LiveBench-1: Continuous benchmarking of protein structure prediction servers

JANUSZ M. BUJNICKI,[1] ARNE ELOFSSON,[2] DANIEL FISCHER,[3] AND LESZEK RYCHLEWSKI[1]

[1]Bioinformatics Laboratory, International Institute of Molecular and Cell Biology (IIMCB), 02-109 Warsaw, Poland
[2]Stockholm Bioinformatics Center, Stockholm University, 10691 Stockholm, Sweden
[3]Department of Bioinformatics and Computer Science, Ben Gurion University, Beer-Sheva 84105, Israel

(RECEIVED September 26, 2000; FINAL REVISION November 10, 2000; ACCEPTED November 20, 2000)

Abstract

We present a novel, continuous approach aimed at the large-scale assessment of the performance of available fold-recognition servers. Six popular servers were investigated: PDB-Blast, FFAS, T98-lib, GenTHREADER, 3D-PSSM, and INBGU. The assessment was conducted using as prediction targets a large number of selected protein structures released from October 1999 to April 2000. A target was selected if its sequence showed no significant similarity to any of the proteins previously available in the structural database. Overall, the servers were able to produce structurally similar models for one-half of the targets, but significantly accurate sequence-structure alignments were produced for only one-third of the targets. We further classified the targets into two sets: easy and hard. We found that all servers were able to find the correct answer for the vast majority of the easy targets if a structurally similar fold was present in the server's fold libraries. However, among the hard targets—where standard methods such as PSI-BLAST fail—the most sensitive fold-recognition servers were able to produce similar models for only 40% of the cases, half of which had a significantly accurate sequence-structure alignment. Among the hard targets, the presence of updated libraries appeared to be less critical for the ranking. An "ideally combined consensus" prediction, where the results of all servers are considered, would increase the percentage of correct assignments by 50%. Each server had a number of cases with a correct assignment, where the assignments of all the other servers were wrong. This emphasizes the benefits of considering more than one server in difficult prediction tasks. The LiveBench program (http://BioInfo.PL/LiveBench) is being continued, and all interested developers are cordially invited to join.

Keywords: Automated protein structure prediction; benchmarking; meta server; CAFASP; LiveBench

It is crucial for a molecular biologist to know the three-dimensional (3D) structure of a protein to gain a detailed understanding of its function. In most cases, the structure is a useful guide for the rationalization and planning of mutations. Because it is often difficult or impossible to determine a structure experimentally, one may attempt to model the structure in silico. For those proteins whose homologs are available in the structural database, a method of choice for obtaining a 3D model is to infer it on basis of the general rule that homologous proteins exhibit highly similar structures (Overington et al. 1990). The main difficulty with the homology modeling approach is selecting the most appropriate template out of a large set of proteins with known structure and then generating an accurate sequence-structure alignment. This problem is often referred to as fold recognition or threading (Jones et al. 1992; Taylor 1997; Torda 1997).

In the last few years, much progress has been made in the field of fold recognition, partially the result of rigorous external (Abagyan and Batalov 1997; Brenner et al. 1998; Park et al. 1998) and community-wide benchmarking experiments, that is, critical assessment of structure prediction

Reprint requests to: Dr. Leszek Rychlewski, Bioinformatics Laboratory International Institute of Molecular and Cell Biology, ul. ks. Trojdena 4, 02-109 Warsaw, Poland; e-mail: leszek@bioinfo.pl; fax: 48-22-668-5288.

Article and publication are at www.proteinscience.org/cgi/doi/10.1110/ps.40501.

(CASP) (Moult et al. 1999) and critical assessment of fully automated structure prediction (CAFASP) (Fischer et al. 1999). CASP is an assessment of the predictive value of various strategies used by different groups of experts. The strategies lately have become more and more complex; they involve the use of various prediction programs and validation methods, often combined with the development of in-house fold-recognition tools. The experience of the group of experts has substantial effect on the quality of the predicted models. The main outcome of CASP is thus a community internal evaluation of approaches. However, an outsider cannot determine to what extent the success of some group in CASP was the result of application of more sensitive tools or of manual intervention by the experts.

For an outsider, CASP can answer the question "Who should I contact to help me to predict the structure of my protein?" CAFASP is an alternate project intended to evaluate only methods without intervention from an expert. Thus, the outcome of CAFASP allows researchers to address the question "What method should I use to obtain a structural model of my protein?" Both CASP and CAFASP effectively correspond to blind prediction experiments, because the structures of the prediction targets are unknown at the time the predictions are made. However, one of the drawbacks of both CASP and CAFASP is the limited number of protein targets (a few dozen targets in the latest experiments) used to assess the quality of groups or prediction methods.

LiveBench, the project presented in this paper, follows the CAFASP ideology, but it aims to overcome the problem of a limited number of targets by selecting a large number of prediction targets through weekly scanning of the protein structure database PDB (Bernstein et al. 1977) for novel proteins. It is somewhat less rigorous than CAFASP, because the structures of the targets are known at the time the predictions are made; thus, LiveBench it is based on the assumption that the evaluated servers do not use any hidden features that would direct the prediction toward the correct answer. Yet the immediate availability of the structures allows almost immediate assessment of the predictions.

