ASSESSING THE PERFORMANCE OF FOLD RECOGNITION METHODS BY MEANS OF A COMPREHENSIVE BENCHMARK.

DANIEL FISCHER, ARNE ELOFSSON, DANNY RICE & DAVID EISENBERG[a].
UCLA-DOE Lab. of Structural Biology & Molecular Medicine
Molecular Biology Institute, UCLA
BOX 951570 Los Angeles, CA-90095-1570

Recently there has been an explosion of methods for fold recognition. These methods seek to align a protein sequence to a three-dimensional structure and measure the compatibility of the sequence to the structure. In this work, we present a benchmark to assess the performance of such methods. The benchmark consists of a set of protein sequences matched by superposition to known structures. This set covers a wide range of protein families, and includes matching proteins with insignificant sequence similarity. To demonstrate the usefulness of this benchmark, we apply it here to compare different fold-recognition methods developed through the years in our group as well as several sequence-sequence substitution matrices. The results show that "global-local" alignments are superior to either local or global alignments. The most effective sequence-sequence matching matrix is the Gonnet table. The best performance overall is obtained by a method which combines the 3D-1D profiles of Bowie et al. [1] with a substitution matrix and takes into account residue pairwise interactions.

1 Introduction

In the fold-recognition problem we ask: "Is the sequence of a protein of unknown structure 'compatible' to the fold of a known protein, and if so, to which one?" The practical goal of a fold-recognition method is to assign each new amino acid sequence to the known three-dimensional fold which it most closely resembles. The classical method of making this assignment has been to establish a similarity of the new sequence to some *sequence* of known structure. In 1991, Bowie et al.[1] developed an alternative method: to score the compatibility of the new sequence against a known *three-dimensional structure*. This method has been termed inverted protein folding or 3D profiles[1]. Since then, a variety of fold-recognition methods have been published [2,3,4,5,6,7]. The approaches used differ in one or more of the four essential components of fold recognition, namely, (i) the representation of the protein, (ii) the evaluation of the compatibility between the unknown sequence and a fold, (iii) the algorithm to search for the optimal alignment and (iv) the way the ranking is computed and the way significance is estimated. The representation of the protein structure can

[a]Corresponding author. Fax: 310-206-3914, Tel: 310-825-3754.

be an all atom structure, a backbone structure, a string of β-carbon atoms, a set of inter-residue distances or even, in the simplest case, a string of amino-acid names (that is, a sequence). The evaluation of compatibility can be a table of scores for matching residue to residue (such as Dayhoff's[8] or Gonnet's[9] substitution matrices), or residue to its environment (sometimes called 3D-1D scores[1]). The method used for aligning the sequence to the structure can be a dynamic programming algorithm[10,11], multi-level dynamic programming[3], matching of segments with a Monte Carlo[7] or a branch and bound algorithm[12]. The ranking can consider either the raw scores of the alignments or some normalized scores. Assessing significance can be achieved by considering some measure of statistical significance such as a z-score. Each of these steps involves representations and parameters. Selecting the best approximations and parameters is crucial to success, but is hindered by the complexity of the entire procedure. It is this problem that this paper addresses.

Our goal is to devise a benchmark that can aid in assessing the performance of a fold-recognition method in an objective, unbiased and thorough way. The benchmark is **independent** of the representation of the proteins, the compatibility definition, the search algorithm, and the ranking and significance estimation procedures used in the method being evaluated. Thus, it allows a systematic comparison of different methods. Benchmarks are routinely used to assess performance of sequence-sequence alignment (e.g.[13,14]) and secondary structure prediction methods (e.g.[15]). However, in fold recognition, no standard procedure to assess performance has been established. This benchmark is a first attempt to establish such a standard in the field of fold recognition. This benchmark may also aid in determining the strengths or weaknesses of different fold-recognition methods.

Performance assessment should address the balance between sensitivity - the ability to calculate high-ranking scores for the correct answer- and selectivity -the ability to calculate low-ranking scores for unrelated folds[14]. Another important aspect in assessing the performance of a method is the evaluation of the accuracy of the alignments obtained. The benchmark presented here quantifies both the sensitivity and selectivity. Alignment accuracy, however, is the subject of a different study.

This paper is organized as follows. In the Materials and Methods section we first present the benchmark and then describe the various fold-recognition methods evaluated using the benchmark. In the Results section we present the results of the performance assessment of some of these methods. In the last section we analyze the results of the evaluations and we discuss the merits and limitations of this benchmark.

2 Materials and Methods

2.1 The Benchmark

The benchmark consists of three components. The first is a set P of proteins of known structure obtained from a structurally non-redundant dataset of proteins. The second is a list S of test sequences. The third component is a set L of pairs of the form (s, p), where $s \in S$ and $p \in P$. L identifies for each test sequence, which fold in P is the most similar to it. For each s, the most compatible fold in P is objectively determined by structural comparison[16], as the structure of s is actually known. Each fold-recognition method being evaluated considers each s as a probe (obviously ignoring its structure), aligns it into each $p \in P$ and produces a ranked list of the compatibility of s with each p. The benchmark uses L to assess how well the method succeeded in each test sequence, i.e. the ranked list is searched to find at what position the expected p is. An ideal performance would be one that identifies the expected p at rank 1. To assess sensitivity, an overall score comprising the performance on all pairs in L is given for each method. To assess selectivity a reliability level is also computed (see below).

