FOLD			Threading method					
			unfiltered		randomly filtered		filtered	
structure pdb code	sequence pdb code	% sequence identity	*ASns*	*ASn4*	*ASns*	*ASn4*	*ASns*	*ASn4*
1alc	153l	11	0.0	0.0	0.0	0.0	0.0	0.0
2lzm	153l	12	0.0	19.6	0.0	19.6	0.0	19.6
1bgc	1alu	16	73.1	100.0	73.1	100.0	50.5	100.0
2baa	1am7A	14	0.0	37.7	0.0	37.7	22.6	22.6
9rnt	1aqzA	26	0.0	8.3	0.0	0.0	0.0	0.0
1rec	1auiB	26	14.9	45.9	0.0	41.9	33.8	43.2
2sns	1bcpD	2	0.0	24.1	13.8	41.4	13.8	37.9
2cyp	1bgp	21	0.0	0.0	35.7	49.1	22.3	35.7
1lis	1br0	6	0.0	0.0	0.0	0.0	0.0	0.0
2had	1brt	17	41.5	71.7	40.6	77.4	30.2	58.5
351c	1c52	18	0.0	0.0	17.6	32.4	17.6	17.6
5cpv	1cll	35	62.0	88.0	22.0	72.0	62.0	88.0
1rcb	1cnt3	12	74.3	100.0	33.8	33.8	51.4	74.3
1dhr	1cydA	19	19.5	50.4	19.5	71.7	25.7	64.6
5cytR	1cyj	19	71.4	71.4	71.4	71.4	71.4	71.4
1pkp†	1dar	15	TIME-OUT		0.0	94.7	31.6	94.7
4tgl	1din	14	0.0	32.8	19.7	68.9	23.0	42.6
1aba	1erv	14	0.0	68.4	7.9	50.0	7.9	50.0
5tmnE†	1ezm	31	0.0	3.9	3.9	5.5	3.9	33.1
1cde	1fmtA	16	14.3	38.8	11.2	59.2	20.4	38.8
3adk	1gky	16	33.8	62.0	43.7	85.9	52.1	70.4
1f3g	1hcz	18	TIME-OUT		0.0	0.0	0.0	0.0
1mbd	1ithA	15	13.5	30.3	13.5	30.3	13.5	30.3
2ca2	1kopA	34	0.0	36.8	50.0	76.5	64.7	76.5
1byh	1led	11	0.0	16.1	3.4	16.1	0.0	12.6
1ifc	1eal	21	17.8	89.0	61.6	100.0	56.2	100.0
1ubq	1lxdA	11	32.3	80.6	19.4	100.0	32.3	80.6
1cewI	1molA	20	0.0	25.6	0.0	17.9	33.3	33.3
1apa	1mrj	28	11.6	41.9	31.8	67.4	40.3	65.1
2end	1mtyG	4	0.0	23.1	0.0	23.1	0.0	23.1
2mhr	1nfn	8	0.0	0.0	0.0	0.0	0.0	0.0
7rsa	1onc	27	0.0	40.7	79.6	100.0	0.0	85.2
1atu	1ovaA	30	31.8	47.0	27.3	62.1	42.4	42.4
2hpr	1pfh	35	70.8	100.0	72.9	91.7	50.0	100.0
1bp2	1poc	27	0.0	0.0	0.0	0.0	0.0	0.0
5nll	1rcf	23	0.0	23.4	45.3	95.3	70.3	95.3
1yat	1rot	27	31.9	83.0	31.9	91.5	61.7	100.0
3chy	1srrA	26	13.7	100.0	69.9	100.0	47.9	100.0
3est	1svpA	12	0.0	31.1	0.0	57.8	0.0	31.1
2act	1theA	28	0.0	8.7	27.5	48.7	36.2	57.5
2mcm	1tvdB	9	9.7	22.6	0.0	41.9	0.0	41.9
8dfr	1vdrA	24	29.9	68.7	29.9	44.8	40.3	61.2
1hoe	1wkt	6	8.3	41.7	0.0	16.7	0.0	83.3
5fd1	1xer	31	0.0	0.0	0.0	0.0	0.0	0.0
1lec	2ayh	11	0.0	38.2	0.0	47.2	13.5	39.3
256bA	2ccyA	17	28.6	77.9	0.0	19.5	28.6	58.4
1tie	2il1b	11	0.0	10.0	0.0	24.0	0.0	28.0
4fgf	2il1b	14	0.0	21.7	0.0	45.7	0.0	8.7
2cpl	2nul	30	0.0	20.0	35.0	35.0	35.0	35.0
1s01	2pkc	38	TIME-OUT		16.2	86.5	51.4	91.0
2aak	2uce	33	0.0	43.7	0.0	43.7	0.0	20.8
1plc	7paz	25	0.0	26.7	0.0	0.0	0.0	0.0
Average			15.8	42.6	22.9	52.9	27.6	52.1

Table 1: Comparison of three threading methods: the UNT, the RFNT and the FNT method. "TIME-OUT" indicates threadings that did not converge within the time limit. The † indicates structures defined as multidomain by SCOP.

