Protein Design by Optimization of a Sequence-Structure Quality Function

Steven E. Brenner
S.E.Brenner@bioc.cam.ac.uk

MRC Laboratory of Molecular Biology
Hills Road
Cambridge CB2 2QH, UK

Alan Berry
ab117@mole.bio.cam.ac.uk

Department of Biochemistry and Molecular Biology
University of Leeds
Leeds LS2 9JT, UK

Department of Biochemistry
University of Cambridge
Tennis Court Road
Cambridge CB2 1QW, UK

Abstract

An automated procedure for protein design by optimization of a sequence-structure quality has been developed. The method selects a statistically optimal sequence for a particular structure, on the assumption that such a protein will adopt the desired structure. We present two optimization algorithms: one provides an exact optimization while the other uses a combinatorial technique for comparatively rapid results. Both are suitable for massively parallel computers. A prototype system was used to design sequences which should adopt the four-helix bundle conformation of myohemerythrin. These appear satisfactory to secondary structure and profile analysis. Detailed inspection reveals that the sequences are generally plausible but, as expected, lack some specific structural features. The design parameters provide some insight into the general determinants of protein structure.

Introduction

Despite decades of research, our present understanding of protein structure is still incomplete. While structural analysis has yielded many of the fundamental rules governing the general architecture of protein folds (Chothia & Finkelstein 1990), it has been limited by the relatively few known protein structures and the fact that they have been limited by evolution (Pastore & Lesk 1991). Therefore, mutation analysis has become one of the most powerful tools for studying the structure of proteins and has provided a wealth of information about particular folds. However, this data has left unanswered many general questions about the large-scale structural features of proteins. The most clear evidence of our imperfect knowledge of protein structure is that the protein folding problem remains largely unsolved: we can not determine the conformation a protein will adopt from the knowledge of its sequence alone.

Protein design has emerged as a promising technique for understanding protein structure because finding a sequence which will fold into a desired structure may be considerably simpler than computationally folding a sequence into a structure (Yue & Dill 1992). In addition to providing new scientific knowledge, new proteins have potential uses in biotechnology and medicine.

Many groups have taken up the challenge of building new proteins using "physical, statistical, and intuitive criteria" (Sander 1991) in a procedure called *design by modeling* (Brenner 1994). The four-helix bundle is one of the most popular structures being designed (Regan & DeGrado 1988; Hecht et al. 1990; Hill et al. 1990; Sander et al. 1992a; Schafmeister et al. 1993), and such one structure has been proven correct by crystallography. A related designed protein, consisting of two antiparallel alpha-helices, has had its structure solved by NMR (Kuroda et al. 1994). Attempts have also been made to build other structures (Sander et al. 1992b) including alpha-beta barrels (Goraj et al. 1990; Tanaka et al. 1994), a beta bell (Richardson & Richardson 1987), and a crystallin (Hubbard & Blundell 1989), as well as some folds not occurring in nature such as a miniature antibody (Pessi et al. 1993) and an alpha-beta "open sandwich" (Fedorov et al. 1992).

The success of several of these experiments has been encouraging and the partial failures have frequently been instructive. However, the manual procedure used to design the proteins means that the procedure is fundamentally irreproducible: the knowledge used to build one protein cannot be directly applied to the construction of another.

Some experimentalists have gone to the opposite extreme and made random sequences of amino acids either with hydrophobic/hydrophilic patterns (Kamtekar et al. 1993) or a limited set of residues (Davidson & Sauer 1994). While intriguing, these "design" experiments cannot be directly used to select single proteins which should adopt a given fold. However, some systems have been developed to find proteins sequences which meet particular criteria. Ponder and Richards designed sequences whose side chains could adopt acceptable rotamer positions and pack densely (Ponder & Richards 1987), while Yue and Dill found binary sequences of polar and non-polar residues which would form more hydrophobic contacts in the desired structure than in any alternative structure (Yue & Dill 1992).

A Quantitative Methodology

We have created a quantitative methodology which attempts to combine the comprehensibility and success of the design by modeling experiments with the reproducibility of the more general methods. It relies principally on the statistical features of all known protein structures to form a function which measures the compatibility of given sequence and structure. This function is then optimized over a selected structure to produce an "ideal" sequence for that fold. Thus, the procedure can be easily repeated any number of times on any structure.

The parameters entering into sequence-structure quality function, detailed elsewhere (Brenner & Berry 1994), can be divided into four main categories: position, *neighbor*, *uniqueness*, and *hints*. The first two are the main contributors to the design procedure and are statistically based. Position preferences describe the preference of a given amino acid for particular position within a protein (e.g., lysine is well suited for a solvent exposed helical position) The statistical tendency for two or more amino acids, such as leucine and valine, to be near each other falls within the neighbor preferences category. Uniqueness is a parameter which attempts to ensure that the designed sequence will be comparatively unstable in folds other than the desired one (Yue & Dill 1992), and hints incorporates a wide variety of details which are specific to the protein being designed, such as large motifs and functionality.

To form the evaluation function, each parameter is weighted according to its relative importance and summed with the others. This means that an arbitrary number of detailed criteria can be incorporated in the design system and that the method can be extended and modified as both experimental results and theoretical research provide additional information.

Figure 1. General, automated protein design. The quantitative protein design methodology accepts any plausible protein structure and selects a sequence which should be stable in that configuration. The figure is by M. Elms.

`g(x, u)` returns *position preference* of residue u at position x
`h(u, v)` returns *neighbor preference* of residue u adjacent to v

```
function tdso (l, r, s, t)  returns (score, seq)

  base case: no middle
  if (l + 1 == r)
    score = h(s, t)
    seq = <nothing>

  inductive case: ternary sum
  else
    score = 0    initialize to be lower than any actual score
    m = ceil((l + r) / 2)   middle position
    foreach a in amino acids
      retl = tdso (l, m, s, a)
      retr = tdso (m, r, a, t)
      cur = retl.score + retr.score + g(m, a)
      if (cur > score)
        score = cur
          concatenate left, middle, and right sequences
        seq = retl.seq . a . retr.seq
      endif
    endfor
  endif

  return (score, seq)
endfun
```

Figure 2. Pseudocode for the 1-dimensional TDSO algorithm. To optimize a function of n residues 1...n, with position and neighbor preferences specified by the functions `g` and `h`, respectively, run `tdso(0, n+1, L, R)`, where `L` and `R` are the left and right endcaps. This will return the optimal sequence and its score.