P is obtained from the representative dataset derived in[16]. This is a sequence-independent dataset, obtained by structural criteria only, using a 1994 release of the PDB (Protein Data Bank[17]). It covers all the different folds known at that time. It is non-redundant both in structure and sequence, i.e. no two chains in P are structurally nor sequentially similar up to given thresholds [b]. The size of P is 301 and is available listed from the authors. The sequences to be used as probes were selected by analyzing the pairwise comparisons carried out during the construction of the representative dataset. First, every chain s from the PDB which is represented by some $p \in P$ (i.e. is structurally similar to p), and which has less than 30% sequence identity with p is selected, and the pair (s, p) added to L. If two chains s_1 and s_2 are represented by the same p and share more than 30% sequence identity, then only one of them is selected. Second, the results of an all-against-all structural comparison of the representative chains (the P set) is analyzed and pairs of chains in P belonging to the same super-family or fold and which are just below the structural threshold used in the derivation of P, are also included in L, using either one of them as probe. By definition, these pairs also have below-thresholds sequence similarity.

[b] No two entries have a sequence identity percentage above 35% and when optimally superimposed, no more than half the residues of the larger structure are matched to residues of the other structure at a distance of at most 3Å; for more details on the derivation of the dataset and on the structural comparison algorithm used see[16]

TABLE I. THE SEQUENCE-STRUCTURE PAIRS*.

s	p	%	DIFF.	s	p	%	DIFF.
1mdc	1ifc	21	1.0	1mup	1rbp	14	4.4
1npx	3grs	20	1.0	1cpcl	1cola	17	4.6
1onc	7rsa	26	1.0	1ak3a	1gky	17	5.3
1osa	4cpv	24	1.0	1atna	1atr	15	5.3
1pfc	3hlab	22	1.0	1arb	4ptp	20	6.7
2cmd	6ldh	23	1.0	2pia	1fnr	18	7.4
2pna	1shaa	29	1.0	3rubl	6xia	18	8.0
1bbha	2ccya	21	1.0	2sara	9rnt	12	8.7
1c2ra	1ycc	23	1.0	3cd4	2rhe	25	9.3
1chra	2mnr	20	1.0	1aep	256ba	14	9.6
1dxtb	1hbg	19	1.0	2mnr	4enl	18	9.9
2fbjl	8fabb	22	1.0	1ltsd	1bova	19	10.9
1gky	3adk	24	1.1	2gbp	2liv	16	11.6
1hip	2hipa	19	1.1	1bbt1	2plv1	20	11.9
2sas	2scpa	17	1.1	2mtac	1ycc	15	12.1
1fc1a	2fb4h	19	1.1	1taha	1tca	16	12.6
2hpda	2cpp	18	1.1	1rcb	1gmfa	21	12.7
1aba	1ego	21	1.3	1saca	1ayh	14	12.7
1eaf	4cla	21	1.3	1dsba	2trxa	13	13.1
2sga	4ptp	21	1.4	1stfi	1mola	8	13.4
2hhma	1fbpa	13	1.4	1afna	1aoza	19	14.6
1aaj	1paz	31	1.6	1fxia	1ubq	18	15.3
5fd1	2fxb	21	1.7	1bgeb	1gmfa	12	15.4
1isua	2hipa	16	1.9	3hlab	2rhe	15	16.4
1gal	3cox	18	2.0	3chy	4fxn	14	17.3
1caub	1caua	18	2.0	2azaa	1paz	11	18.0
1hom	1lfb	19	2.4	1cew	1mola	10	18.1
1tlk	2rhe	24	2.4	1cid	2rhe	13	20.0
1omf	2por	17	3.7	1crl	1ede	17	20.0
1lgaa	2cyp	16	3.7	1sim	1nsba	12	20.0
1mioc	1minb	16	3.7	1ten	3hhrb	18	20.0
4sbva	2tbva	19	3.7	1tie	4fgf	14	20.0
8i1b	4fgf	18	4.1	2snv	4ptp	15	20.0
1hrha	1rnh	24	4.1	1gp1a	2trxa	17	20.0

* The 68 sequence-structure pairs of the benchmark, showing for each pair, the probe sequence s, the target fold p, the sequence identity percentage of the pair (as computed by GCG's (Genetics Computer Group, 1991) GAP program with default parameters), and the difficulty index (see text). The sequences are given by their PDB code. The mean sequence identity between s and p is 18.6% with a standard deviation of 4.4. The minimum sequence identity is 8% and the maximum is 31%. The average difficulty index is 7.4, with a standard deviation of 6.8.

There are 68 sequence-fold pairs in L, which are listed in Table I. This list provides a standard-of-truth to gauge which is the most compatible fold to each test sequence. The table shows the the sequence identity percentage and the difficulty index assigned to each pair. The difficulty index is computed as the average rank achieved by 7 standard comparison methods, including 6 substitution tables (the 5 substitution matrices shown in Table III plus a new, unpublished matrix developed in our group) and Bowie's 3D-1D profiles. If the rank of one particular pair was above 20, it was considered to be 20. The difficulty index tends to increase as the sequence identity percentage decreases. Table I shows that there are 12 test sequences for which even the simplest sequence-sequence comparison methods succeed in finding their most compatible fold. The presence of these "easy" pairs in the benchmark may be beneficial, because it provides a balancing factor in the assessment of a method. A good fold-recognition method should also be able to identify these pairs easily. Table I shows that the sequence identity percentages of these 12 "easy" pairs are all above 19%. The sequence identity percentages of the 17 "hardest" pairs are all below 20%. The other 39 pairs have sequence identity percentages ranging from 12% to 31%.