Hybrid Fold Recognition: Combining Sequence Derived Properties with Evolutionary Information.

Daniel Fischer
Dept. of Math and Computer Science
Faculty of Natural Science,
Ben Gurion University
Beer-Sheva 84015, Israel
e-mail: dfischer@cs.bgu.ac.il

Recent assessments of structure prediction have demonstrated that (i) although fold recognition methods can often identify remote similarities when standard sequence search methods fail, the score of the top-ranking fold is not always significant enough to allow a confident prediction; (ii) the use of structural information such as secondary structure increases recognition accuracy; (iii) modern sequence-based methods incorporating evolutionary information from neighboring sequences can often identify very remote similarities; (iv) there is no one single method that is superior to other methods when evaluated over a wide range of targets, and (v) extensive human-expert intervention is usually required for the most difficult prediction targets. Here, I describe a new, hybrid fold recognition method that incorporates structural and evolutionary information into a single fully automated method. This work is a first attempt towards the automation of some of the processes that are often applied by human predictors. The method is tested with two fold-recognition benchmarks demonstrating a superior performance. The higher sensitivity and selectivity enable the applicability of this method at genomic scales.

1 Introduction

Protein fold recognition aims to assign each new amino acid sequence to the known three-dimensional fold which it most closely resembles. The assignment is carried out by searching a library of known structures for a compatible fold. Fold-recognition methods have demonstrated their capabilities in computer-aided assessment experiments such as CASP[1] as well as in fully automated assessment experiments such as CAFASP-1[2]. In the former, fold-recognition programs coupled with human intervention were able to correctly predict the folds of proteins of (then) unknown structure. In the latter, the performance of the methods was not as good, but still it was superior to sequence-comparison methods such as PSI-BLAST[3]. CAFASP-1 demonstrated that no single approach was markedly superior to the others evaluated when considered across the entire range of targets. In some cases, exploiting evolutionary information from neighboring sequences resulted in the correct fold identification (e.g. [4,5]); in other cases, the use of structural information such as predicted versus observed secondary structure (e.g. [6,7]) allowed recognition of the correct fold.

Since the appearance of PSI-BLAST, several fold-assignment methods exploiting the evolutionary information in the sequence databases have been developed. These include approaches using neighbors of the target sequence, neighbors of the folds in the fold library and both. The evolutionary information is usually compiled in the form of a profile or a HMM[5]. The results of the application of these new methods to complete genomes also demonstrated that some of the predictions from fold-assignment methods are not detectable by sequence-based methods[1,8], and conversely, that sequence-based methods sometimes identify distant relationships that fold-assignment methods do not detect[9,10]. Current sequence-based methods succeed in these cases because of their incorporation of evolutionary information from neighboring sequences, whereas traditional fold-assignment methods do not exploit this information to the same extent. It is thus clear that a new generation of hybrid, fold-assignment methods, like the one presented in this work, which combine structural and evolutionary information should result in even more sensitive methods.

Another clear conclusion of recent fold-recognition assessment experiments was that, in many cases, although the correct fold was identified at rank-1, its score was not significant; in such cases, human intervention was required to discriminate true from false positives. This aspect is of particular interest for structural genomics. Automated approaches for fold recognition are essential if the wealth of data in genomes is to be exploited (e.g.[11,12,9] and [8] for a recent review). For genomic fold assignment to work it is necessary that folds be assigned with a high degree of confidence. That is, a method needs to discriminate correct match scores from incorrect ones. A major conclusion from CAFASP-1 was that improvements in this aspect are required to allow a much wider applicability of automated fold-recognition methods at a genomic scale. The new method presented here is a first attempt to automate some of the procedures a human predictor often applies when trying to discriminate true from false positives.

In this work I describe a new, hybrid fold-recognition method that combines evolutionary information from neighboring sequences with structural information. This new method is based on principles similar to those of the previously developed fold-recognition method SDP[6] and is aimed to overcome some of the limitations described above. The new method is fully automated and is available for the academic community at: http://www.cs.bgu.ac.il/bioinbgu. The sensitivity and selectivity of the method was tested using two standard benchmarks, and the results show that significant improvements have been achieved.