Optimization Methods

Ternary Division Sum Optimization

One Dimension. The optimization of a sequence-structure quality function containing position preferences is trivial: at each position select the best residue (e.g., by table lookup). So long as they are not overly complex, the uniqueness and hints criteria serve only as masks limiting the residues at particular positions. Thus, their inclusion does not increase the computational complexity of the design task.

Unfortunately, the neighbor preferences complicate sequence selection considerably. If the residue at each position interacts with the residues at every other position, then the optimization procedure requires the evaluation of every potential sequence—an exponential time procedure. Indeed, this exponential complexity is one of the major

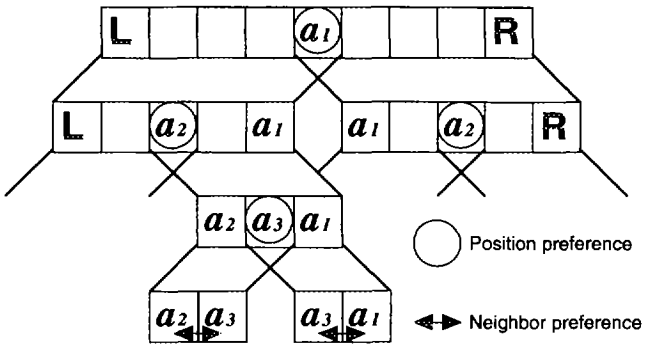


Figure 3. Schematic of TDSO Operation. L and R are end caps. See text for description

barriers to computational protein folding. However, one of the principal reasons that protein design may be a computationally easier task than protein folding is that the positions of all the residues are known in advance. Thus, interactions between residues which are distant in sequence need only be considered if the residues are close in space. In one dimension, the evaluation function F (with an implicit structure) over a sequence S has the form:

$$F(S) = \sum_{i=1}^{n} g(i, s_i) + \frac{1}{2}\left(h(s_{i-1}, s_i) + h(s_i, s_{i+1})\right)$$

where s_i is the residue at position i, $g(x, u)$ is the position preference of residue u at position x, and $h(u, v)$ is the neighbor preference of residues u and v.

A procedure for optimizing position and neighbor preferences in one dimension was developed by Steven P. Ketchpel (personal communication). A considerably more efficient version of this algorithm, ternary division sum optimization (TDSO), has been developed and its complexity analyzed. The principle underlying TDSO's operation is that the optimal sequence is that which has the best beginning, middle, and end. Pseudocode for the algorithm is provided in Figure 2 and its operation is diagrammatically portrayed in Figure 3.

TDSO begins with a sequence that has left and right end-caps. Then it computes the position preference for each of the 20 types of residue at the central location. To each of these it adds the recursively computed qualities of the left and right sides using the current central residue as the right and left end-cap, respectively, for these sides. Recursion is ceased when no central position can be selected, and the remaining neighbor preference is computed. The algorithm is $O(nk^{\log n})$ where k is the number of amino acids, 20, to be considered. Thus, the single-dimension procedure is polynomial and, although it is high order, is generally tractable for relatively small n.

Many Dimension. When TDSO is extended beyond one dimension to an arbitrary graph, it is impossible to pick a center pivot position which will divide the graph into two disjoint ones. Instead, it is necessary to find a set of nodes which together partition the graph into two subgraphs, as shown in Figure 4. However, instead of needing to test k residues at the center position at each recursive level, it is necessary to test all combinations of residues at these positions. If there are m border nodes, then k^m different partitions must be tested and summed with the best "left" and "right" subgraphs. As expected, the complexity of the algorithm increases as the connectivity increases. The graph version of the algorithm is $O(nk^{c\log n})$ where c is a measure of the connectivity of the graph and typically is a function of n. For example, c is $n^{(d-1)/d}$ for a regular grid of dimension d. Therefore, the complexity of TDSO grows exponentially as a root of n making this procedure unfeasible except for trivial cases.

However, though the running time of this algorithm grows dramatically with sequence length, it may still be tractable if various approximations are applied, including alpha/beta cutoffs (Horowitz & Sahni 1978; Aho et al. 1983) and heuristics that reduce k, the number of choices at each position, from 20 residues to just a few classes (e.g., hydrophobic, polar, and neutral). Moreover, the main foreach loop in TDSO is straightforward to parallelize, and with the exception of the selection of the best middle residue, no communication is necessary. Consequently, near-ideal speedup on massively parallel computers should be easy to achieve.

Simulated Annealing

While the exact optimization provided by TDSO is essential for verification of results, it is too time consuming when the design procedure is being developed and extended. For this reason, combinatorial optimization methods have been used to design sequences with a prototype sequence-structure quality function. In particular a straightforward simulated annealing protocol (Metropolis et al. 1953; Press et al. 1988) and several variants have been applied.

Serial. On a single processor, a sequence can be optimized for a particular structure by iterating over the sequence and randomly mutating residues at each position to any other residue. After each change, the modified sequence is

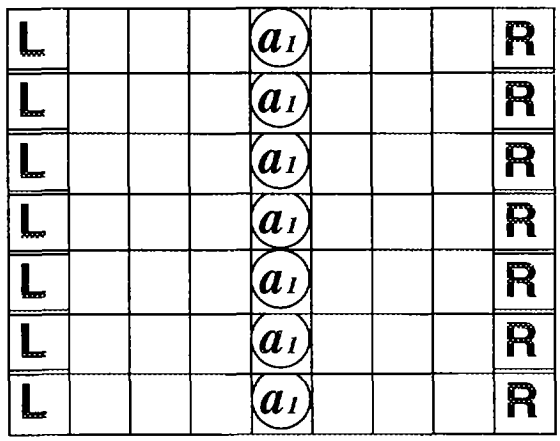


Figure 4. First level of multi-dimensional TDSO.

analyzed by the sequence-structure evaluation function and the mutation is accepted or rejected according to the variation in quality and the current temperature.