Figure 1 shows that L contains proteins of different sizes (in number of residues). In addition, the figure shows that there are a significant number of pairs where the difference of size (in number of residues) between probe and target is considerable. Table II lists the test sequences grouped by structural class. The table shows that the major superfamilies and domain superfolds are included in this benchmark[18,16]. The pairs represent divergent sequences from the same family (e.g. the globin pair 1dxtb-1hbg or the immunoglobulin pair 1fc1a-2fb4h) as well as unrelated sequences with similar folds (e.g. phycocyanin 1cpcl - colicin 1cola, both having the globin fold). The percentages of test pairs in each of the major structural classes (mostly-α, mostly-β, α/β and $\alpha+\beta$) are 19%, 36%, 29% and 10%, respectively. Except for the mostly-β class, the proportion of test sequences in each class is similar to the proportion of proteins of the same class in P. The β class is over-represented, mainly because of the presence of 8 test sequences with an immunoglobulin-like fold. However, this β class over-representation does not actually bias our test set (see legend of Table II).

2.2 Grading the overall performance

For each evaluated method we assess its sensitivity (how well the method performed in ranking the correct fold at the top) and its selectivity (how many false positives are obtained at the top ranks).

TABLE II. THE DISTRIBUTION OF THE TEST SEQUENCES IN THE DIFFERENT STRUCTURAL CLASSES*.

Class/fold	probe sequences
α: 13 pairs	
Globin-like	1dxtb 1cpcl
Cytochrome	1c2ra 2mtac
Helical bundle	1bbha 1bgeb
	1rcb 1aep
EF-hand	1osa 2sas
Other alpha	1hom 1lgaa 2hpda
α/β: 20 pairs	
TIM barrel	1chra 2mnr 3rubl
Hyrdolase	1crl 1taha
Thieredoxin	1aba 1dsba 1gp1a
Ribonuclease	1atna 1hrha
Open sheet	3chy 1ak3a
	1gky 2cmd
	1eaf 2gbp 1mioc
	2pia 1gal 1npx
Other : 3 pairs	
Mixed α and β	2hhma
Small	1hip 1isua

Class/fold	probe sequences
β: 25 pairs	
IG	1fc1a 2fbjl
IG-like	1cid 1pfc 1ten
	1tlk 3cd4 3hlab
Copredoxin	1aaj 1afna 2azaa
Virus	4sbva 1bbt1
Lectin-like	1saca
OB fold	1ltsd
Trefoil	1tie 8i1b
Trypsin	1arb 2sga 2snv
Lipocalin	1mdc 1mup
Propeller	1sim
Other beta	1caub 1omf
$\alpha+\beta$: 7 pairs	
UB fold	1fxia
cystatin	1cew 1stfi
SH2	2pna
other $\alpha+\beta$	2sara 1onc 5fd1

*The different structural classes and folds covered by the probe sequences of the benchmark. The number of test sequences in each class is roughly in the same proportion as that in the representative set of folds except for the β class, which is over-represented. Note that this over-representation is partly due to the abundance of immunoglobulin (IG) -like probes. However, out of the 8 IG-like folds, only 2 are immunoglobulins.

Sensitivity

For each probe sequence the evaluated method produces a list of structures, sorted by the compatibility score in decreasing order. The benchmark registers at what rank the expected fold of each probe sequence is found. The number of correct folds which were identified at rank 1, below rank 5 and below rank 10 are computed. In addition, the overall performance of a method is computed as $\frac{\sum 1/r_i}{|L|}$, where the sum is taken over all probes, r_i denotes the rank of the correct fold achieved by probe i and $|L|$ is the number of probes in the benchmark: 68. Thus, in total, we report 4 values for each method [c]. These empirical measures proved to reflect well the sensitivities of different methods.

[c] It may be the case that a particular sequence s has a fold which is similar to more than one chain in P, as some weak structural resemblances exist between the chains in P, e.g. several TIM barrels. These cases can be regarded as true positives. In order to avoid the possibility that another true positive be ranked above the expected p, an additional list of

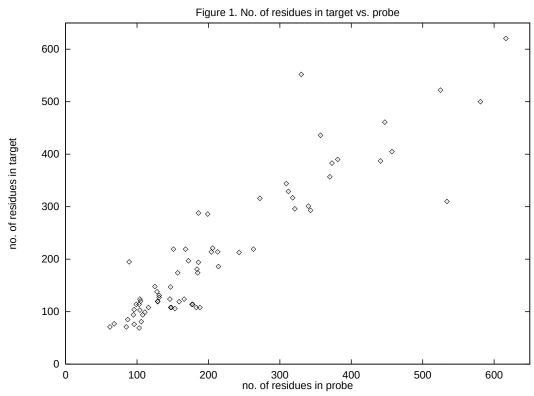

Figure 1: The length of the probe sequence and the target structure need not be equal. Each point represents one of the 68 test pairs in the benchmark. The horizontal axis gives the number of residues of the probe; the vertical axis gives the number of residues of the target. Notice that there are a number of pairs which contain a significantly different number of residues. The mean difference is 21% (minimum: 0%, maximum: 119%) with a standard deviation of 22.9.