2 Methods

The new, hybrid fold-recognition method is a consensus method that is composed of five components. These components are based on an extension of the fold-recognition method SDP[6] which computes sequence-structure compatibility using sequence-derived predictions and the so-called "global-local" dynamic programming algorithm for alignment[6,13]. The sequence-structure compatibility in SDP is computed as:

$$g(i,j) = f(i,j) + w_j \times h(i,j) \qquad (1),$$

where g relates the information at position i of the target sequence with position j of the fold and is composed of two parts, f and h. f corresponds to one of the five sequence-structure compatibility functions described below. f reflects the similarity of a position in the target sequence with a position of the assigned fold, using either a standard 20 x 20 sequence comparison matrix, a multiple alignment of homologous sequences, a sequence profile built from the multiple alignment or other sequence-structure compatibility functions[14]. h is a function that scores the compatibility of the sequence-derived properties of position i of the target, with the observed structure of position j of the fold. The only sequence-derived property used here is the predicted secondary structure[15,16]. w_j is a position dependent empirical weight. h depends not only on the compatibility of observed versus predicted secondary structure, but also on the per-position reliability given in the secondary structure prediction.

I chose compatibility functions of the form of g because it has been demonstrated that the use of predicted secondary structure in fold recognition increases its sensitivity and selectivity[6]. In previous works the predicted secondary structure was computed by PHD[15] using homologous sequences to the target, compiled using a single BLAST[17] iteration on the SWISSPROT database. The first source of improvement of the current method is due to the increase in the predicted secondary structure accuracy obtained by compiling the homologous sequences from the larger "nr" database, and using the newer PSI-BLAST program[3]. Further improvements in the secondary structure prediction are likely to contribute to additional improvements in fold-recognition performance[6] (e.g. by replacing PHD by the reportedly more sensitive PsiPred[18] program). This option is currently being evaluated.

2.1 The five components

Each of the five components of the new method use a different f function (see Table I), each exploiting the sequence and evolutionary information differently. The first component is termed GONP and considers only the amino

acid sequence of the target. The compatibility is measured using the sequence comparison matrix of Gonnet et al.[19]. That is:

$$f_{GONP}(i,j) = Gonnet(target[i], fold[j]) \quad (2)\ ,$$

where $target[i]$ denotes the i-th amino acid of the target sequence, and $fold[j]$ denotes the amino acid at position j of the fold.

TABLE I. The f compatibility functions used in each of the components.

Symbol	Information used		Comments
	for the target	for the fold	
GONP	a.a. sequence	a.a. sequence	Gonnet matrix.
GONPM	multiple alignment	a.a. sequence	Gonnet matrix.
PRFSEQ	PSI-BLAST profile	a.a. sequence.	
SEQPPRF	a.a. sequence	PSI-BLAST profiles.	
SEQPMPRF	multiple alignment	PSI-BLAST profiles.	

The compatibility between predicted and observed secondary structures is measured via the h function in Eq. (1), and thus it is not specified here.

The second component is termed GONPM and uses a multiple alignment of sequences homologous to the target. The compatibility is measured by:

$$f_{GONPM}(i,j) = \sum_{k=1}^{20} ma_i[k] \times f_{GONP}(k,j) \quad (3)\ ,$$

where ma denotes the frequencies of each amino acid at position i of the multiple alignment. GONP and GONPM are essentially the same methods as those previously described[6]. The difference is that the multiple alignment used in GONPM (and in the other components below) is now compiled using PSI-BLAST and the "nr" database. This more information-rich multiple alignment is the second source of improvement in the current method over the previously described methods.

The third component termed PRFSEQ replaces the multiple alignment (ma) of GONPM with a profile (P_{target}) computed by PSI-BLAST:

$$f_{PRFSEQ}(i,j) = P_{target}[i, fold[j]] \quad (4),$$

where $P_{target}[i, fold[j]]$ represents the value of the profile of the target sequence at row i, column $fold[j]$.

The last two components of the new method, SEQPPRF and SEQPMPRF use PSI-BLAST generated profiles (P_{fold}) for the folds in the fold-library. SEQPPRF compares the single target sequence with each profile in the fold-library:

$$f_{SEQPPRF}(i,j) = P_{fold}[j, target[i]] \quad (5)\ .$$

SEQPMPRF compares a multiple alignment of sequences homologous to the target (as in GONPM) with each profile in the fold-library:

$$f_{SEQPMPRF}(i,j) = \sum_{k=1}^{20} ma_i[k] \times P_{fold}[j,k] \quad (6)\ .$$

Other alternatives considered the use of multiple alignments for each fold in the library. Because of the non-homogeneity in the number of homologous sequences for the different folds in the library, these alternatives proved not to be sensitive enough. Yet another alternative, which has been investigated by Godzik's group[4], but not considered here, is to match multiple alignments from both the target and the folds.

With these five f functions, we have five different g compatibility functions. Each of these g functions are used in separate fold recognition runs: the target sequence information and the predicted secondary structure are compared to each of the folds in the library, and the result of each run is a ranking of the folds based on their sequence-structure compatibility scores. That is, for run i (for i equal to GONP, GONPM, PRFSEQ, SEQPPRF and SEQPMPRF), each fold j in the library receives two numbers: $r_{i,j}$ and $s_{i,j}$, where $r_{i,j}$ denotes the rank that fold j achieved in run i, and $s_{i,j}$ is its corresponding score.