The parameters selected for the model system (below) caused the system to be cooled too quickly to converge to a single designed sequence. However, the different sequences generated from each set of parameters showed no significant difference in quality. Moreover, while (nearly) every natural protein adopts only a single structure, any given natural structure can be formed by several different sequences. Thus it is unnecessary to select the best possible sequence as stipulated by the objective function; near-optimal sequences will provide satisfactory results. Indeed, since the margins of error in the various rules which contribute to the overall methodology are considerable, the best sequence computed by the statistics is unlikely to optimize the underlying parameters that the statistics attempt to describe. In addition, the variable residues in the sequences designed with similar criteria help provide some insight into which positions are most constrained by the quality function.

Parallel. While the serial Metropolis method can produce results very quickly for small proteins with easily computable quality measures, it becomes inconvenient when working with large proteins and complex objective functions. Even in these complex instances, the computational design procedure is much faster than actually synthesizing the designed protein. However, to develop, test, and improve the design methodology, it is necessary to generate large numbers of sample sequences. Therefore, methods to increase the speed of designing new sequences are of considerable practical utility.

Because the quality function F is the sum of the qualities of each of the individual positions (although each position's quality depends on its neighbors), it is not difficult to parallelize the simulated annealing algorithm. One method of implementing this is to block partition the sequence

```
left, right
  delimit region of sequence to be optimized for this node
sequence
  the whole sequence, a portion of which will be optimized
procedure dupsa()
  foreach temperature
    gsync   synchronize all nodes
    while not ready to drop temperature
      foreach position between left and right
        foreach node with a neighbor residue
          if message received from this node
            set asynchronous receive for a
             new message from this node
          endif
        endfor
        choose a random new residue for current position, and
        accept or reject it according to normal annealing criteria
        if residue changed
          foreach node with neighbors to residue
            asynchronous send node the change
          endfor
        endif
      endfor
    endwhile
  endfor
endproc
```

Figure 6. Pseudocode for the DUPSA algorithm. Prior to running dupsa, each node has the global variables left and right set to delimit the regions of sequence to be optimized. Thus, on each node, positions left...right of the array sequence are selected by the optimization procedure. Positions of sequence outside this region are filled in by the asynchronous receive messages from neighbor nodes, and those positions which do not neighbor the region being optimized have random values. Following dupsa, each node must send the region of sequence it optimized to a leader node which collates the information. A considerable amount management detail has been omitted for clarity. Full source is available from SEB.

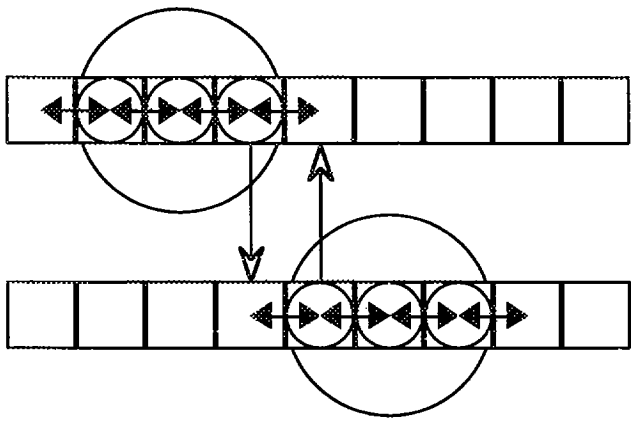


Figure 5. Schematic of DUPSA data distribution. Boxes are the sequence, and the large circles show segments of sequence to be optimized by a single node. Small circles show position preferences and doubleheaded arrows, neighbor preferences. Open-headed arrows show where communication is necessary.

evenly across the compute nodes of the parallel computer. Because of the neighbor preferences, every node needs to keep a record of the current neighbors of all of the positions which it is trying to optimize. Therefore, as shown in Figure 5, each node has an array for storing the whole sequence and "imagines" that it knows the entire sequence at the current level of annealing. However, each optimizes only the small subset of the sequence assigned. To be isomorphic with the serial optimization algorithm, synchronous communication must be used. That is, each node must wait until it has received messages containing information about the current neighbors before proceeding with the annealing. Coded this way, the parallel algorithm does provide significant speed enhancement over the serial one in many cases. However, as the number of neighbors increases, the communication overhead grows dramatically.

DUPSA. For this reason, an asynchronous version of the optimization procedure has also been developed. In this

algorithm (Figure 6), called delayed update parallel simulated annealing (DUPSA), messages about residue positions are sent to nodes containing their neighbors only if there has been a change. More importantly, each compute node optimizes its sequence with the most recently received information rather than waiting for messages with the most current neighbors to be received. This means that computation continues during the communication delay between when a neighbor residue changes and when this change is registered. As shown in Figure 7, even for with the simple prototype quality function (see below), there was a significant speedup over the serial algorithm when using this procedure, although for very small proteins the communication overhead still becomes predominant when more than two processors are used. However, when a (hypothetical) more complex is employed and multiple annealings are run in parallel, up to 100 compute nodes can be profitably used for even very small proteins.

There was some concern that the time period between residue change and update at neighbor nodes could potentially be problematic. However, for the design systems tested, this gap where out-of-date neighbor information is used had no detrimental effect on the final sequences. Thus, the DUPSA selects sequences as well as the serial simulated annealing protocol in a fraction of the time. Moreover, DUPSA is general enough to be applied to any similar optimization where local interactions are present.