LiveBench assesses publicly available fold-recognition servers. It is aimed to provide biologists with valuable information regarding the server's performance. One of the main outcomes of this study is an outline of a general strategy for biologists (non-experts in protein structure prediction) who, in their study of protein structure–function relationships, would like to take advantage of the variety of publicly available fold-recognition servers.

Results

Collected data

The LiveBench program resulted in 125 targets submitted during the period between October 29, 1999 and April 6, 2000. The target proteins were divided into 30 easy targets (Easy category) and 95 difficult targets (Hard category). Per definition, easy targets were correctly predicted by PDB-Blast (see Materials and Methods), with an expectation (e) value (or E value) of $< 1e-5$. PDB-Blast was used for the division of targets because it is very similar to PSI-Blast (Altschul et al. 1997), a popular method for protein sequence comparison. PSI-Blast is often used as reference for evaluating fold-recognition methods. Of the 95 difficult targets, 5 had no structurally similar proteins in PDB at the time they were released, leaving 90 targets in the Hard category. The DALI server (Holm and Sander 1998a) was used for this search.

Evaluated servers

The LiveBench program involved the following six protein structure–prediction servers:

(1) PDB-Blast (http://bioinformatics.burnham-inst.org/pdb_blast/) is based on the PSI-Blast program (Altschul et al. 1997). Before the fifth iteration, the sequence profile is saved and used to scan the database of proteins with known structure (the PDB database from the previous week). A locally installed version of the service was used. This server is maintained by one of the authors of this article.

(2) FFAS (http://bioinformatics.burnham-inst.org/FFAS/) (Rychlewski et al. 2000) is based on comparing sequence profiles with each other. Profiles are generated for protein families in a different way than they are in PSI-Blast, but PSI-Blast is used to collect the proteins of a family. This server is co-developed by one of the authors of this article.

(3) T98-lib (http://www.cse.ucsc.edu/research/compbio/HMM-apps/model-library-search.html) (Karplus et al. 1999) is based on an iterative hidden Markov model method for constructing protein family profiles. The server evaluated in the work is not the newest structure prediction tool offered by the group. The newer server Sam-T99 (http://www.cse.ucsc.edu/research/compbio/HMM-apps/T99-query.html) is reported by the authors to outperform the older version in most tests. (This server was not available during the LiveBench period investigated in this work.) The old T98-lib server is no longer supported.

(4) 3D-PSSM (http://www.bmm.icnet.uk/servers/3dpssm/) (Kelley et al. 2000) is based on a threading approach using 1D and 3D profiles coupled with secondary structure and solvation potential. The newer version of this

server (not evaluated in this experiment) uses a fold library, which is automatically updated every week.

(5) GenTHREADER (http://www.psipred.net/) is based on a combination of various methods, including sequence alignment with structure-based scoring functions; it uses a neural network–based jury system to calculate the final score for the alignment (Jones 1999). The newer server mGenTHREADER is currently available and will be tested in the next iterations of LiveBench. GenTHREADER is now also being transformed to use a weekly updated fold library.

(6) INBGU (http://www.cs.bgu.ac.il/~bioinbgu/form.html) is a combination of five methods that exploit sequence and structure information in different ways; the combination produces one consensus prediction of the five. It uses predicted versus observed secondary structure and sequence profiles both for the target and for the folds in the library. The results used here are taken from the consensus prediction only (Fischer 2000). This server is maintained by one of the authors of this article.

The first three methods mentioned above use only sequence information to find the appropriate template in the library of proteins with known structure, whereas the other three methods include the structural information available for the templates in the target-template fitting functions.

Evaluation of sensitivity

The entire analysis presented in this paper focuses only on the first model (hit) returned by the prediction servers. Table 1 shows the number of correct hits found by each server.

Correct hits were determined on the basis of the definitions of correct and false predictions given by the two evaluation methods: alignment-dependent MaxSub and alignment-independent LGscore (see Materials and Methods). Table 1 shows the results in two parts, for the categoreis Easy and Hard (results for all targets are available at the web site http://BioInfo.PL/LiveBench/).

When looking at the number of total correct hits in the Easy target category, PDB-Blast and FFAS perform best in that they identify almost all of the targets. Targets in this category should not present problems for any of the evaluated servers, and the substantial differences in the obtained scores are most likely the result of differences in the fold library used by the servers. PDB-Blast uses a library of folds that is updated every week; thus, it has access to the most complete structural information. Because, from the point of view of the user, the most current database is very important, we deliberately reward PDB-Blast for that in our ranking. When looking at the number of total correct hits in the more challenging and much larger category of difficult targets (Hard), INBGU and 3D-PSSM find the largest number of correct hits, corresponding to between 20% (MaxSub) and 40% (LGscore) of the hard targets.