The individual $s_{i,j}$ scores are computed as follows. Each sequence-structure alignment produces a "raw" score which represents the sequence-structure compatibility. For each run, the distribution of the raw scores of the folds in the library were used to compute z-scores. The z-score measures the number of standard deviations that the raw score lies above the mean score.

2.2 The consensus method

The consensus method takes all the $r_{i,j}$ and $s_{i,j}$ and computes for each fold j in the library a consensus score, c_j as, $c_j = \sum_{i=1}^{5} s_{i,j}/r_{i,j}$. To produce the final ranking, the c_j's are sorted from best scores to worst. Notice that c_j could be computed differently, possibly with different weights for each component tuned using for example a neural network. This possibility is currently being considered. The rationale behind the consensus method is to allow for relatively weak predictions that are consistent among the various components to receive a more confident score. The fact that different methods using various types of information rank the same fold at the top can be an indication of the validity of the prediction. In addition, as will be shown below, in some cases, only one of the components is able to score its rank-1 prediction highly; the consensus method will in most such cases also place this high score prediction at rank-1 with a significant score (see below).

2.3 The Benchmarks.

Each of the five components and the consensus methods were evaluated here using two benchmark tests. In these benchmarks, the 3D structures of the

probe sequences are actually known, but are ignored during the test.

One of the benchmarks used[13] consists of a library of 301 known target structures and a set of 68 probe sequences which cover a wide range of structural classes and folds. This benchmark was originally published in the pages of these proceedings in 1996, and since then it has been extensively used to evaluate the performance of various fold-recognition methods (e.g.[6,20,4]). I refer to this benchmark as the 68-Benchmark.

The second benchmark is based on the targets used in the CAFASP1 evaluation[2], and is referred here as the CAFASP1-benchmark. The fold library used with this benchmark contains about 2000 different folds, representing a minimally redundant set of structures and domains taken from the Protein Data Bank (PDB[21]) available by mid 1998. The CAFASP-1 benchmark consists of 21 targets selected from the CASP3[22] competition, none showing any sequence similarity to the available proteins of known structure by mid 1998; only for one of the targets could PSI-BLAST identify a similar structure. The lists of targets and of folds considered to be the correct hits for each target are included in the CAFASP-1 web page at http://www.cs.bgu.ac.il/dfischer/cafasp1/cafasp1.html.

For each of the target sequences in each benchmark, the evaluated methods scan the library of known folds and produce a ranked list of compatibilities. Both benchmarks register the rank at which a correct fold of each target sequence is assigned by the method. The number of correct folds identified at rank 1 are registered and the overall performance score of a method is computed as $S^* = \frac{\sum 1/r_i}{n}$, where the sum is taken over all targets, r_i denotes the rank of the correct fold achieved by probe i and n is the number of targets in the benchmark. This scoring system is similar to the one used in previous benchmarks[13,2], and its rationale is as follows: suppose a program always has the correct answer within the top i ranks; if only a single answer is desired, then, on average, the correct fold will be predicted with probability $1/i$. S^* equals 1.0 for perfect fold assignment (and < 0.01 for random assignment).

In addition to the sensitivity of the methods (i.e. the number of correct predictions), I have analyzed their selectivities, again following the same approach used in CAFASP-1. For a given threshold score s, selectivity is defined as the number of true positives at rank-1 with scores better than s. To this end, for each component I compiled its rank-1 predictions, and set three threshold scores, Th1 , Th2 and Th3 (different for each component), corresponding to the scores of the first, second and third rank-1 wrong predictions (i.e. false positives), respectively. Finally, I counted the number of rank-1 true positives with scores above Th1, Th2 and Th3. Confidence thresholds help the user of an automated method to determine the reliability of a prediction.

3 Results

Table II is a summary of the evaluation using our first benchmark, for each of the five components and for the consensus method.

TABLE II. 68-BENCHMARK EVALUATION

COMPONENT	SENSITIVITY		SELECTIVITY		
	= 1	S*	trues/Th1	trues/Th2	trues/Th3
GONP	48	0.77	27/4.7	36/3.7	39/3.4
GONPM	52	0.80	35/3.7	39/3.7	43/3.4
PRFSEQ	52	0.82	31/5.1	41/3.2	41/3.1
SEQPPRF	52	0.83	39/4.0	43/3.5	46/3.3
SEQPMPRF	57	0.87	46/3.8	47/3.5	50/3.4
consensus	58	0.89	48/12.0	51/11.6	55/10.6

The first column gives the symbol of the compatibility function used as described in Table I. The first five rows correspond to the individual components, and the last row corresponds to the consensus method. The second column indicates the number of targets that identified their correct fold at rank 1. The third column shows the overall score S^* (see text). A perfect sensitivity would be 68, with an S^* score of 1.00. The selectivity columns indicate the number of targets identified ("trues") with scores above Th1, Th2 and Th3, respectively. For example, the highest scoring rank-1 false positive of the consensus method had a score of 12.0, and 48 rank-1 true positives had scores > 12.0. The GONP and GONPM results shown here correspond to those previously published.