A Model System

A model system containing only a few of the parameters in the full methodology has been used to evaluate the potential of the quantitative methodology (Brenner 1993; Brenner & Berry 1994). By formulating an evaluation function which incorporates only naive secondary structure, solvent accessibility, and primary structure neighbor preferences, we did not expect to produce a system which would produce reasonable sequences. However, the model system which operates on those parameters, as well as a hint (called diversity) to force the composition of the designed sequences to reflect that of the SwissProt database (Bairoch & Boeckmann 1991), has done surprisingly well. The system has been used to design sequences for phage 434 Cro (Brenner 1993; Brenner & Berry 1994), hemoglobin, two SH3 domains (Brenner 1994), lactate dehydrogenase, and ubiquitin. Here we present new sequences to adopt the fold of *Themiste zostericola* myohemerythrin (Sheriff et al. 1987), as represented by the coordinates in PDB entry 2mhr.

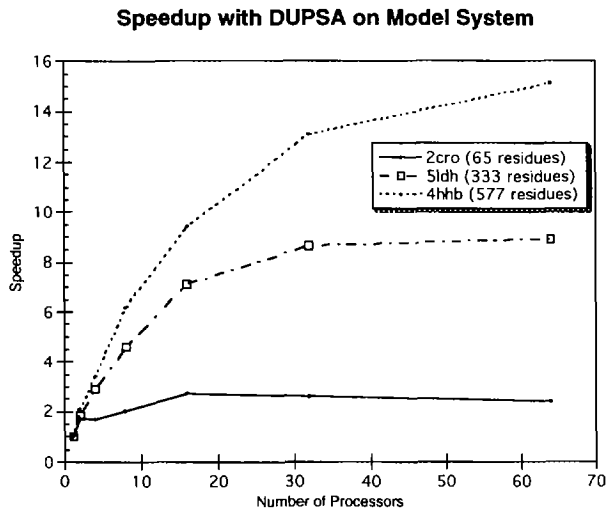


Figure 7. Speedup when the model evaluation function without the diversity hint was optimized for several structures using DUPSA.

Myohemerythrin is perhaps the canonical four-helix bundle, with four long, well formed helices. The natural protein binds oxygen through two irons which are coordinated by several histidines. However, no rules for ligand binding or other functionality were incorporated into the model design system used to select these sequences.

Designed Sequences

To explore the range of the design system and to gain a better understanding of the relative importance of its constituent parameters, a wide variety of different parameter weightings were used in designing the sequences shown in the Sequence Table. Thus some sequences were built using secondary structure preference almost exclusively (e.g., 44-46), while others were mainly selected on the basis of solvent accessibility (as in 47 and 48). However, most sequences were designed using a mix of all four parameters.

When inspected, most sequences appeared normal, aside from a possible excess of certain residues such as lysine and alanine. However, few sequences, particularly those selected with a strong bias towards secondary structure, look positively bizarre because of their extremely biased residue composition. Intriguingly, none of the designed proteins showed significant homology ($p < 0.001$ for blastp (Altschul et al. 1990) with the seg filter and match matrix on the May 1994 nr database) to any natural proteins. As a test of the sequences' propensity to adopt the correct local structure, they were subjected to secondary structure prediction (Rost & Sander 1993; Rost et al. 1994) and scored on how well the predicted structure matched the desired structure. Perhaps surprisingly, nearly all produced very good results; indeed, the designed sequences generally scored higher than the native.

Structural Models

In addition, structural models were made threading the designed sequence onto the structure of the natural myohemerythrin structure, using the method in (Brenner & Berry 1994). A schematic view of the natural and a designed protein (Figure 8), shows that the selection of residues looked reasonable, although, the packing between the helices was much looser in the designed sequence than in the natural. In addition, none of the many hydrogen-bonded turns in the natural sequence appeared in the designed sequences. However, since that no information about either

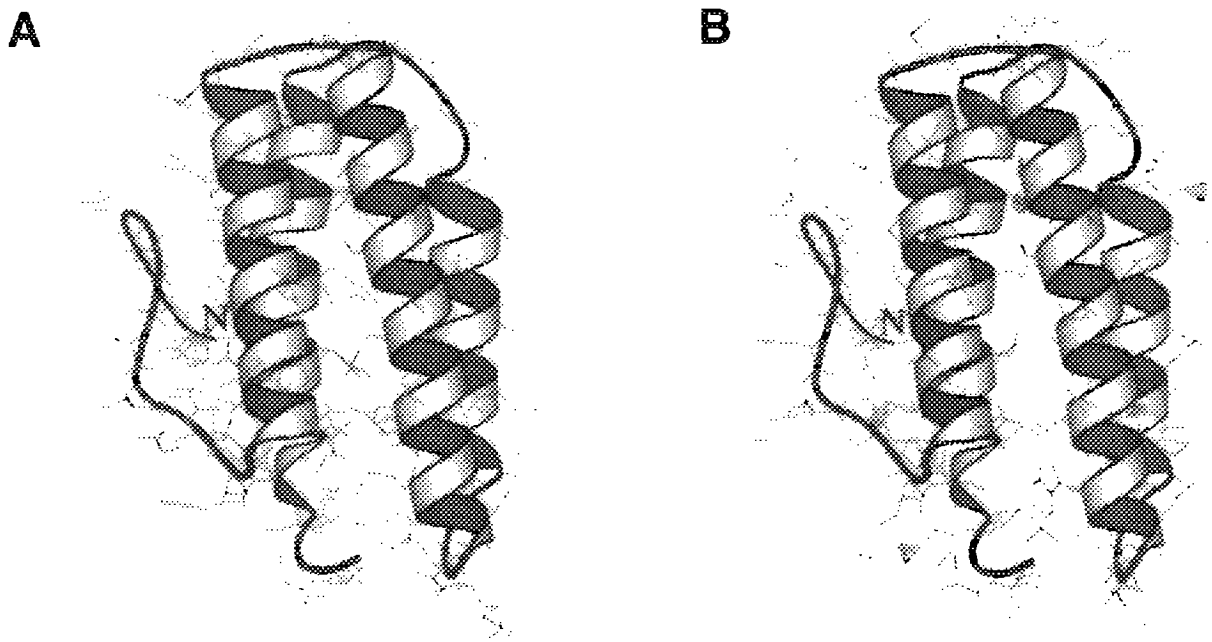


Figure 8. *Themiste zostericola* myohemerythrin structure (coordinates from PDB entry 2mhr) threaded with natural and designed sequences. The natural protein is shown in *A* , while a model has been constructed by threading a designed sequence (Sequence Table, line 1) onto the 2mhr backbone in B. The force field used for generating the structural model from the designed sequence contained an error which results in misshapen aromatic residues.

of these features was included in the optimization function, these results were expected.