Table 1. *Sensitivity of the fold-recognition servers*

	Hits[a]		Score-sum[b]		First[c]		Only[d]	
	MaxSub	LG score	MaxSub	LG score	MaxSub	LG score	MaxSub	LG score
Easy (30)								
FFAS	29 (97%)	29 (97%)	23.5	26.8	12	12	0	0
3D-PSSM	21 (70%)	24 (80%)	13.1	19.3	2	1	0	0
GenTHREADER	23 (77%)	24 (80%)	16.1	21.0	6	5	0	0
INBGU	24 (80%)	25 (83%)	17.2	20.6	8	6	0	0
PDB-Blast	30 (100%)	29 (97%)	22.2	25.8	5	5	0	0
T98-lib	22 (73%)	23 (77%)	13.9	17.7	1	2	0	0
Hard (90)								
FFAS	13 (14%)	22 (24%)	7.8	26.0	8	10	4	3
3D-PSSM	18 (20%)	29 (32%)	8.8	33.9	6	7	3	2
GenTHREADER	10 (11%)	25 (28%)	6.4	29.2	4	11	1	1
INBGU	18 (20%)	37 (41%)	11.1	36.7	8	15	1	6
PDB-Blast	4 (4%)	19 (21%)	2.9	18.7	2	8	1	2
T98-lib	12 (13%)	24 (27%)	8.0	28.2	6	9	2	2

Targets are divided into 2 categories: Easy with 30 targets and Hard with 90 targets with known fold (represented in the PDB). In each column, two values are shown: the left corresponds to the evaluation using alignment-dependent MaxSub, and the right corresponds to the evaluation using alignment-independent LGscore.

[a] The total number of correct hits obtained for each prediction server (FFAS, 3D-PSSM, GenTHREADER, INBGU, PDB-Blast, T98-lib).

[b] Values indicate the total sum of relative scores obtained for each model. The relative scores for each prediction is equal to the score obtained by the prediction server divided by the score obtained by the DALI structure comparison server (if available). In case of LGscore, the negative logarithm of the original score is used to calculate relative scores (for LGscore, a positive value closer to 0.0 indicates a better prediction).

[c] The number of targets where a particular server generated the best model.

[d] The number of times a prediction server was the only server from the evaluated group that was able to return a correct prediction. This happened only in the case of hard targets, never for easy targets.

Additional information can be gained by looking at the servers' total performance and distinguishing between better and worse models, and rewarding the better models. Both evaluation methods were used to calculate relative scores for every prediction by using the model obtained from the DALI server as a reference. If the model returned by DALI was of low quality (i.e., if the structural similarity between the target and the best template was low), then the predictions for this target are more difficult, and they were given higher weight. In some cases, the predicted models were very similar to the model obtained from the DALI server (i.e., "better," according to our evaluation methods). For example, for one of the hard targets, 1om2_A, T98-lib returned a model with 38 superimposable Cα atoms with a MaxSub score of 0.37, whereas the model generated by DALI contained 33 superimposable Cα atoms with a MaxSub score of 0.33. In such cases, the highest score obtained by the prediction server was used for scaling. Using this procedure and summing over all targets, we obtained the values reported under Score-sum in Table 1. The evaluation using model quality as shown in the Score-sum column confirms the general trends found after evaluating the total number of hits. PDB-Blast and FFAS score highest for the easy targets, whereas INBGU and 3D-PSSM remain the top two servers in the category of hard targets. It is also evident, however, that the models of FFAS are evaluated higher than the models of PDB-Blast. The model-based evaluation seems also to favor INBGU over 3D-PSSM in the Hard category.

One more piece of information can be gained by asking how many times a server produced the best model. The answer to this question is provided in the column labeled First. The results are similar to those obtained using the Score-sum evaluation. However, this column shows that servers that do not perform best in the analysis based on total hits are also capable of producing models that are better than those obtained using other servers. This is one of several reasons that potential users should take advantage of the variety of prediction servers. To distinguish between better and worse models, a user can take into account such additional information as the expected conservation of active site residues.

To show whether all servers can under some circumstances provide crucial predictions, we have calculated the number of times each prediction server was the only server to produce a correct model.

The results obtained are shown in the column titled Only in Table 1. It is evident that all servers have at least one "crucial" prediction. This column shows the contribution of each prediction server (the added value of each server) to the community of prediction servers. The values demonstrate that it would be unwise to neglect any of the servers, especially in the case of hard targets. (In the case of easy targets, all values in this column are 0, which means that for each target at least two prediction servers produce a correct model.)

Evaluation of specificity

There are two important aspects of a prediction, namely, its accuracy and its confidence. To assess the server's confidence level, we evaluated the specificity of the reported results by using the scores provided by the servers. Table 2 shows the number of correct predictions with confidence higher than five selected cutoffs. The values of the cutoffs are specific to every server and correspond to the values where a limited number (0, 2, 5, 10, and 20) of incorrect models (with lower quality than allowed) was encountered.

Looking at the specificities calculated for all targets, we find that GenTHREADER and INBGU scored highest in the first column, where no false positive predictions are allowed. In this column, most servers have very low values, which means that they have at least one very confident (i.e., high-scoring) but wrong prediction. When considering more than one false positive, MaxSub and LGscore obtain slightly different results. INBGU scores highest when using the alignment-independent evaluation of LGscore, whereas FFAS performs best according to the more rigorous MaxSub-based ranking.

Analyzing the specificity on hard targets only shows that the 90 targets used here are indeed hard.