Table II shows that the most sensitive and selective component in this benchmark was SEQPMPRF. However, although it identified the largest number of correct folds at rank-1, the other components succeeded to identify the correct fold in rank-1 for a number of targets for which SEQPMPRF failed. Consequently, it is possible for the consensus method to improve over the individual performances of its components. Besides the improvement in sensitivity (correct rank-1 identification), probably the most dramatic improvement of the consensus method is in its selectivity; it scored over 80% (48/58) of its rank-1 predictions with a score > 12.0.

Table III shows the results for each of the 21 targets in the CAFASP1-benchmark. For each target and component are shown the score and fold identified at rank-1, followed by the rank and fold of the first correct fold. If a correct prediction was obtained at rank 1, then the fold identified at rank-1 is the same as the one listed as the "1st true" (for some targets more than one "correct" fold exist). The table shows that for 8 targets (T0043, T0054, T0059, T0063, T0071.1, T0071.2, T0071 and T0080) no component was close to identify the correct fold. These 8 targets include some of the most difficult targets in CASP3 and CAFASP-1. However, for the other targets in Table

III, the correct fold was identified by at least one component either at rank-1 or at the top 5 ranks; for seven targets (T0044, T0046, T0074, T0079.1, T0079, T0081 and T0083.1) almost all components identified the correct fold at rank-1.

TABLE III. THE INDIVIDUAL RESULTS ON THE TARGETS OF THE CAFASP-1 BENCHMARK.

	T0043			T0044			T0046		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.3	1lvl	15 8atc	4.6	2pol	2 1eps	3.9	1cid	1 1cid
SEQPM	5.5	2fx2	12 1pil	6.6	1eps	1 1eps	3.8	1cid	1 1cid
PRFSEQ	4.2	1opc	21 1aw0	4.8	1eps	1 1eps	5.5	1cid	1 1cid
SEQPPRF	4.4	1cgm	6 8atc	5.2	1eps	1 1eps	4.8	1cid	1 1cid
SEQPMPRF	4.0	1fcd	6 1afi	7.1	1eps	1 1eps	5.0	1cid	1 1cid
consensus	5.8	2fx2	>20	20.5	1eps	1 1eps	21.0	1cid	1 1cid

	T0053			T0054			T0059		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.1	1ak1	1 1ak1	4.0	1pox	>25	4.2	1hng	>25
SEQPM	4.0	1ak1	1 1ak1	4.0	1scu	>25	3.0	1auu	12 1vie
PRFSEQ	4.0	1aoa	4 1ak1	4.5	1pox	>25	5.4	1bib	1 1bib
SEQPPRF	6.5	1aq6	3 1ak1	4.3	1lz1	>25	3.8	1kdu	>25
SEQPMPRF	N.A.		N.A.	4.2	1rhd	18 1lbu	2.9	1hng	6 1bib
consensus	13.2	1ak1	1 1ak1	9.9	1pox	>20	10.4	1hng	2 1bib

	T0063.1			T0063.2			T0063		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.3	1pkn	12 8shf	4.8	1amy	4 1ah9	4.0	1lcl	9 1eip
SEQPM	2.9	1csq	2 1ckb	4.0	1csp	1 1csp	3.6	1gof	9 1umu
PRFSEQ	3.0	1ckb	1 1ckb	3.8	1iyv	4 1csp	4.1	2bpk	3 1lts
SEQPPRF	4.1	1tit	17 1bib	3.8	1tuc	4 1ah9	4.1	1pbk	5 1eip
SEQPMPRF	3.0	1bib	1 1bib	4.1	1sro	1 1sro	3.6	2bpk	24 1sro
consensus	5.9	1csq	4 1ckb	8.8	1amy	3 1sro	9.9	2bpk	12 1lts

	T0067			T0071.1			T0071.2		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	3.6	1an3	3 1lla	4.8	1lfo	7 1cid	3.9	1ecm	>25
SEQPM	4.3	1mfn	1 1mfn	3.9	1cpa	2 1tit	4.0	1ecm	>25
PRFSEQ	5.0	1piv	3 1ttg	4.5	1ifc	4 1jrh	4.5	5paa	10 2prf
SEQPPRF	4.4	1an3	3 1cid	3.8	1eal	2 1bgl	4.3	1lfb	8 3pmg
SEQPMPRF	3.1	1an3	2 1cid	4.0	1gof	1 1gof	3.5	1ayy	>25
consensus	12.7	1an3	3 1mfn	9.4	1eal	5 1gof	8.0	1ecm	>20