What is more surprising, however, is how well the designed sequences fared when the structural models were analyzed with environmental profiles (Lüthy et al. 1992). Most scored about as well as the native structure and none of the sequences failed this verification assay.

In general, these uniformly positive evaluations of the designed sequence over a extraordinary range of different weightings agree with the results found for the all-helix Cro protein. It may be that this indicates that al weightings are acceptable. However, it seems more likely that the evaluation techniques used simply fail to differentiate between some good and poor sequences. If this is the case, then selecting appropriate weightings will be one of the most challenging tasks in constructing fully successful quantitative design systems. Nonetheless, some quality variation is visible, and can be exploited to ensure that experimentally constructed proteins have the optimal set of weighted parameters. Moreover, data culled from beta sheet proteins, which show more variation in quality (Brenner 1994), can also be used to tune the design system.

Conclusion

Protein design is a challenging task which provides a new way of understanding protein structure and has practical utility. However, most work in this field has been irreproducible and unverifiable because of the qualitative and intuitive nature of the design process. We have presented a quantitative methodology which automatically designs proteins by optimizing a sequence-structure quality function over a particular structure to yield an optimal sequence. With the parallelizable supporting algorithms, the task is relatively quick and amenable to exploration. Indeed, a model system containing only a few criteria has already designed sequences which score well according to secondary structure and tertiary structure evaluation.

We are initiating the essential experimental work to synthesize and characterize proteins designed by the quantitative system. Concomitantly, more sophisticated versions of the protein design system are being constructed. If successful, the method will be able to generate new proteins with pre-determined structure and confirm the relationships between statistically observed features of proteins and their relative structural importance.

Acknowledgments

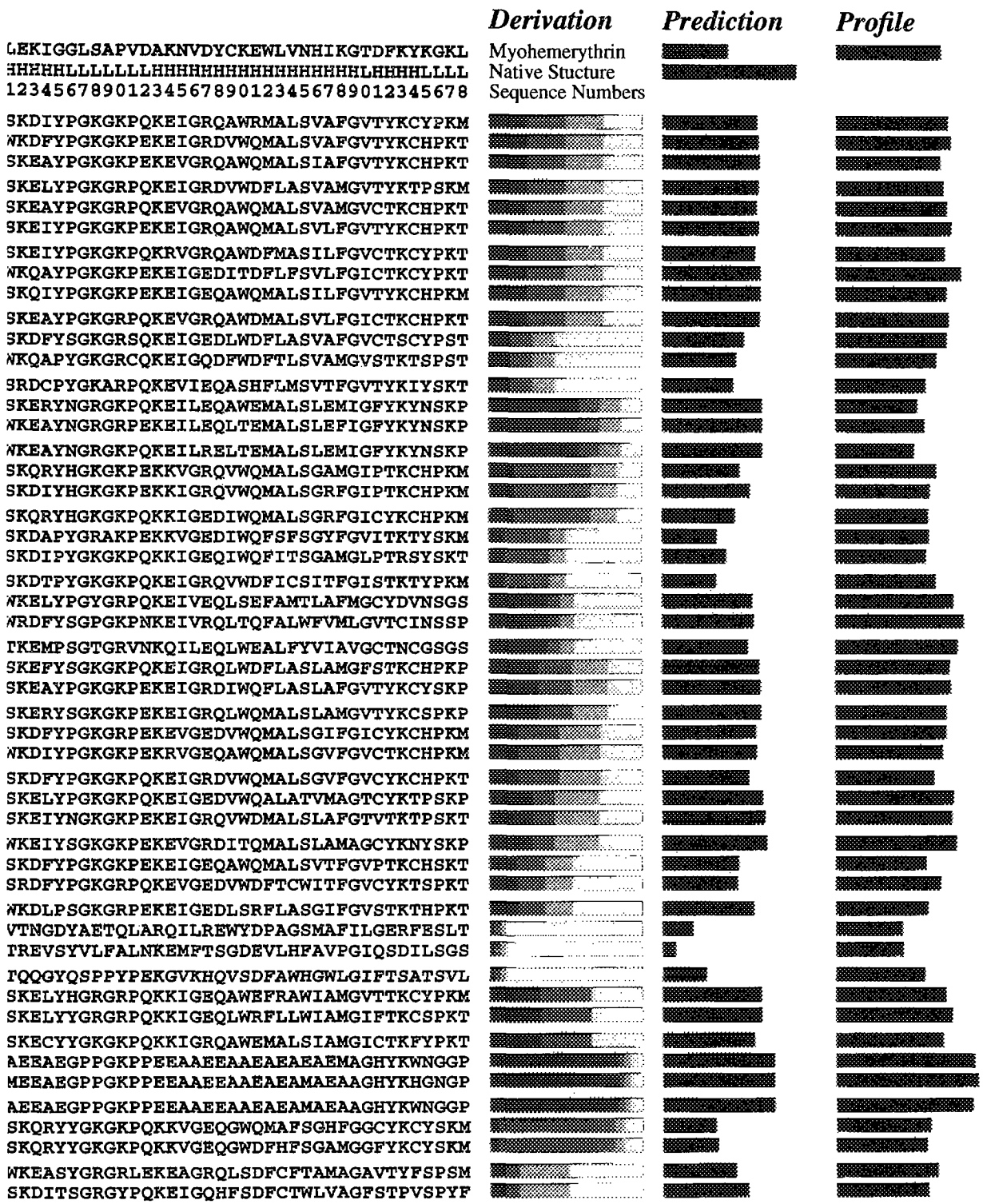
S.E.B. was supported by a Herchel Smith Harvard Scholarship and his research was sponsored by the NSF and the SDSC. A.B. was a Royal Society 1983 University Research Fellow and was supported by the SERC. Invaluable computing resources were provided by the members of the Cambridge Centre for Molecular Recognition and the University of Cambridge School of Biological Sciences. The NCBI network BLAST service was used for homology searches. Steven P. Ketchpel originally developed the algorithm from which TDSO is derived and helped evaluate the version presented here. Drs. Cyrus H. Chothia and Andrew D. McLachlan provided stimulating discussion and Anne M. Joseph critically read portions of the manuscript.