Selectivity

When a probe sequence is compared to all the folds in the library, one obtains a list of scores, indicating the compatibility of each fold to the sequence. There will always be a rank-1 fold. This does not necessarily imply that the probe sequence has such a fold. Thus, one needs to be able to determine how significant this rank-1 fold is, or in other words, how (un)likely it is that this match arises by chance.

A valuable feature for a fold-recognition method is the potential to give a reliability level to a prediction. For example: "there is an 80% probability that this sequence has the globin fold". To this end we can express the result of an alignment in the form of a z-score (the number of standard deviations above the mean score). To rank the results, some methods normalize the raw scores of the alignments into a z-score. For such methods, the benchmark uses the z-score provided by the method. Other methods do not normalize the scores into a z-score (but rank the results using either the raw scores or some other normalized score). For these methods, the benchmark computes a z-score from the distribution of scores obtained in the alignment of s to

true positives for each probe s is also kept. These true positives would not lower the rank that p achieves, if they rank higher than p. This list contains 110 pairs of true positives, which may be used as additional test cases (the list is available from the authors).

each p in P. Having attached a z-score to each alignment, the benchmark computes a selectivity measure as follows. The z-scores of the first ranks in each of the 68 test cases are considered. The benchmark reports the number of pairs successfully recognized at 100%, 80% and 60% reliability levels, and their associated z-score values. For example, if we report 20 pairs at 80% reliability and a z-score of 3.0, this means that (i) there are 20 test cases which identified the correct fold at rank 1 having a z-score of 3.0 or higher, and that (ii) there are other 5 test cases where a false positive was found at rank 1, with z-scores higher than 3.0.

2.3 The evaluated methods

As described in the Introduction, a fold-recognition method has four main components. We have evaluated various fold-recognition methods which use different compatibility functions, different optimal alignment algorithms and different ranking and significance assessment procedures. In what follows we describe the different choices in each of the components which we have evaluated.

The compatibility functions

The compatibility functions that we have considered in the comparisons are shown in Table III. These include various sequence-sequence substitution tables, Bowie's 3D-1D profiles and two combined sequence-structure profiles. The functions compared are all functions which can be evaluated at each position of the alignment locally and independently of the aligned residues at other positions.

The Optimal Alignment Algorithms

The search method used in all the comparisons is the dynamic programming algorithm[10,11]. Dynamic programming is a good method to find an optimal alignment when the compatibility function can be evaluated at each position of the alignment independently of the aligned residues at other positions. The functions compared in this work all fall in this category. Finding an optimal alignment with a compatibility function that evaluates an alignment at more than one position at a time is an NP-complete problem[21]. Methods based on inter-residue interactions, overcome this problem either (i) by applying approximations, actually transforming their compatibility function to one that can be evaluated locally [22,6,20], or (ii) by using a heuristic optimal alignment algorithm[3,7,12].

TABLE III. THE COMPATIBILITY FUNCTIONS*.

name	description	ref.
	SEQUENCE SUBSTITUTION TABLES	
identity	1 for identical residues, 0 otherwise	
gcg	normalized pam250 matrix	GCG
pam250	point mutations in aligned families	[8]
blosum62	blocks of aligned motifs	[19]
gonnet	substitutions from database alignments	[9]
	3D-1D SCORES	
bowie	3D-1D profile	[1]
elofsson1	combined 3D-1D profile, using gcg matrix and areas	[20]
elofsson2	combined 3D-1D profile using blosum62 matrix, distances & areas	[20]

*The different compatibility functions used in this work. The "name" column refers to the name used in this work. "elofsson1" and "elofsson2" are new profile methods combining sequence-sequence information with structural information and are described in Elofsson et al.[20].

We have evaluated three dynamic programming algorithms: the *local*, the *global* and the *global-local* alignments. The "local" algorithm [11], finds the highest scoring aligned segment, allowing unpenalized-unaligned N- and C- termini both in the sequence and in the structure. The global alignment algorithm[10], allows at most two unaligned N- and C- termini without penalization but requires that at least one N-terminus segment and one C-terminus segment of either the sequence or the structure be either aligned or penalized. The "global-local" alignment algorithm *does not* penalize unmatched N- or C- termini segments in the probe sequence (as in the local alignment), but *does* penalize any gaps in the target structure (as in the global alignment with ends penalization). (We did not consider the global algorithm with ends penalization, nor the "local-global" algorithm. These two variants are of no interest as they both penalize any unaligned amino acids from the sequence. Thus, their applicability is limited to special cases.).

Gap penalty optimization Dynamic programming algorithms require the user to specify the values of the gap penalties to be used. Usually, gap penalties are specified as a gap opening penalty (O) and a gap extension penalty (E). The overall penalty for a gap in the alignment is given by $O + nE$, where n is the length of the gap. There is no single set of values which is best for different methods. Even for different sequences, the optimal gap penalties vary. In the present work, gap penalties are optimized for each method separately. Since there is no analytic method to calculate optimal penalties [23], the approach taken here is a brute-force search method. For each evaluated method, a range of gap penalties was tested using a reduced P set. The best combination of O and E was then used with the full size of P.