	T0071			T0074			T0079.1		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.7	1grl	17 3hla	4.1	4icb	1 4icb	3.7	1apl	1 1apl
SEQPM	5.3	1grl	5 1frt	4.6	4icb	1 4icb	3.2	1apl	1 1apl
PRFSEQ	4.1	1bak	6 2prf	8.6	1trf	1 1trf	3.9	1fjl	1 1fjl
SEQPPRF	4.0	1ppr	2 3pmg	5.5	1a4p	1 1a4p	3.1	1r69	1 1r69
SEQPMPRF	4.1	1yas	10 2prf	4.7	1a4p	1 1a4p	3.3	1lcc	1 1lcc
consensus	12.1	1grl	7 3pmg	11.5	1trf	1 1trf	8.2	1apl	1 1apl

	T0079			T0080			T0081		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.5	2pna	2 1tns	4.6	1tml	7 1fmt	4.7	1etu	2 1jdb
SEQPM	3.1	1neq	1 1neq	5.6	1ndk	>25	5.8	1jdb	1 1jdb
PRFSEQ	3.5	1lea	1 1lea	5.2	1ndk	>25	5.8	1jdb	1 1jdb
SEQPPRF	4.3	1hcr	1 1hcr	4.7	2stv	>25	14.4	1jdb	1 1jdb
SEQPMPRF	3.8	1aoy	1 1aoy	4.2	2mnr	>25	20.0	1jdb	1 1jdb
consensus	7.1	1aoy	1 1aoy	12.8	1ndk	>20	48.0	1jdb	1 1jdb

	T0083.1			T0083			T0085		
	rank 1	1st true		rank 1	1st true		rank 1	1st true	
SEQP	4.5	1lmb	1 1lmb	2.9	1fha	9 1lmb	5.5	1fgj	1 1fgj
SEQPM	4.3	1lmb	1 1lmb	3.1	1lmb	1 1lmb	4.4	1fcd	5 1fgj
PRFSEQ	5.1	1lmb	1 1lmb	4.6	1lmb	1 1lmb	5.0	2mta	7 1fgj
SEQPPRF	3.7	1r69	1 1r69	3.8	1a0b	7 1r69	4.7	2cy3	14 1fgj
SEQPMPRF	4.2	1r69	1 1r69	4.1	1a0b	3 1r69	N.A.		N.A.
consensus	15.9	1lmb	1 1lmb	8.2	1lmb	1 1lmb	10.2	1fcd	3 1fgj

The utility of the consensus method is illustrated in several targets. One example is T0046, in which all five components identified the same fold at rank-1, albeit with low scores. The consistency of the predictions is reflected in the very high score of the consensus method. Similar results are observed for T0044, T0083.1 and T0083. Another example of the utility of the consensus method is given by target T0081. Only when using the fold-library profiles, the score of the rank-1 fold is very high (in SEQPPRF and SEQPMPRF). On the other hand, the fold-library profiles actually harm the correct prediction for target T0053, where the SEQPPRF and SEQPMPRF methods place the correct fold at rank 3. Nevertheless, the consensus method exploits the information from the five components and assigns the correct fold at rank-1.

Another way to compute a consensus score is to base it on the fold type of the hits, rather than the individual pdb entries. This can be useful when different pdb entries of the (correct) fold type are found by a method, but none is hit with a high score. This type of consensus scoring would increase the score for targets T0074, T0079 and T0083.

Table IV is a summary of the results shown in Table III. Tables III and IV show that the CAFASP-1 benchmark is a much more demanding test for fold recognition than the 68-benchmark. Consequently, the scores (S^*) achieved in this benchmark are considerably lower. Interestingly, in this benchmark the sensitivities of PRFSEQ and SEQPMPRF are very similar (the differences are not likely to be significant given the relatively small size of the benchmark). The performance of SEQPPRF and SEQPMPRF components is not as high probably due to the presence of a number of targets for which their "correct" folds had no or few sequence neighbors. The GONP and SEQPPRF components were the worse, indicating that the contribution of the multiple alignment used in GONPM and SEQPMPRF is significant. In addition, the

S^* score of the consensus method did not improve over the scores of the components. However, Table IV shows that there exists a significant improvement in the selectivity of the consensus method. While most of the individual components identified only one or two correct folds with scores above the first false positive (Th1), the consensus method identified five. This is a significant achievement that has important implications for automatic fold prediction.