Table 1 shows that the most sensitive servers generated a good model for <20% of the hard targets and identified the

Table 2. *Specificity of the fold-recognition servers*

	MaxSub					LGscore				
Errors:	#0	#2	#5	#10	#20	#0	#2	#5	#10	#20
All										
FFAS	13	28	30	33	36	13	26	31	37	38
3D-PSSM	19	21	21	27	29	21	22	27	30	31
GenTHREADER	24	26	28	28	30	24	26	30	33	36
INBGU	23	24	26	29	34	26	31	35	42	43
PDB-Blast	13	27	27	32	33	13	27	27	34	39
T98-lib	7	19	24	28	31	7	22	30	32	36
Hard										
FFAS	1	2	3	4	7	1	2	6	9	10
3D-PSSM	3	4	8	10	10	5	5	9	11	12
GenTHREADER	4	5	6	6	7	4	5	8	10	13
INBGU	3	4	5	8	11	6	9	14	18	20
PDB-Blast	1	2	2	2	3	1	2	2	8	11
T98-lib	1	2	5	9	10	1	3	9	13	15

Predictions were sorted on the basis of descending significance as reported by the server. The table shows the number of correct predictions that could be obtained with reported significance (score) higher than the significance (score) of the first error (#0), and with cutoffs allowing 2, 5, 10, and 20 (#2, #5, #10, and #20) false predictions. The specificity is calculated using all (120 + 5) targets and using only hard (90 + 5) targets (5 targets with novel fold were included in this calculation). Both evaluation methods, alignment-dependent MaxSub and alignment-independent LGscore, are used to divide predictions into wrong and correct. FFAS and PDB-Blast obtained 100% specificity (using MaxSub) on the easy targets, whereas other servers made more than two errors with higher scores than the correct predictions (data not shown).

correct fold for ~40% of the cases. However, identification of these hard targets comes with a cost: low specificity. Less than a handful of correct results were obtained at scores higher than the first false positive. Table 2 shows that among the hard targets, the scores of most of the correctly identified targets fall within the "twilight zone" of the servers. For example, among the 20 highest scoring models of 3D-PSSM, only 10 were correct, corresponding to a specificity of 50% (a MaxSub-based evaluation). The fold-recognition specificity is obviously higher, but still only few predictions can be made without expecting a substantial number of errors.

We emphasize that very strict evaluation criteria have been applied here. First, when using a sequence-dependent evaluation method, we considered the result wrong if the sequence-structure alignment was not good enough, even if the correct fold was identified. Second, we considered only the highest scoring result for each server and target. However, in many cases, the score difference between the highest scoring result and the second highest is very small. In a number of cases where the highest scoring result is a false positive, we found that a correct result occurs in ranks 2 to 5 (results not shown). Third, the 90 hard targets are very hard, and they do not include any sequence that is correctly recognized by PDB-Blast. Fourth, many targets consist of multidomain proteins, which are harder to recognize with high scores when submitted as a single target. We prefer to take these stricter criteria and be clear about the low specificities of the servers. It is obvious that relaxing some of the above criteria (such as including more easier targets, partitioning targets into individual domains, etc.) would positively influence the reported specificities, but this would not be in the best interest of non-expert users when they are dealing with very hard targets.

The low specificity of the automated servers has already been observed in previous experiments (e.g., CAFASP1) and comes as no surprise; in fold recognition, increased sensitivity leads to a lower specificity. However, in many cases in which standard methods such as BLAST (Altschul et al. 1997) and its variations do not provide a clear result, a biologist is encouraged to try more risky methods and use his or her expertise to extract a useful result.

Table 2 shows also that almost all servers exhibit conditions under which they have higher specificity than the other servers. The results strengthen the assertion that all prediction servers provide additional value for the structure prediction experiments, and even though some might find the correct template less frequently, in all cases they provide valuable information. Nevertheless, high specificity is extremely important for large-scale structure prediction experiments. The main problem there is to identify those predictions that are most likely correct. From this point of view, among those methods with comparable sensitivity, one would prefer the methods with higher specificity.

False positives

Table 3 shows the five highest scoring false positive models (according to MaxSub) produced by each server. The scores assigned by the servers for these predictions are generally above the confidence level suggested by the authors. Thus, it is clear that the suggested confidence levels do not mean 100% error-free predictions. In some cases, the scores were surprising to the authors of this benchmark. Nevertheless, it is important to keep in mind that the table shows the models (target-template pairs) that, according to MaxSub, are considered false positives. These include a number of cases in which even if the correct fold were identified, the alignment had significant shifts, and thus MaxSub scored it as wrong. The table shows both of these models—those with poor alignments as well as those that correspond to real false

