. It assigned all P residues to bovine pancreatic trypsin inhibitor (BPTI; PDB identifier 1bba), finding no position buried enough to be assigned as an H. Functionally, inhibitors such as BPTI need an exposed hydrophobic surface area so that they can bind the active sites of their respective enzymes. The GCSE algorithm does not account for such functional constraints. For the hybrid protein constructed by Osmark *et al.* (1993), with secondary structural elements from chymotrypsin inhibitor-2 and subtilisin (1cis), the algorithm succeeded in only 63% of the monomer positions (Figure 2I and J and Table I). Here, buried polar residues like proline 82 and glycine 85 were misassigned. Such residues play a role in structure as well as stability. In addition, as with BPTI, the native molecule has exposed hydrophobic surface in an inhibiting loop for functional reasons, which GCSE designs to be polar. As expected, GCSE does a relatively poor job (62%, Table I) on calmodulin (3cln) because of its unusual shape. Hence, the GCSE algorithm performs best on the most globular proteins.

Dimeric proteins. We also designed three dimers: the ARC repressor (1arr), the MET repressor (1cmb) and κ-bungarotoxin (1kba). Comparisons in Figure 3 and Table I show that when the full dimer structure of 1arr was the target, sequence designs were much closer to the native than when only the monomer was the target (73 versus 62% of residues correct). The strips of hydrophobic residues down the center of the horseshoe-shaped monomer contact each another in the dimer, but in the monomer they appear exposed to solvent. Hence, the program judged these positions to be suitable for P residues in the

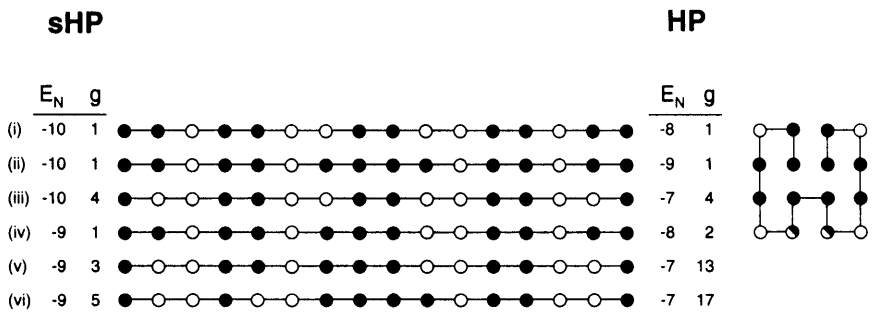

Fig. 4. Comparing sHP and HP potentials. Six sequences are generated by the inverse folding genetic algorithm described in the text to fold to the 2-D target structure on the right with the sHP potential (H, black bead; P, white bead). The native energy E_N and degeneracy g of the sequences are determined for both potentials by exact enumeration. The target structure is either the unique native structure ($g = 1$) or one of many lowest energy conformations ($g > 1$) of all six sequences for both the sHP and HP potentials. Two of the sequences with the lowest sHP energy of −10 fold uniquely to the target structure with both potentials (i and ii). The target structure is shown in these two sequences, and the half-filled beads are Ps for sequence (i) and Hs for sequence (ii). However, sequence (iii) with the lowest sHP energy does not fold uniquely, while sequence (iv) with higher sHP energy folds uniquely with the sHP potential. Sequences with lowest sHP energies tend to fold uniquely to the target structure, while those with higher sHP energies tend not to fold uniquely with the HP potential.

monomer but assigned them to H when given the dimer coordinates. Results with 1cmb were similar.

Sequences designed for the 1kba monomer were almost identical to those designed when the full dimer was the target; both designs were ~72% successful. The explanation is simple: the dimer interface is composed of backbone atoms; side chains are mainly tucked into monomer interiors. Hence, to the inverse folding algorithm, these residues look buried in each monomer, so dimerization has little effect on HP assignment in such a case.

While the GCSE method involves considerably more computer time than the Burial algorithm, and only slightly higher success rates, it offers one significant advantage. Because GCSE is based on an energy function rather than a structural property like burial, it can be readily generalized and used to explore and compare other energy functions. This may offer an opportunity to learn about the relative importance of other types of energy beyond simple hydrophobic and polar energy functions.

A lattice test of the GCSE method

Will the designed sequences fold to unique native conformations? At present, the only computational way to determine the uniqueness of folding rigorously is by using lattice models for which the full conformational space can be explored. Accordingly, we used GCSE to design lattice model HP sequences (Lau and Dill, 1989; Chan and Dill, 1991), and then studied whether those sequences folded uniquely. In the lattice model, a protein is represented by a linear self-avoiding path of a chain of 'beads' on a 2-D square lattice or a 3-D simple cubic lattice. Side chains are not considered in this model, and H or P monomers along the chain are centered at their sites.

Before describing the design results in the lattice system, we first address a technical issue involving the potential function in Equation 1, which we call the sHP potential because of its additional solvent term relative to the well-studied HP model (Lau and Dill, 1989; Chan and Dill, 1991, 1994; Yue and Dill, 1992; Dill *et al.*, 1995):

$$E = \varepsilon \sum_{i<j}^{N} h_{ij}. \qquad (3)$$

The surface term σ in Equation 1 is empirical and serves to prevent evolution towards the all-H sequence. Does the introduction of the extra solvent term change the model? We show in the section below that while the extra solvent term in the sHP potential imparts the ability to perform grand canonical design, where we do not have to know the HP composition in advance, it nevertheless does not seriously alter the HP model.

Lattice tests of the sHP potential. We performed exhaustive folding simulations on the 2-D lattice for the 2^{16} different sequences of chain length $n = 16$, using either the HP or the sHP potential (Table III). We used folding enumerations to address questions about the two energy functions described above. Does a given HP sequence fold to a different native structure, or with a different degeneracy, when the potential is sHP rather than HP? Our exact enumeration results showed that 98.8% (1520 out of 1539) of those sequences that fold uniquely under HP also fold uniquely under sHP. However, a randomly selected sequence is more likely to fold uniquely under the sHP potential than under HP: 13.9% (9107 sequences) of sequences fold uniquely under sHP, while 2.3% (1539 sequences) fold uniquely under HP (Table IV). In the sHP potential, those structures with the lowest energy for a given sequence must not only maximize H–H contacts but also minimize H–solvent contacts. As fewer configurations are likely to meet both criteria, the lowest energy state is more likely to fold uniquely under the sHP potential than under HP. Suppose we are given a particular lattice native structure and a set of sequences that fold uniquely to it (i.e. its convergence set; Chan and Dill, 1991). If a sequence is the lowest energy sequence for that structure when evaluated with the sHP potential, we call it a lowest sHP energy sequence. As demonstrated in Table IV, the majority (73.9%) of these lowest sHP energy sequences uniquely encode their target structures when folded not only under the sHP potential but also under HP. Thus, minimizing sHP energy in sequence design very likely leads to a unique folder in these lattice model tests.

Uniqueness. Enumeration provided us with the lowest energy sequence for every structure possible with 16mers. When we performed GCSE design on the structures, the algorithm nearly always produced these lowest energy sHP sequences. Hence, GCSE converges on the correct answer in these lattice model tests. Because these lowest sHP energy sequences also usually fold uniquely (see above), GCSE-designed sequences are often, but not always, unique folders. Figure 4 shows that GCSE is successful for the given target structure. Two of the three