Designed Sequence

```
a  GWEIPEPYVWDESFRVFYEQLDEEHKKIFKGIFDCIRDNSAPNLATLVKVTTNHFTHEEAMMDAAKYSEVVPHKKMHKDF
b  LLLLLLLLLLLLHHHLLLLHHHHHHHHHHHHHHHHHHHLLLHHHHHHHHHHHHHHHHHHHHHHHLLLLLHHHHHHHHHH
c  12345678901234567890123456789012345678901234567890123456789012345678901234567890

1  MKRNGKRPQGHKDFSPKTKEALDIAEKALKAINILDKETVEKAGQHFYKVLNEALKNIFHWLNRGKGRPVKELQKDLKDL
2  MKRHGKRPQGYKDFSPKTRKMADILEKALKALNVLQKETVEKAVDNIYKGINEALKNIFNSLHRGKGRPVKEAQKDIKQA
3  MRKYGKRPQGNKDVSPKTKEALDIREKMAKILNIFDKNTVEKAVDNIYKGFDEALKNIFHWLHRGKGRPVKEAQKQLKDL
4  MRKYGKRPQGNKDVSPKTKEALDIAEKALKALNIFDKNTVEKAGQHFYKGINEALKNIFHWLHRGKGRPVKEAQKDIKQL
5  MKRYGKRPQGYKNVSPKTKEALDILEKILKALHIFDKDTVEKAVDNFYKGINEALKNIFNWIHRGKGRPVKEAQKDLKDF
6  MKRDGKRPQGYKDFSPKTKELADVAEKAMKALNIFDKNTVEKAGQNIYKGWHRALKNIFNSLHRGKGRPVKEADKDIKQL
7  MKRYGKQPQGNKNVPSKTKEALDILEKAEKALNILDKETVEKAVDHFYKGAHRLAKHLMNWINRGKGRPVKEAQKDIKDF
8  MKRYGKRPQGDKNVSPRTKEALDILEKALKALHIFDKNTVEKAVDNIYKGAHRALKHLMNWMNRGKGRPVKEAQKQLKQA
9  MRKHGKRPQGYKDFSPKTKEALDVAEKALKAINVLDKETVEKAVDNIYKGINRALKNAFNWLHRGKGRPVKRLQKDLKDF
10 MKRYGKRPQGYKDVPSKTKEALQVLEKALKAINIFDKNTVEKAVDNIYKGINEALKNIFHWMHRGKGRPVKEAQKDLKDF
11 MKTYGKRPTGNKNVPSRTRKVINILQRIMKIAHVADYPTAEKAVDHLPEIAQEALKHLMNWLNEGSGRPVKEAQKQLKDF
12 MKRHGRYPYGYKNCSPRTKEMADIAEKALKVLNVAQENTVEKAIDHISKVAHRALKNIFNSTPEGSGRPVKEIQKDLKDF
13 MEKTGKRPYGTKSFSPKTKEALHVADKALKLAHGIDKNTVQKAIDNIPEGCNRAVKNWFNSWQRGSGRPVKEADEDMKQI
14 MKRDGKRPNGDKDFSPRTEKEAEIAEKEAEALHLEDEDTPEKALQHFYELEQEAEKHLFNWLQEGNPRSIKEAEKDEKEL
15 MKRDGKRPDGDKDFSPRTKEAEDIAEKEAEALHLEQENTPEKALENIYEIEDEAEKHLMHWLQEGNPRSIKELQEQEKEA
16 MKRDGKRPDGDKDFSPRTEKEAEIAEKAEKELHLEQEDTPEKALQHFYEIENEALKNLFHWLQEGNPRSIKEAQEDEKEA
17 KRKYGKRPQGYKDVSPKTKKIFDVAEKEAKALNGFDKETAEKAGDNIYKGINEALKNIFNSWHRGKGRPVKEAQKDIKDF
18 KRKYGKRPQGYKDVSPKTKKALDVAEKEMKAINGFDKETAEKAGDNIYKGFNEALKNIFNSWHRGKGRPVKEAQKDLKQA
19 KRKYGKRPQGYKDVSPKTKKALDVAEKALKAINGFDKETAEKAGDNIPKGFNEAIKNWMNSLHRGKGRPVKEAQKDEKDF
20 KRKYGRQPEGNKHVSPRTKKTCQVAEKAMKTAHGIDKDTAEKAGQHVPKGINECVKNVFNSINRLKARPIKEAQKDWKDM
21 MRKYGRRPQGHENVSPKTKKAVDVAEKAVKITNVADKETVEKAGDNFYKGFNEAIKHWMNSCHRLKARPVKEADKQCKDF
22 RRRYGRQPEGYKNVSPKTKKAVDIAEKAIKMANGAQKNTAEKAGDHVPKGWNEAVKNCMHSFHRLKVESVKEAQKDLKDF
23 MTPTGSNPTGNKHFSPRTKEALDVLERLVEIFHLADESTPEKAVDYAWQILQRLIKNIANTAHRGSGRSIKECDEQAEDL