Ranking and Significance Assessment

There are two commonly used ways to consider the resulting score of an alignment. One is simply the raw score for compatibility of sequence to structure obtained from the alignment. The other is a statistical measure that indicates the probability that the raw score of the alignment was obtained by chance. One way to obtain such a measure is to analyze the raw scores of aligning to the same fold many sequences (of same length and composition), and compute their mean and standard deviation. Then, the result of the alignment of the native, non-randomized sequence is given as the number of standard deviations from the mean. This scoring procedure has the advantage of somewhat correcting for length and composition similarities between the sequence and the structure. A third score normalization procedure divides the raw scores by the logarithm of the length of the target's sequence[14].

To assess significance we follow the procedure described in the "Sensitivity" section above.

3 Results

We have evaluated the performance of fold-recognition methods using different compatibility functions, different alignment algorithms and different ranking procedures. In Elofsson et al.[20], different compatibility functions were evaluated using the local algorithm and a ranking procedure using the z-scores of randomized sequences. Other evaluations using the global algorithm and other ranking procedures will be presented elsewhere. From our evaluations we have found that the global-local algorithm performs better than the global or local algorithms. In addition, we have found that for several compatibility functions, when using the global-local algorithm, the ranking procedure based on the raw scores is comparable, if not superior, to the ranking procedure based on the z-scores (results not shown).

In this work we chose to show an interesting subset of our evaluations, for the purpose of illustrating the applicability of the benchmark. This subset includes the evaluations of different compatibility functions using the global-local algorithm and the ranking procedure based on the raw scores (i.e. the results are sorted and ranked by the raw score). Keeping both the alignment algorithm and the ranking procedure the same, we can systematically compare the performances of different compatibility functions. Note however, that the benchmark is independent of the choice of the alignment algorithm, the ranking procedure and the compatibility function used by a particular method.

TABLE IV. THE SENSITIVITY ASSESSMENT*.

COMPATIBILITY FUNCTION	**GAP PEN.** O	E	**IN RANK** **<10**	**<5**	**1**	**OVERALL SCORE**
bowie	1.8	0.20	43	35	25	0.455
identity	1.4	0.15	43	40	29	0.497
gcg	4.6	0.20	46	37	31	0.518
pam250	5.5	1.25	47	44	35	0.589
blosum	5.2	1.00	52	45	37	0.613
elofsson1	2.4	0.20	48	43	40	0.626
gonnet	10.8	0.60	51	**50**	40	0.664
elofsson2	3.2	0.20	**53**	**50**	**46**	**0.710**

*The results of the sensitivity assessment. The first column gives the name of the compatibility function used as described in Table III. The second column describes the optimal gap opening (O) and gap extension (E) penalties as obtained by the brute-force method described in the text. The next three numbers indicate the number of test probes that identified their target structure in ranks < 10, < 5 and $= 1$. The last column gives the overall score $(\sum 1/r_i)/68$. A perfect sensitivity would be: 68, 68, 68, with an overall score of 1.000.

Table IV shows the results of our sensitivity analysis of several compatibility functions using the global-local alignment algorithm and the ranking procedure based on the raw scores. The table shows the optimal gap penalties for each function as computed by a brute-force search (see Methods). It also shows the sensitivity performance of each method. The latter is described as 4 numbers: the number of test probes that identified the expected fold at rank 1, below rank 5 and below rank 10 and the overall performance $(\sum 1/r_i)/68$. Among the sequence substitution tables, the modern "gonnet"[9] matrix performs the best. The "identity" matrix performs the worst. However, to our surprise, its performance is not much worse than the "gcg" matrix. This may be due to the effectiveness of the global-local alignment algorithm, combined with the use of optimal gap penalties (see Discussion below). Using a local algorithm, the performance of the identity matrix is much worse than the other matrices (results not shown).

The new combined profile "elofsson2" performs significantly better than any other function tested so far. This compatibility function combines sequence-sequence information from the Blosum62[19] table with Bowie's 3D-1D profiles[1] and with other structural properties such as pairwise interactions (see[20] for details). This method assigns the correct fold in rank 1 in over two thirds of the test probes (46 out of 68). This is a significant improvement over the other functions. The best substitution table identifies the correct fold in rank 1 in only 59% of the test probes.

TABLE V. THE SELECTIVITY ASSESSMENT*.

COMPATIBILITY FUNCTION	**TRUE POSITIVES/Z-SCORE**							
	100%		**80%**		**60%**		**ALL RANK 1**	
bowie	4	3.32	6	2.93	7	**2.84**	25 (37%)	1.20
identity	2	2.61	2	2.61	26	1.35	29 (43%)	1.26
gcg	2	2.81	14	2.04	27	1.41	31 (46%)	1.30
pam250	9	2.43	9	2.43	35	1.72	35 (51%)	1.21
blosum62	8	3.74	10	**3.16**	35	2.18	37 (54%)	1.25
elofsson1	5	4.36	31	1.92	40	1.26	40 (59%)	**1.39**
gonnet	**13**	2.91	32	1.95	40	1.27	40 (59%)	1.36
elofsson2	9	**4.42**	**40**	1.60	**46**	1.18	**46 (68%)**	1.28

*The results of the sensitivity assessment. The first column gives the name of the compatibility function used as described in Table III. The following columns report the number of true positives and their associated z-score for 100%, 80% and 60% reliability levels (see text). The last column gives the number of true positives in rank 1, the percentage out of 68 test cases and their lowest z-score. A perfect selectivity would be 68 pairs at 100% reliability level, with a very high z-score.