TABLE IV. CAFASP1-BENCHMARK EVALUATION

COMPONENT	SENSITIVITY		SELECTIVITY		
	= 1	S*	trues/Th1	trues/Th2	trues/Th3
GONP	6	0.42	1/4.8	1/4.8	1/4.7
GONPM	11	0.61	2/5.6	2/5.5	2/5.3
PRFSEQ	10	0.57	4/5.2	5/5.0	5/5.0
SEQPPRF	7	0.46	1/6.5	4/4.7	4/4.7
SEQPMPRF	10	0.56	2/6.5	3/4.7	4/4.2
consensus	10	0.53	5/12.8	5/12.7	5/12.1

See footnotes to Table II. The total number of probes in this benchmark is 21, and thus, a perfect sensitivity is 21, with $S^* = 1.00$. For comparison, the scores for the rank-1 of the two CASP3 folds which corresponded to new folds received scores below Th3.

4 Discussion

The fold-recognition methods evaluated here incorporate evolutionary and structural information. The evolutionary information corresponds to homologous sequences compiled by PSI-BLAST[3], and the structural information corresponds to the matching of predicted and observed secondary structures. The inclusion of evolutionary information results in improved sensitivities and selectivities. The evolutionary information is exploited by (i) PHD and the GONPM and GONPMPRF components, which use a multiple alignment of sequences homologous to the target; by (ii) the PRFSEQ component, which uses a sequence profile for the target sequence, and by (iii) the SEQPPRF and SEQPMPRF components, which use profiles for the folds in the library.

The consensus methods combines five different components, each using the evolutionary information in a different way. The significant increase in the selectivity of the consensus method is a step towards the wider applicability of fold recognition in an automatic fashion. The consensus method exploits the strengths of each component, and is an attempt to automate some of the procedures a human would apply when using fold-recognition programs. For some targets the use of sequences homologous to the target may be beneficial

because the latter "bridge" the distance of the target to its compatible fold. However, for other cases, the homologous sequences may increase that distance. Similarly, while for some targets, the use of sequences homologous to the compatible fold may be beneficial, for others, it can be detrimental. However, in most of the cases, the use of neighboring sequences for both the target and the fold (as in SEQPMPRF) appears to contribute to a better performance.

The new consensus method and its five components were evaluated here using two benchmark tests. In both tests significant improvements were observed over previously evaluated methods. The method presented here is a first attempt to combine valuable evolutionary information (obtained by the PSI-BLAST program) with structural information in a fold-recognition method. Various directions of improvements are possible, and some of these are being currently investigated. Further research is necessary to establish better estimates of confidence thresholds as well as to further automate some of the processes used by human experts when interpreting the output of the various fold-recognition approaches.

The improvements shown here allow to better recognize distantly related proteins. As more sequences and structures are deposited in the databases, more genome proteins will find distant relatives of known structure. However, there is a non-negligible percentage of genomic orphan ORFs, or ORFans, which have no sequence neighbors[23]. For these, the inclusion of evolutionary information can not help, because ORFans, by definition, have no sequence neighbors. Thus, to be able to assign folds for these ORFans, improvements in the classical sequence-structure compatibility functions that fold-recognition methods use are required.

One limitation of the present work is that none of the benchmarks used here evaluates alignment accuracy. Thus, some of the correct rank-1 predictions can produce poor models which should not be credited points. Extension of the CAFASP-1 benchmark to evaluate alignments is work in progress[24]. Finally, a further test of the new method will be to assign folds to complete genomes, and to use it in future CASP and CAFASP assessments.

Acknowledgments: Thanks to Erez Karpas for his assistance in setting up the *bioinbgu* server and to the anonymous referees for their helpful comments.

References

1. CASP3. Critical Assessment of Protein Structure Prediction Methods (CASP), Round III. *Proteins*, 1999. To appear; see also http://Prediction center.lnl.gov.
2. CAFASP1. Critical assessment of fully automated protein structure prediction methods. *Proteins, Special Issue*, 1999. See http://www.cs.bgu.ac.il/ dfischer/cafasp1/cafasp1.html.
3. S.F. Altschul, Madden T.L., A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database