24 MTPTGTYPNGPKDVSCGTKELFDVIEKVLEMAHILDFSTVEKALDNLYRALYEALRQAAHSADRGSGNSIKEIHRQLEQA
25 MPSTGRYPTGSDSVPPSTKELADVLEKALRLFYVIQRDTVEKAIDHFYELFHRALKNAMHWLNEGNPDSIKEAQEDIRQA
26 MKRYGKRPQGYKDVPSKTKELADIAEKALKALNILQKNTVEKAIDNIYKGAHRALKNIFHWMNRGKGRPVKELQKDEKDL
27 MKRDGKRPQGDKNFSPKTKEALQVAEKAMKEINILQKNTVEKAIDNIYKGFHRALKNAMHWLHRGKGRPVKELDKQLKDL
28 MKRYGRQPQGNKDFSPKTKEALDIAEKALKALNIFDKNTVEKAVDHFYKILHEALKNIFNWLHRGKGRPVKEAQKDIKDL
29 MKRYGKRPQGNKDVSPKTKEAVDVARKALKALHGAQKETVEKAGQHFYKGINEALKNIFNWINRGKGRPVKEAQKDLKDI
30 MKRHGKRPQGYKDVSPKTKEALDVAEKALKAINIFDKETVEKAGQNIYKGFNRALKNIFNSLHRGKGRPVKEAQKDIKDY
31 MRKYGKRPEGHKDVSPRTKEAVDIAEKMAKALNIFDKETVEKAGQNIYKGINEALKNAFHWINRGKGRPVKEAQKDLKQL
32 MKRYGRQPTGNKDFSPRTKEALQILEKAMKALNILQKNTVEKAVDNFYKIAHRALKNIFHWLHRGKGRSIKELDEDLKDF
33 MKRYGRQPNGYKDFSPKTKEALDVAEKAMKILHLAQKNTPEKALQHFYKILDEALKNIFNWLHRGKGRPVKEAQRDIKDL
34 MKRTGRQPTGNKDFSPKTKEAVDILEKILKALNIADKDTPEKALQHFYRVLNEALKNIFHWLHRGKGRPVKELQKQAKDF
35 MKRHGKRPDGYKDVSPRTKRLAQIAEKVAKIMNIFDKYTVEKAGDNVYKITNEALKNAFHWINRGKPRSVKEAQKQCKDL
36 MKRHGKQPEGYKSFSPKTKEAVDVLEKALKALNIAQKNTVRKAIHNIYKIAQEAMKNAMNSIHRGPGRSVKEAQKDLKDL
37 MKRNGRQPQGYKDCSCKTKEALQVAEKMAKAVNIAQKYTVEKAIDHFYKIVNRAVKNMFNWTHEIPGRPVKEADKQLKDY
38 GETPASPGYGNNPMVGRLKRATNIFDALDKWIHVTMLKVVEDLGCSVHLVPKYKEKNFLISCSDVPSFHIQQAQKSLART
39 KGSQVLNQFADSQAVCGNKECINIPAEAATKTHLYEKDMAQTLLKRVWKGLERGFRWLLPMDINGPTRYITRYDPPVHSA
40 PAPPGREIVANELGSVKYKTLKTLFDQCKRDITVMEHNDADILMNSVTFGLLEIARNCARSFNRLYAMSGERVDLAIEKL
41 KKKYGKRPQGDKDCPPKTKKLMDIIEKALKFFHILQKNTVEKAGDNLYKGANEALKHLFNSLNRGKPRSIKEAQKDLKDA
42 KKKYGKQPQGHKDCPPKTKKRADIFRKAAKAINIADKNTVEKAGQNAYKGLNELMKHAMNSFHEGKPRSVKELDKDLKDE
43 KKRNGKRPQGDKDVPPKTKKLADIFEKAEKMLNIAQKNTVEKAGDNLYKGLHRLAKHLFHWLNRGKPRSIKEADKQEKDF
44 PPPPPKNPPPPKDHPPKPEEAAEAAEEAAEAAEAAEENPPEEAAEEAAEAAEEMAEEAAEMAEEGGPNGAEEAEEEAEEA
45 PPPPPKNPPPPKDWPPKPEEAAEAAEEAAEAAEAAEENPPEEAAEEAAEAAEEAAEEAMEAAEEGGPNGAEEAEEEAEEA
46 PPPPPKNPPPPKDHPPKPEEAAEAAEEAAEAAEAAEENPPEEAAEEAAEMAEEAAEEMAEAAEEGGPNGAEEAEEEAEEA
47 KKKHGKQPQGDKDVSPKTKKRFDVAEKREKFINGIDKETAEKAGDNGPKGINEAGKNIMNSINRGKGRPVKERQKDIKQA
48 KKKHGKQPQGHKDVSPKTKKRFDVAEKERKICNGIDKETAEKAGDNGPKGINEAGKNIMNSINRGKGRPVKERQKDIKQA
49 MTPTGRQVTGPESLPFSTKEILQVLQYCWKVINIVDLSTAEKALQHFYRGINEALKNIANSLNRGHPHPVKDIDEDLRDV
50 MRWNGSYCTGPETVPGRMKEAVDILEKLARLANVIDLNTAEKALQHFSTVAQRALKNLMHYINEIPGRSVKEADEDLRQL
```