Table 3. *False predictions of the fold-recognition servers*

	Target Score	Template
FFAS		
1b9nB	41.8	1a13_
1c17M	14.8	1awk_
1cavB	12.8	1cfg_
1joyb	11.7	1b3gA
1d0vA	11.4	1dabA
3D-PSSM		
1cm2A	4.1E-02	1occR
1b22A	4.3E-02	3ezzA
1b91A	5.6E-02	1hmf_
1d4bA	5.9E-02	1ddf_
1ct7A	6.5E-02	1aoo_
GenTHREADER		
11cfJ	0.96	1ps2_
1d8bA	0.94	1j11_
1d6kA	0.94	1ptf_
1qsdB	0.92	2ccyA
1dioM	0.87	1nfn_
INBGU		
1gd5A	20.4	1bxw_
1t12A	20.1	1pex_
1dwnA	19.0	12bdB
1dbtA	18.1	1p11_
1d3dB	17.9	2hft_
PDB-Blast		
1b9nB	2E-32	1a13_
1cg1A	4E-14	1c1gC
1dabA	6E-08	1clgc
1dbgA	1E-07	3btaA
1d0vA	5E-06	1maeH
T98-lib		
1q1wB	106	1preA
1dwnA	61	8fabD
1ccwD	51	letu_
1gmmA	34	1nagB
1d2mA	21	3kar_

List of the five wrong predictions with highest reported score based on the evaluation with MaxSub. For all servers, the five highest false positives are above the significance level recommended by the authors.

positives. It also includes a number of cases in which the score was slightly below MaxSub thresholds. Thus, the table shows a strict, nonpermissive evaluation. Another important factor to consider is that MaxSub takes into account the full length of the target, even for multidomain targets. Thus, a number of the false positive predictions correspond to large multidomain targets that correctly identified a part of a domain, but because of the normalization used by MaxSub, they received a below-threshold score (however, this is not the case for the highest scoring false predictions). A more lenient evaluation would lower the number of false positives. Nevertheless, to be on the safe side, we prefer to use the stricter approach. This is important if one aims to know the reliability of the servers when building 3D models for targets in general—a very important question in large-scale, genomic fold predictions.

The highest score for a false positive prediction made by PDB-Blast was confirmed by running regular PSI-Blast at NCBI (in October 2000). In the fifth iteration, both PDB-Blast and PSI-Blast reported the similarity between two transcription factors with PDB codes 1b9n and 1al3 with an E value < 1e − 32. The functional similarity between both proteins is evident; however, none of the structure evaluation methods or classifications could report any significant similarity between both structures. Both the target structure and the template are shown in Figure 1 for comparison. It is clear from the figure that there is no obvious structural similarity between these two proteins. The models of the highest wrong predictions for other servers can be viewed on our web site (http://BioInfo.Pl/LiveBench).

Detailed analysis of the errors is a very interesting project, but one beyond the scope of this article.5

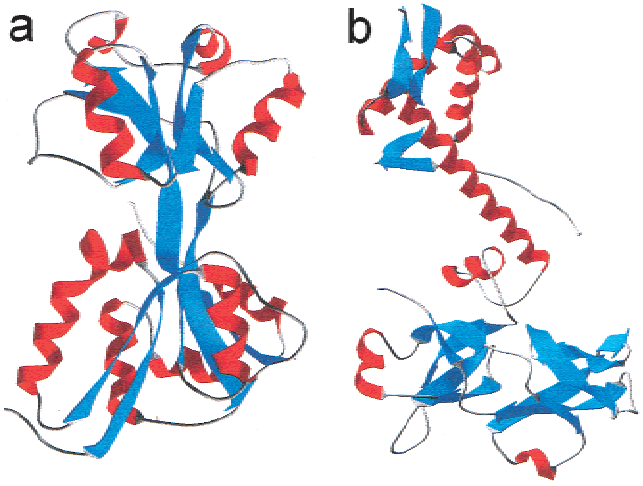

Fig. 1. Highest scoring wrong prediction of the PDB-Blast server. The template 1AL3 (*a*) and the target 1B9N (*b*) predicted by the PDB-Blast server to share the same fold (with an E value of 1e − 32 after four PSI-Blast iterations).

The knowledge gathered from analysis of the errors will be very helpful for further improvement of the servers, which can be best conducted by the developers themselves. We observed some trend among the highest false predictions of other servers: they involve small proteins with similar secondary structure pattern but different topology. Thus, an important goal for the LiveBench experiment should be to point out such errors. It has already been reported to us that in one of the servers, the presence of high-scoring false positives led to the discovery of a program bug, which, after being fixed, significantly raised the specificity of the server (results not shown).

Similarity between servers

To further test the thesis that all presented prediction servers are valuable, we have estimated the percentage of the predictive power of one server that can be reproduced by the others. We have calculated a value, called the prediction coverage, for each target. For each pair of servers, we have computed the number of times that one server performs better than the second. The prediction coverage corresponds to the percentage of the score obtained by the first server relative to the score obtained by the second server. If the first score is higher than the second (that is, the ratio is >1.0), then the score is truncated to 1.0 for this target. Table 4 shows the average prediction coverage between all servers obtained for hard targets only, and using both evaluation methods.

Table 4 shows clearly that no prediction server is able to completely reproduce the accuracy of any other. The highest coverage obtained is in the area of 87% for INBGU reproducing PDB-Blast models using alignment-independent evaluation. This means that if INBGU and PDB-Blast were combined perfectly, >10% of the performance of PDB-Blast could be added to the performance of INBGU. If looking at the values obtained using MaxSub as the evaluation method, this increase in performance could be much higher. For example, it would be very useful to combine FFAS and GenTHREADER, because their mutual coverage is very low (as evaluated by MaxSub).