Table V shows the selectivity assessment of the different compatibility functions. The table shows the number of true positives and their associated z-scores at reliability levels of 100%, 80% and 60% (see Methods). The last column shows the total number of test probes that identified the correct fold in rank 1 (same as in Table IV), the percentage (out of 68) and the lowest z-score of the true positives ranked 1. For example, at the 80% reliability level, "elofsson2" identifies the correct fold at rank 1 for 40 test probes. These have z-scores above 1.60. However, there are 8 other probes which identified the wrong fold at rank 1 with z-scores above 1.60. Table V shows that the selectivity of no method is as yet very good. The best method identifies only 19% (13 out of 68) test probes at a reliability level of 100%. The total number of correctly identified folds lies below a reliability level of 68%.

4 Discussion

We present here a benchmark to assess the performance of fold-recognition methods. The benchmark allows a systematic comparison of different methods. The benchmark is independent of the particular choices in each of the components of a fold-recognition method and can aid in the analysis of the strengths and weaknesses of the four steps involved in fold recognition.

The advantage of using a benchmark such as the one presented here, is that the set of test sequences (S) and the library of known folds (P) were derived in an unbiased way and represent varied sequence-structure compatibility problems with insignificant sequence similarity, which cover homogeneously many different families. This is important when building a benchmark, because a method that works best at one particular type of fold could score higher using a test set in which that particular fold is over-represented. As long as all fold classes are present, and no fold is over-represented, any representative dataset of the known structures can serve as the set P. Also, the set of sequence-structure pairs can be selected to contain any number of pairs, as long as each structural family is equally represented. We have found that a larger test set does not increase the discriminative power of the benchmark[20]. However, since our P set was built, proteins with novel folds have been deposited in the PDB. Thus, we estimate that using the current release of the PDB, L and P could be about 10% larger. [d]

The performance assessment of this benchmark addresses two issues: sensitivity and selectivity. The four empirical measures of sensitivity assessment used in this benchmark are quite consistent and correlated. We have found that if a method has an overall performance over 0.5, then the overall performance alone is a good measure. For a lower overall performance, the other measures provide some additional information.

The performance of a method based on dynamic programming does not only depend on the compatibility function used, but also on the gap penalties used. Instead of applying rules of thumb in assigning their values, for each method compared, we have carried out a brute-force search to determine the optimal gap penalties. In order to avoid overfitting the parameters, the test sequences of the benchmark could be split into two sets: a training set for optimizing gap penalties and a test set to evaluate performance. Alternatively, an independent training set, containing pairs different from those in the benchmark, could be used. The values of the optimal gap penalties obtained using

[d] It should be noted however, that if one would like to use P as a library of folds for an actual prediction using a particular method, the following procedure to extend P is suggested. Test each sequence s from the PDB against P. If the highest ranking $p \in P$ corresponds to the actual most compatible fold for s, and its score is significantly high, then proceed to the next s. If however, the score is not significant or the correct fold is not ranked first, then add s to P. This procedure expands P to an ideal size for the particular method's capabilities, ensuring that every sequence of known structure is either in P or a similar fold to it can unambiguously be found. On the other hand, it keeps P at a reasonable size, which has the advantage of saving computer time. This extension of P is important to avoid the possibility of a method that *could* recognize the correct fold, but *fails* to do so, simply because the correct fold was absent in the dataset used.

different training sets (results not shown) are very similar to the ones obtained using the full benchmark, and the performances (using the latter sets) are also very similar to those reported above. Hence, the results shown in this work, and in particular, the relative performance of the different methods, do not reflect overfitting.

We have applied the benchmark to different fold-recognition methods which differ in at least one of their components. In Elofsson et al.[20], a local alignment algorithm was used to compare different compatibility functions using a ranking procedure based on the z-scores. Other choices in each of the components have also been evaluated. In this work we showed the evaluations of methods using different compatibility functions, but using the same alignment algorithm and the same ranking procedure. Our results show that the blosum and gonnet tables perform better than pam250. The relative performance between the pam250, blosum62 and gonnet tables obtained in this work are consistent to several previous comparison reports[13,14,19,20]. These works use different test sets, either local or global alignments, search for alignment accuracy or method sensitivity. The relative performance of the three structural profiles studied in this work is also consistent to the findings of Elofsson et al.[20]. The results also demonstrate that the 3D-1D profiles combined with sequence information and pairwise interactions are superior to classical sequence-sequence comparison. The best performing compatibility function evaluated so far is "elofsson2", a new combined profile to be described in [20].

The results shown in Table IV require further analysis. The number of correctly identified folds is surprisingly high, in particular for the sequence-sequence tables. Even the identity matrix has a performance not much worse than the "gcg" table. This is outstanding, as the pairs used in the benchmark have low sequence similarity. We attribute this enhanced performance to the combination of three factors: (i) the use of optimal gap penalties, (ii) the application of the global-local algorithm and (iii) the use of the ranking procedure based on the raw scores. As both the global-local algorithm and the raw scores ranking procedure are not as widely known as other algorithms and ranking procedures, in what follows, we analyze their properties in more detail.