search programs. *Nucleic Acid Res.*, 25:3389–3402, 1997.
4. L. Rychlewski, B. Zhang, and A. Godzik. BASIC .
5. K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.
6. D. Fischer and D. Eisenberg. Protein fold recognition using sequence-derived predictions. *Prot. Sci.*, 5:947–955, 1996.
7. L. A. Kelley, R. M. MacCallum, and M. J. E. Sternberg. Recognition of remote protein homologies using three-dimensional information to generate a position specific scoring matrix in the program 3d-pssm. *RECOMB*, 1999. In the press.
8. D. Fischer and D. Eisenberg. Predicting Structures for Genome Sequences. *Curr. Opin. Struc. Biol*, 9:208–211, 1999.
9. M. Huynen, T. Doerks, F. Eisenhaber, C. Orengo, S. Sunyaev, Y. Yuan, and P. Bork. Homology-based fold predictions for *Mycoplasma genitalium* Proteins. *J. Mol. Biol.*, 280:323–326, 1998.
10. S.A. Teichmann, J. Park, and C. Chothia. Structural assignments of the *Mycoplasma genitalium* proteins show extensive gene duplications and domain rearrangements. *Proc. Natl. Acad. Sci. (USA)*, 95, 1998.
11. D. Fischer and D. Eisenberg. Assigning folds to the proteins encoded by the genome of *Mycoplasma genitalium*. *Proc. Natl. Acad. Sci. (USA)*, 94:11929–11934, 1997.
12. L. Rychlewski, B. Zhang, and A. Godzik. Functional insights from structural predictions: analysis of the Escherichia coli genome. *Protein Science*, 1999. In the press.
13. D. Fischer, A. Elofsson, D.W. Rice, and D. Eisenberg. Assessing the performance of inverted protein folding methods by means of an extensive benchmark. *Proc. 1st. Pacific Symposium on Biocomputing*, pages 300–318, January 1996. http://www.mbi.ucla.edu/people/fischer/BENCH/benchmark1.html.
14. A. Elofsson, D. Fischer, D.W. Rice, S. Le Grand, and D. Eisenberg. A study of combined structure-sequence profiles. *Folding & Design*, 1:451–461, 1996.
15. B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.
16. B. Rost. TOPITS: Threading one-dimensional predictions into three-dimensional structures. *Proc. Conf. Intelligent Systems in Molecular Biology, ISMB-95*, pages 314–321, 1995.
17. S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment tool. *J. Mol. Biol.*, 215:403–410, 1990.
18. D.T. Jones. Psi-pred. 1999. In press.
19. G.H. Gonnet, M.A. Cohen, and S.A. Benner. Exhaustive matching of the entire protein sequence database. *Science*, 256:1433–1445, 1992.
20. D.T. Jones. Genthreader: An efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.*, 287:797–815, 1999. In press.
21. 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.
22. CASP3. Third community wide experiment on the critical assessment of techniques for protein structure prediction. *Asilomar, USA.*, December 1998. http://Prediction center.lnl.gov/casp3.
23. D. Fischer and D. Eisenberg. Finding Families for Genomic ORFans. *Bioinformatics*, 1999. In the press.
24. D. Fischer. MaxSub: A measure ofr quality assessment of Protein Structure Predictions. In preparation.

FOLDING NUCLEI IN 3D PROTEIN STRUCTURES

O.V. GALZITSKAYA
Biomolecular Engineering Research Institute, 6-2-3 Furuedai, Suita, Osaka 565-0874, Japan (on leave from the Institute of Protein Research, Russian Academy of Sciences, Pushchino, Moscow Region, 142292 Russia)

A.V. SKOOGAREV, D.N. IVANKOV, A.V. FINKELSTEIN
Institute of Protein Research, Russian Academy of Sciences, Pushchino, Moscow Region, 142292 Russia

This paper presents and analyzes the results of several new approaches to the problem of finding the folding nucleus in a given 3D protein structure. Firstly, we show that the participation of residues in the hydrophobic core and the secondary structure of native protein has a rather modest correlation with the experimentally found Φ values characterizing the participation of residues in the folding nuclei. Then we tried to find the nuclei as the free energy saddle points on the network of the folding/unfolding pathways using the branch-and-bound technique and dynamic programming. We also attempted to estimate the Φ values from solving of kinetic equations for the network of protein folding/unfolding pathways. These approaches give a better correlation with experiment, and the estimated folding time is consistent with the experimentally observed rapid folding of small proteins.

1 Introduction

An understanding of the mechanism of protein folding can help in design of new proteins, in understanding of correct and wrong folding of proteins, in attempts to predict protein structure from sequence.

A key event in protein folding is the formation of the folding nucleus [1-4]. This "nucleus" is *un*stable: it corresponds to the transition state (TS), i.e., to the free energy maximum at the folding/unfolding pathway (or, the better to say, to a saddle point at the free energy landscape covered with the network of such pathways).

So far, there is only one, very difficult experimental method to identify the folding nuclei in proteins: to find the residues whose mutations affect the folding rate by changing the TS stability as strongly as that of the native protein [5].

Several approaches have been recently suggested for the theoretical search of folding nuclei in proteins. The first is based on a search for a set of highly conserved residues having no obvious functional role [6,7]; however, this approach can give only a common part of the nuclei existing in homologous proteins. The second approach is based on the correlation between the participation of residues in the folding nucleus and their fluctuations in partly unfolded stationary states [8] or in native proteins [9]. The third, more direct approach is based on all-atom molecular dynamic simulations of protein unfolding [10-13]. However, these simulations need extremely denaturing conditions (500°K, etc.) to be completed in a reasonable time. Therefore the TS found for such extreme unfolding can be rather different from that existing for folding [14].