The total number of correct predictions can be raised substantially if we combine servers in a hypothetical perfect manner. Using MaxSub evaluation, we found that out of the 90 hard targets, 34 correct models can be identified if all servers are ideally combined. This is substantially (47%) higher than the 18 correct models generated by the most sensitive server. A similar trend is observed when using LGscore evaluation: here, the increase is from 37 correct templates for the best of the servers to 59 for the combined prediction. This gives us an upper bound for estimating the sensitivity of a hypothetical jury system that would combine the first models generated by the associated servers to produce a final model. Approximately 60% of the folds could

Table 4. *Average prediction coverage between servers on hard targets*

	FFAS	3D-PSSM	GenTHR	INBGU	PDB-Blast	T98-lib
MaxSub						
FFAS	1	0.351	0.066	0.389	0.452	0.308
3D-PSSM	0.381	1	0.432	0.606	0.591	0.436
GenTHREADER	0.077	0.265	1	0.300	0.250	0.363
INBGU	0.494	0.698	0.556	1	0.446	0.561
PDB-Blast	0.144	0.176	0.082	0.118	1	0.083
T98-lib	0.271	0.329	0.465	0.367	0.239	1
LGscore (all-indep)						
FFAS	1	0.635	0.664	0.618	0.779	0.641
3D-PSSM	0.742	1	0.754	0.714	0.823	0.769
GenTHREADER	0.719	0.720	1	0.671	0.794	0.707
INBGU	0.808	0.801	0.786	1	0.874	0.817
PDB-Blast	0.487	0.435	0.490	0.418	1	0.476
T98-lib	0.736	0.667	0.711	0.676	0.786	1

The values show how well the servers listed in the left column can reproduce the predictive power of the other servers (those in columns 2–7). All values are calculated using hard targets only. For example, the first row shows how well the FFAS server can reproduce other servers using the MaxSub method as evaluation. The highest number in this row is 0.452, which means that only 45% of the value of the models predicted by PDB-Blast could be reproduced by FFAS. The table is of course not symmetric, and PDB-Blast could reproduce only 14% of the value of the FFAS models (using MaxSub as evaluation).

potentially be identified this way. This figure is significantly higher than the expected results for any prediction method alone reported for genome-wide structure prediction experiments (Fischer and Eisenberg 1997; Koonin et al. 1998; Jones 1999; Rychlewski et al. 1999), even though the genome-based evaluations take into account all genes (targets), not only the hard ones. Even this number could be increased if not only the first model reported by the server were taken into account. In many cases, the correct template can be found in the top 10 models generated by the servers (data not shown).

Discussion

The six servers evaluated in LiveBench (or their older versions) were also evaluated in CAFASP1 (Fischer et al. 1999). Although a direct comparison is not possible because the assessment methods are different, in general we found the results fairly consistent. Both PSI-BLAST and the threading servers can correctly identify the easy targets. The hard targets are undetectable by PSI-BLAST, but the servers can identify some of them, although with low specificities. However, because the assessment methods are so different and especially because the ranking in CAFASP-1 was probably not very significant as a result of the small number of targets considered, we find some differences in the overall ranking between the CAFASP-1 and LiveBench assessments. This emphasizes one of the most valuable features of LiveBench—the larger number of targets considered. It will be interesting to compare the results of the upcoming CAFASP-2 experiment and of future LiveBench assessments when more servers are included.

As an aside, we find it interesting to estimate the rate at which our structural knowledge is growing. In the period from October 29, 1999 to April 6, 2000, 1814 new PDB entries were deposited. Of these, only 125 sequences were unique, that is, they showed no significant sequence similarity to previous entries, as computed by Blast. This finding corresponds to a growth in unique sequences of 7%. PDB-Blast could identify a remote relative with a significant score for 30 of these 125 sequences. Of the remaining 95 entries, only 5 were genuine new folds (no structural similarity to existing proteins was found by DALI). Future analysis of the predictions of the servers for targets with no structural relatives may shed some light on the ability of the servers to predict novel folds. The current ability of the methods available to distinguish these targets is expected to be low, taking into account the sensitivities and specificities of the servers on hard targets. We found that the low number of new folds in half a year is surprising. We expect the ongoing structural genomics projects to deliver "fresh" folds with higher rates.

The LiveBench program offers various advantages over other benchmarking experiments:

(1) Through continuous scanning of the protein structure database PDB, the program identifies a large number of potential prediction targets for evaluating various prediction methods. This is an advantage over other prediction experiments such as CASP or CAFASP, which suffer from the relatively low number of prediction targets.

(2) A large number of prediction methods can be compared at the same time under identical conditions, which is one of the major advantages of this experiment. In many cases, reports compare the performance of a new method by using established benchmark test sets and

show a superior performance when compared with previous works. However, a significant portion of the improved performance is the result of the mere availability of more sequences and structures in the databases. This obviously results in an unfair comparison with the older reports. LiveBench provides an ideal scenario in which the comparison is equitable to all servers because this timing factor is removed.

(3) The computational costs of the tests are partially covered by the developers of the servers. Even the evaluation of results can be out-sourced to other servers (as were the DALI server, the LGscore server, and the MaxSub server in the current experiment).

(4) Not all methods must be publicly available as source code or executable programs, which offers the possibility for commercial companies to test their products against academic servers under equal and fair conditions.