Sequence Table. Sequences designed to adopt the fold of *Themiste zostericola* myohemerythrin fold (Sheriff et al, 1987), PDB entry 2mhr. Line *a* shows the natural protein sequence and *b* lists the crystallographically assigned secondary structure (H: helix; E: sheet; L: loop). Sequences designed by the prototype system to adopt this fold are given on line *1-50*. A variety of different weightings of the constituent parameters were used, as shown in the *Derivation* column; from darkest to lightest, the regions represent the relative weighting, on an arbitrary scale, of secondary structure preference, solvent accessibility preference, primary structure neighbor preference, and diversity. The *Prediction* column (lines *a, 1-50*) shows how well the sequences' secondary structure prediction agreed with the desired secondary structure



on line *b*. The score here was computed by adding two points for every position which was predicted to have the correct structure and deducting one for every position which was predicted to have an undesired structure. For scale, line *b* shows the score that would be received by a perfect prediction. The *Profile* column shows how well the structural models of the designed sequences rate in a profile analysis. The bars show by how much the assigned score exceeds the minimum quality cutoff of 24.0. The profile score of the natural myohemerythrin is 48.8.

References

Aho AV, Hopcroft JE, Ullman JD. 1983. Data Structures and Algorithms. Reading, MA: Addison-Wesley.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic Local Alignment Search Tool. Journal of Molecular Biology 215:403-410.

Bairoch A, Boeckmann B. 1991. The Swiss-Prot protein-sequence data-bank. Nucleic Acids Research 19 S:2247-2248.

Brenner SE. 1993. Development of a Quantitative Methodology for the de novo Design of Proteins. Thesis. University of Cambridge.

Brenner SE. 1994. Methods and Algorithms for the Design of Proteins. In: Schulze-Kremer S, eds. Advances in Molecular Bioinformatics. Amsterdam: IOS Press.

Brenner SE, Berry A. 1994. A quantitative methodology for the de novo design of proteins. *submitted*

Chothia C, Finkelstein AV. 1990. The Classification And Origins Of Protein Folding Patterns. Annual Review Of Biochemistry 59:1007-1039.

Davidson AR, Sauer RT. 1994. Folded proteins occur frequently in libraries of random amino-acid sequences. Proceedings of the National Academy of Sciences, USA 91:2146-2150.

Fedorov AN, Dolgikh DA, Chemeris VV, Chernov BK, Finkelstein AV, Schulga AA, Alakhov YB, Kirpichnikov MP, Ptitsyn OB. 1992. De novo design, synthesis and study of Albebetin, a polypeptide with a predetermined three-dimensional structure: Probing the structure at the nanogram level. Jour*nal of Molecular Biology* **225**:927-931.

Goraj K, Renard A, Martial JA. 1990. Synthesis, purification and initial structural characterization of Octarellin, a de novo polypeptide modeled on the alpha/beta-barrel proteins. *Protein Engineering* **3**:259-266.

Hecht MH, Richardson JS, Richardson DC, Ogden RC. 1990. De novo design, expression, and characterization of Felix: A four-helix bundle protein of native-like sequence. *Science* **249**:884-891.

Hill CP, Anderson DH, Wesson L, DeGrado WF, Eisenberg D. 1990. Crystal structure of Alpha-1: Implications for protein design. *Science* **249**:543-546.

Horowitz E, Sahni S. 1978. *Fundamentals of Computer Algorithms*. Rockville, MD: Computer Science Press.

Hubbard TJ, Blundell TL. 1989. The design of novel proteins using a knowledge-based approach to computer-aided modelling. In: van Gunsteren WF, Weiner PK, eds. *Computer Simulations of Biomolecular Systems: Theoretical and Experimental Applications*. Leiden, Holland: ESCOM. pp 168-82.

Kamtekar S, Schiffer JM, Xiong HY, Babik JM, Hecht MH. 1993. Protein design by binary patterning of polar and nonpolar amino-acids. *Science* **262**:1680-1685.

Kuroda Y, Nakai T, Ohkubo T. 1994. Solution structure of a de novo helical protein by 2D-NMR spectroscopy. *Journal of Molecular Biology* **236**:862-868.

Lüthy R, Bowie JU, Eisenberg D. 1992. Assessment of protein models with three-dimensional profiles. *Nature* **356**:83-85.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21**:1087-1092.

Pastore A, Lesk AM. 1991. Brave new proteins: What evolution reveals about protein structure. *Current Opinion in Biotechnology* **2**:592-598.

Pessi A, Bianchi E, Crameri A, Venturini S, Tramontano A, Sollazzo M. 1993. A designed metal-binding protein with a novel fold. *Nature* **362**:367-369.

Ponder JW, Richards FM. 1987. Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology* **193**:775-791.

Press WH, Flannery BP, Teukolsky SA, Vetterling WT. 1988. *Numerical Recipies in C: The Art of Scientific Computing*. Cambridge: University of Cambridge Press.

Regan L, DeGrado WF. 1988. Characterization of a helical protein designed from first principles. *Science* **241**:976-978.

Richardson JS, Richardson DC. 1987. Some design principles: Betabellin. In: Oxender DL, Fox CF, eds. *Protein Engineering*. New York: Alan R. Liss, Inc. pp 149-163.

Rost B, Sander C. 1993. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology* **232**:584-599.

Rost B, Sander C, Schneider R. 1994. PHD: An automatic mail server for protein secondary structure prediction. *Computer Applications in the Biosciences* **10**:53-60.

Sander C. 1991. De novo design of proteins. *Current Opinion in Structural Biology* **1**:630-637.

Sander C, Scharf M, Schneider R. 1992a. Design of protein structures. In: Rees AR, Sternberg MJE, Wetzel R, eds. *Protein Engineering: A Practical Approach*. Oxford: Oxford University Press. pp 89-115.

Sander C, Vriend G, Bazan F, Horovitz A, Nakamura H, Ribas L, Finkelstein AV, Lockhart A, Merkl R, Perry LJ, Emery SC, Gaboriaud C, Marks C, Moult J, Verlinde C, Eberhard M, Elofsson A, Hubbard TJP, Regan L, Banks J, Jappelli R, Lesk AM, Tramontano A. 1992b. Protein design on computers: Five new proteins: Shpilka, Grendel, Fingerclasp, Leather, and Aida. *Proteins: Structure, Function, and Genetics* **12**:105-110.

Schafmeister CE, Miercke LJW, Stroud RM. 1993. Structure at 2.5 angstrom of a designed peptide that maintains solubility of membrane-proteins. *Science* **262**:734-738.

Sheriff S, Hendrickson WA, Smith JL. 1987. Structure of Myohemerythrin in the Azidomet State at 1.7/1.3 Å Resolution. *Journal of Molecular Biology* **197**:273-296.

Tanaka T, Kimura H, Hayashi M, Fujiyoshi Y, Fukuhara K, Nakamura H. 1994. Characteristics of a de novo designed protein. *Protein Science* **3**:419-427.

Yue K, Dill KA. 1992. Inverse protein folding problem: Designing polymer sequences. *Proceedings of the National Academy of Sciences, USA* **89**:4163-4167.