The superior performance of the global-local algorithm.

There are two common variations of dynamic programming: the "global" [10] and "local"[11] alignment algorithms. A third, less widely known variation is the "global-local" alignment (see Methods). Each of these alignment algorithms was devised for one particular type of comparison, and each has both pros and cons. In what follows, the pros and cons are evaluated in the context of fold recognition, i.e. we will refer to the alignment of a sequence (a probe) to a structure (a target).

Using a local alignment algorithm, a relatively short alignment, matching a segment of sequence to some super-secondary motif, may produce a relatively high score. In addition, as the size of the target to which the sequence is being aligned increases, the probability of finding such a high score also increases. Thus, for a given sequence, there can be a number of false positives (incompatible structures) scoring higher than the true positive (a compatible structure). In the global algorithm this problem appears with less severity as the requirement of having at least two N- or C- termini either aligned or penalized, adds a constraint in the alignment alternatives. However, the problem still persists to some extent as the above constraint allows the global algorithm to choose *which* two termini are to be considered. Our evaluations demonstrate that the global and local algorithms perform similarly (results not shown). Several studies have also suggested that global alignments are not inferior to local alignments (e.g.[24,13,14]). Indeed, some existing fold-recognition methods prefer the global alignment over the local (e.g. [22]).

Another problem associated with the global and (especially with the) local algorithms is that as the alignment of a probe sequence with a structure can consist of a relatively short segment of the structure, it may not be very useful for building a model for the sequence. The segment can be composed of some structural fragment which may be meaningless when considered in isolation. The ability to build a model for a probe sequence is the ultimate goal of fold recognition.

The global-local algorithm is based on the principle that the compatibility to one structure should cover the structure globally. The global-local algorithm requires that the unpenalized termini (if any) appear exclusively in the sequence, accounting for every position of the structure, either as an aligned or as a penalized-unaligned position. This is a strong constraint imposed in the search algorithm which has a positive effect, and somewhat overcomes some of the limitations of the global and local algorithms. There are two factors involved. First, as all the positions in the structure are accounted for in the alignment, the possibility of obtaining higher scores for relatively short, local matches is reduced. Second, the tendency of obtaining higher scores for larger structures is also reduced; if the fold is larger than the probe sequence, more gaps need be included, and the score of this match would be lower. In addition, allowing unpenalized termini in the probe does not bias the algorithm towards targets of similar length (see below). This is especially important for probe sequences which may contain more than one domain. The net effects of using the global-local algorithm are a lower rate of high scoring false positives, and in some cases, a higher alignment accuracy. In summary, the global-local alignment produces alignments that cover the full target, and at the same time allows unaligned-unpenalized termini in the probe. This is a desirable property for fold recognition.

The global-local algorithm is *not* aimed at identifying a compatibility in only short portions of a structure. One way to identify such submotifs using the global-local algorithm is to partition the library of known folds into compact domains and subdomains (i.e. "minimally recognizable units"), and place each of these units as separate entries in the library, along with the full fold. The partitioning process can be carried out using any of the automated procedures developed especially for this purpose. This has the advantage of using structural knowledge in the partitioning processes instead of allowing the local algorithm to choose a fragment which may not be a structurally meaningful unit, merely for the purpose of maximizing the compatibility score. Not allowing the dynamic programming algorithm to make such a blind choice is one of the strengths of the global-local algorithm. [e]

At first sight it could appear that the enhanced performance of the global-local algorithm is mainly due to length discrimination. However, an analysis of the rank versus the size difference between probe and target showed no correlation, i.e. the best ranks were not necessarily achieved by the pairs having the smallest differences (results not shown). Furthermore, tests with probe sequences highly padded at both termini with random sequences demonstrated that the performance of the global-local algorithm is still superior to those of the global and local algorithms (results not shown).

The ranking procedure. As the raw scores of the alignments using the local or global algorithm are dependent on the length of the target, methods using these algorithms require normalization of the raw scores to account for this dependency. In contrast, in the global-local algorithm, the raw scores of the alignments are less length dependent and thus more reliable than those of the global or local algorithms.

In systematic evaluations using several compatibility functions, we have observed that (i) the local and global algorithms perform better when the results are ranked using normalized scores than when using the raw scores. Normalizing by dividing the raw scores by the ln of the length of the target is superior to the z-score normalization (in agreement with[14]). (ii) The global-

[e] In the evaluations of this work, the library of known folds contains full chains, without any additional partitioning. If we had partitioned the folds in the above way, the performances of the various compatibility functions using the global-local algorithm would be even better.

local algorithm with the raw scores ranking procedure performs comparably, if not better, than with the z-scores ranking procedure. The ln normalization performs the worst for the global-local algorithm.

The relatively poor performance of the z-scores ranking procedure in conjunction with the global-local algorithm is surprising. By analyzing the individual results of each probe sequence with either method, we observed that in several cases, although the raw score rank of the correct fold was at the very top, its z-score rank was inside the "twilight zone" of this normalization procedure (the region where it is impossible to distinguish random scores from the significant ones). A more detailed analysis of the different ranking procedures is out of the scope of this paper. We should note here that for the global-local algorithm, the poor performance of the ranking procedure based on the z-scores may be attributed in part to the fact that the methods were evaluated using optimal gap penalties (for a poor choice of gap penalty values, the z-scores ranking procedure may show some advantage over the raw scores ranking procedure). It could also be attributed to some bias that may exist in our training or test sets, in the library of folds, or in the compatibility functions evaluated.