(5) The program evaluates methods as seen from the perspective of a potential non-expert user. The user, or the evaluating team, does not need to worry about the choice of optimum parameters or the updating of fold libraries, which remains the responsibility of the developers.

(6) The program can promote standards in the design of servers and standards in the information transfer between them. These standards will make it simple to couple new prediction methods to the LiveBench program. A new prediction server will have preliminary performance analysis confirmed by an independent body in a matter of months with very little extra effort.

We are confident that the program will become a joint project of a large portion of the global community of research groups active in the protein structure prediction field. The progress of LiveBench can be followed every week on the LiveBench home page (http://BioInfo.PL/LiveBench). A similar project named EVA (http://maple.bioc.columbia.edu/eva/) was also launched recently. EVA currently offers an automated evaluation of secondary structure prediction servers.

Conclusion

We can draw a protocol for structure prediction for two general situations in which structure prediction is desired. High prediction specificity is preferred in cases in which the investigator has a set of many proteins and the main question is the choice of one target protein for further analysis. In cases of target identification, the specificity of the prediction is crucial. Because of its simplicity, the frequently updated template library and wide availability of PDB-Blast would probably make it the method of first choice. However, as seen in the analysis of specificities, it is not the safest prediction method to use. Furthermore, the analysis of false positives demonstrated that all servers might produce high-scoring wrong predictions. Thus, our recommendations in this case would be to use as many servers as possible to confirm the predictions.

Another equally frequent situation occurs when biologists focus their research on one protein. In this case, the preferred protocol practically demands the ue of all servers and the comparison of all the results because under some circumstances, any method out of the set may provide the correct answer.

Especially in cases in which the sequence-based methods do not report any significant predictions, the deployment of the structure-based (threading) servers may be important. The specificity of the predictions is expected to be low, but the gains are in the sensitivity. In such cases, the biologist should analyze the results and use his/her intuition or additional information (i.e., functional annotations) (Zhang et al. 1999; MacCallum et al. 2000) to try to distinguish the correct prediction from the wrong ones.

Despite the ranking produced by any evaluation criteria, in general we strongly discourage confining one's fold-recognition attempts to any particular prediction server. As demonstrated several times in this paper, all methods are able to produce unique predictions. To facilitate the use of various prediction servers, a meta server that automates the use of different methods and transforms the results in uniform formats is a very useful tool. An example of such an academic server can be accessed at http://BioInfo.PL/meta/.

Materials and methods

The LiveBench program is based on the following subroutines:

(1) The protein structure database PDB is updated every Wednesday. LiveBench downloads the sequences of newly released proteins.

(2) The sequences of new PDB proteins (targets) are compared with the previous set of PDB proteins (templates) using BLAST (Altschul et al. 1997). If there is a hit to any protein from the previous set (templates) with an E value <0.1 (computed using the size of the database of PDB sequences), then the target protein is skipped. This procedure removes all targets that are trivial based on BLAST.

(3) All remaining proteins are clustered with the nrdb90.pl program (Holm and Sander 1998b), using sequence identity of 90% as a cutoff criterion.

(4) The representatives of the new clusters are then submitted to all participating prediction servers by a meta server.

(5) The results from all servers are collected by the meta server and translated into uniform formats using tailored prediction server parsers.

(6) The target protein is also submitted to the DALI structure comparison server (Holm and Sander 1998a) to evaluate structural similarity of the target with other proteins in the PDB database. This helps to distinguish novel folds from proteins that have an appropriate template in the database and to estimate the relative difficulty of the target.

(7) Simple structural models are built using all results (the Cα atoms of aligned residues of the template proteins are used as model). The native structure is compared with the generated models by using two evaluation methods, MaxSub (Siew et al. 2000) and LGscore (Susana Cristobal Barragan, Adam Zemla, Daniel Fischer, Leszek Rychlewski, and Arne Elofsson, in prep.; see below).

Evaluation methods

We generate Cα models from the alignments returned by the server by cutting out the aligned Cα atoms from the template without connecting the insertions. The two methods used to evaluate the Cα models are MaxSub and LGscore.

(1) MaxSub identifies the largest subset of Cα atoms of a model that superimpose well over the experimental structure, using an alignment-dependent approach. MaxSub returns values between 0.0 and 1.0 where 1.0 indicates the identity of two structures. A value >0.0 indicates usually a nonrandom structural similarity. Thus, the cutoff of 0.0 was used to identify correct predictions (a model with MaxSub score >0.0 is labeled correct). The evaluation between the model and the correct stricture is strictly based on the alignment provided by the prediction server. Thus, using this method, it is the model that is evaluated rather than the similarity between the target and the selected template. MaxSub was tested on CASP3 targets, and results similar to those of the official CASP3 assessment were automatically obtained. In CAFASP-2, MaxSub will be used as the official evaluation method.

(2) LGscore returns values between 1.0 and 0.0 where 0.0 indicates the identity of two structures (opposite to MaxSub). The cutoff value of 0.01 was used for our analysis (a model with LGscore score <0.01 is labeled correct). In this paper, only alignment-independent LGscore evaluation results are reported (MaxSub was used for alignment-dependent evaluation, which resulted in conclusions that would have been identical to those arrived at if alignment-dependent LGscore had been used). The alignment-independent LGscore evaluates the structural similarity of the aligned parts of the target and the template. The detailed alignment of all residues provided by the prediction server is not taken into account. The program tries to find an optimum alignment based on structural similarity. However, because only the aligned parts of the template and the target are used for structural comparison, a badly wrong alignment to a correct template can still result in a negative assessment (wrong prediction).

More information about different model evaluation methods and our comparison of them, which led to our choice, can be obtained from Arne Elofsson (Susana Cristobal Barragan, Adam Zemla, Daniel Fischer, Leszek Rychlewski, and Arne Elofsson, in prep.). The main difference between both measures is that MaxSub focuses on the aligned model returned by the prediction server, whereas the alignment-independent LGscore is more tolerant to shifts in the generated alignment.

Acknowledgments

We thank Liisa Holm, Adam Godzik, David Jones, Kevin Karplus, Lawrence Kelley, Bob MacCallum, and Mike Sternberg for support, valuable discussions, and free access to the protein structure prediction and evaluation servers. We thank BioInfoBank for providing all necessary hardware for the meta server and the project at the IIMCB site.

The publication costs of this article were defrayed in part by payment of page charges. This article must therefore be hereby marked "advertisement" in accordance with 18 USC section 1734 solely to indicate this fact.

References

Abagyan, R.A. and Batalov, S. 1997. Do aligned sequences share the same fold? *J. Mol. Biol.* **273:** 355–368.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25:** 3389–3402.

Bernstein, F.C., Koetzle, T.F., Williams, G.J., Meyer, E.F., Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., and Tasumi, M. 1977. The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.* **112:** 535–542.

Brenner, S.E., Chothia, C., and Hubbard, T.J. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci.* **95:** 6073–6078.

Fischer, D. 2000. Hybrid fold recognition: Combining sequence derived properties with evolutionary information. *Pac. Symp. Biocomput.* 119–130.

Fischer, D. and Eisenberg, D. 1997. Assigning folds to the proteins encoded by the genome of Mycoplasma genitalium. *Proc. Natl. Acad. Sci.* **94:** 11929–11934.

Fischer, D., Barret, C., Bryson, K., Elofsson, A., Godzik, A., Jones, D., Karplus, K.J., Kelley, L.A., MacCallum, R.M., Pawlowski, K., et al. 1999. CAFASP-1: Critical assessment of fully automated structure prediction methods. *Proteins* (Suppl.) **3:** 209–217.

Holm, L. and Sander, C. 1998a. Dictionary of recurrent domains in protein structures. *Proteins* **33:** 88–96.

———. 1998b. Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics* **14:** 423–429.

Jones, D.T. 1999. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287:** 797–815.

Jones, D.T., Taylor, W.R., and Thornton, J.M. 1992. A new approach to protein fold recognition. *Nature* **358:** 86–89.

Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grate, L., and Hughey, R. 1999. Predicting protein structure using only sequence information. *Proteins* (Suppl.) **3:** 121–125.

Kelley, L.A., McCallum, C.M., and Sternberg, M.J.
2000. Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.* **299:** 501–522.

Koonin, E.V., Tatusov, R.L., and Galperin, M.Y. 1998. Beyond complete genomes: From sequence to structure and function. *Curr. Opin. Struct. Biol.* **8:** 355–363.

MacCallum, R.M., Kelley, L.A., and Sternberg, M.J. 2000. SAWTED: Structure assignment with text description—Enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. *Bioinformatics* **16:** 125–129.

Moult, J., Hubbard, T., Fidelis, K., and Pedersen, J.T. 1999. Critical assessment of methods of protein structure prediction (CASP): Round III. *Proteins* (Suppl.) **3:** 2–6.

Overington, J., Johnson, M.S., Sali, A., and Blundell, T.L. 1990. Tertiary structural constraints on protein evolutionary diversity: Templates, key residues, and structure prediction. *Proc. R. Soc. Lond. B Biol. Sci.* **241:** 132–145.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., and Chothia, C. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.* **284:** 1201–1210.

Rychlewski, L., Zhang, B., and Godzik, A. 1999. Functional insights from

structural predictions: Analysis of the *Escherichia coli* genome. *Protein Sci.* **8:** 614–624.

Rychlewski, L., Jaroszewski, L., Li, W., and Godzik, A. 2000. Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci.* **9:** 232–241.

Siew, N., Elofsson, A., Rychlewski, L., and Fischer, D. 2000. An automated measure to assess the quality of protein structure prediction. *Bioinformatics* (in press).

Taylor, W.R. 1997. Multiple sequence threading: An analysis of alignment quality and stability. *J. Mol. Biol.* **269:** 902–943.

Torda, A.E. 1997. Perspectives in protein-fold recognition. *Curr. Opin. Struct. Biol.* **7:** 200and ndash;205.

Zhang, B., Rychlewski, L., Pawlowski, K., Fetrow, J.S., Skolnick, J., and Godzik, A. 1999. From fold predictions to function predictions: Automation of functional site conservation analysis for functional genome predictions. Protein Sci. **8:** 1104–1115.