Limitations of the proposed benchmark.

Assessment of alignment accuracy is not covered in this work, and is a topic for a different study (a thorough analysis of alignment accuracy has been carried out for sequence-sequence comparisons by Vogt & Argos[13] and for 3D-1D profiles by Wilmanns & Eisenberg[25]). Other computational aspects that a benchmark can grade are computer time and space requirements, aspects of practical importance. When one has many probe sequences, a faster, possibly less sensitive method could be used in a first screening, and then a slower, more sensitive one for the cases where the first method did not succeed to unambiguously assign a fold.

There is the possibility that our choice of proteins imparts a bias to the benchmark. We have attempted to extract proteins from the data bank in an unbiased way, by an all-against-all structural comparison of the protein data bank. However, different proteins could be extracted for the benchmark using different thresholds for sequence and structural similarity (e.g.[26,27,13]). Obviously, a different choice might give different results. A more serious matter is that proteins with known structures are a biased subset of all proteins. For example, the PDB contains few membrane proteins, few glycoproteins and few fibrous proteins. Thus our benchmark is not useful in assessing fold recognition work on these other protein types. Despite the limitations, the use of this benchmark, or a different one, may aid in understanding the merits of the different aspects involved in fold recognition.

The sequences and tables used in this work are available from the authors by e-mail at fischer@ewald.mbi.ucla.edu.

Acknowledgments

D.F was supported in part by a grant from the Program in Molecular Biology and Mathematics. A.E. was supported by the Swedish Research Council for Engineering Sciences. This work was supported by the Department of Energy cooperative agreement DE-FC03-8TER60615.

References

1. J. U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.
2. M.J. Sippl and S. Weitckus. Detection of native like models for amino acid sequences of unknown three dimensional structure in a data base of known protein conformations. *Proteins*, 13:258–271, 1992.
3. D.T. Jones, W.R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.
4. A. Godzik, A. Kolinski, and J. Skolnick. Topology fingerprint approach to the inverse folding problem. *J. Mol. Biol.*, 227:227–238, 1992.
5. C. Ouzounis, C. Sander, M. Scharf, and R. Schneider. Prediction of protein structure by evaluation of sequence-structure fitness. Aligning sequences to contact profiles derived from 3D structures. *J. Mol. Biol.*, 232:805–825, 1993.
6. M. Wilmanns and D. Eisenberg. Three-dimensional profiles from residue-pair preferences: Identification of sequences with β/α-barrel fold. *Proc. Natl. Acad. Sci. (USA)*, 90:1379–1383, 1993.
7. S.H. Bryant and C.E. Lawrence. An empirical energy function for threading protein sequence through folding motif. *Proteins*, 16:92–112, 1993.
8. M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. *A model of evolutionary change in proteins. In: Atlas of Protein Sequence and Structure 5:3.* Nat. Biomedical Research Found. , Washington, D.C., 1978,345.
9. G.H. Gonnet, M.A. Cohen, and S.A. Benner. Exhaustive matching of the entire protein sequence database. *Science*, 256:1433–1445, 1992.
10. S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol.*

Biol., 48:443–453, 1970.
11. T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.
12. R. Lathrop and T.F. Smith. A branch and bound algorithm for optimal protein threading with pairwise amino acid interactions. In *Proc. 27th Hawaii Int. Conf. on System Sciences*, 5:365–376, Los Alamitos, 1994.
13. G. Vogt, T. Etzold, and P. Argos. An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *J. Mol. Biol.*, 249:816–831, 1995.
14. W.R. Pearson. Comparison of methods for searching protein sequence databases. *Prot. Sci.*, 4:1145–1160, 1995.
15. B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.
16. D. Fischer, C.J. Tsai, and R. Nussinov. A 3-D Sequence-Independent Representation Of The Protein Data Bank. *Prot. Eng.*, 1995. In press.
17. F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer, M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures. *J. Mol. Biol.*, 112:535–542, 1977.
18. C.A. Orengo, D.T. Jones, and J.M. Thornton. Protein superfamilies and domain superfolds. *Nature*, 372:631–634, 1994.
19. S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. (USA)*, 89:10915–10919, 1992.
20. A. Elofsson, D. Fischer, D.W. Rice, S. Le Grand, and D. Eisenberg. A study of combined structure-sequence profiles. 1995. In preparation.
21. R. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-Complete. *Prot. Eng.*, 1995.
22. H. Flockner, M. Braxenthaler, P. Lackner, M. Jaritz, M. Ortner, and M.J. Sippl. Progress in fold recognition. 1995. To appear.
23. S.F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, 219:555–565, 1991.
24. M.A. McClure, T.K. Vasi, and W.M. Fitch. Comparative analysis of multiple protein-sequence alignmnet methods. *Mol. Biol. Evol.*, 2:572–592, 1994.
25. M. Wilmanns and D. Eisenberg. Inverted protein folding by the residue pair preference profile method. *Prot. Eng.*, 1995. In press.
26. C. A. Orengo et al. Identification and classification of protein fold families. *Prot. Eng.*, 6:485–500, 1993.
27. L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.