3D-SHOTGUN: A Novel, Cooperative, Fold-Recognition Meta-Predictor

Daniel Fischer*
Bioinformatics, Department of Computer Science, Ben Gurion University, Beer-Sheva, Israel

ABSTRACT **To gain a better understanding of the biological role of proteins encoded in genome sequences, knowledge of their three-dimensional (3D) structure and function is required. The computational assignment of folds is becoming an increasingly important complement to experimental structure determination. In particular, fold-recognition methods aim to predict approximate 3D models for proteins bearing no sequence similarity to any protein of known structure. However, fully automated structure-prediction methods can currently produce reliable models for only a fraction of these sequences. Using a number of semiautomated procedures, human expert predictors are often able to produce more and better predictions than automated methods. We describe a novel, fully automatic, fold-recognition meta-predictor, named 3D-SHOTGUN, which incorporates some of the strategies human predictors have successfully applied. This new method is reminiscent of the so-called cooperative algorithms of Computer Vision. The input to 3D-SHOTGUN are the top models predicted by a number of independent fold-recognition servers. The meta-predictor consists of three steps: (i) assembly of hybrid models, (ii) confidence assignment, and (iii) selection. We have applied 3D-SHOTGUN to an unbiased test set of 77 newly released protein structures sharing no sequence similarity to proteins previously released. Forty-six correct rank-1 predictions were obtained, 30 of which had scores higher than that of the first incorrect prediction—a significant improvement over the performance of all individual servers. Furthermore, the predicted hybrid models were, on average, more similar to their corresponding native structures than those produced by the individual servers. This opens the possibility of generating more accurate, full-atom homology models for proteins with no sequence similarity to proteins of known structure. These improvements represent a step forward toward the wider applicability of fully automated structure-prediction methods at genome scales. Proteins 2003;51:434–441.** © 2003 Wiley-Liss, Inc.

Key words: fully automated structure prediction; fold recognition; critical assessment; CASP; CAFASP; LiveBench; Meta-predictors; computer vision

INTRODUCTION

One of the challenges of the postgenomic era is to computationally assign three-dimensional (3D) structures to the proteins encoded in genome sequences.[1–3] As a result of the various genome sequencing and structural genomics projects, structure-prediction methods are playing an increasingly critical role in translating the information on the relatively small subset of proteins whose structures will be solved into accurate models for all proteins.[3,4] Fold-recognition (FR) methods (c.f. Refs. 5–10 for a review) use the structural information of solved proteins to model the structure of those proteins that share no significant sequence similarity to any of the proteins of known structure. Despite significant progress witnessed in the past years, automated methods still have a number of limitations.[11–13] First, FR methods are not able to model novel structures different from those already solved. Second, the models produced are often incomplete (i.e., only parts of the proteins are modeled) and can contain many gaps. Third, FR models often contain only a few correct secondary structure elements (SSE). Furthermore, although two different models of the same target may share a number of similarly modeled SSEs, other SSEs may be modeled completely differently. Thus, it is conceivable that in a given situation, some of the latter SSEs may be correct only in the first model, whereas others may be correct only in the second model. Thus, it may be unclear which of the two models should be selected or how to combine the two to produce a better model. Finally, it has been widely observed that one of the most critical aspects requiring improvement is the relatively low specificity of current methods.[12,13] That is, correct rank-1 predictions can receive scores lower than those obtained from incorrect ones, thereby preventing reliable automatic prediction. Thus, the fully automatic use of these methods at genomic scales is still limited because, at reasonably conservative thresholds, predictions can be obtained for only a fraction of the sequences.[1]

Grant sponsor: Israel's Defence Ministry; Grant sponsor: United States-Israel Binational Science Foundation (BSF), Jersusalem, Israel; Grant number: 9900032.

*Correspondence to: Daniel Fischer, Bioinformatics, Department of Computer Science, Ben Gurion University, Beer-Sheva, Israel. E-mail: dfischer@cs.bgu.ac.il

Received 27 August 2002; Accepted 18 November 2002

Despite these limitations of automated methods, structure-prediction experiments, such as CASP,[11,14] have shown that the combined use of human expertise and automated methods can often result in successful predictions.[2] This, however, requires extensive human intervention, because a human has to "improve" the model manually, has to determine whether the rank-1 model obtained is correct, whether there is a lower ranking model that corresponds to a correct prediction, or whether the results of the method indicate that no prediction at all can be obtained. To this end, human expert predictors have developed a number of semiautomated strategies.[11,14] One particularly successful strategy has been the application of a number of independent methods to extract a prediction from the top ranking predictions. This has proven useful because for some prediction targets, one method may succeed in producing a correct prediction while others fail, yet for other targets, this same method may fail while the others succeed. Because it is impossible to determine a priori for which targets a given method will succeed, human expert predictors attempt to extract any useful information from results obtained with different methods. Thus, the ability to incorporate some of this human expertise into a fully automated method represents a major challenge.[15]

The use of a number of models and methods to produce better predictions has already proven useful in a number of areas including artificial intelligence and computer vision as well as in protein structure prediction. It has been observed in protein secondary structure prediction (consensus of various methods[16]), in homology modeling (multiple-parent structures; e.g., Refs. 17–20), and in ab initio protein folding methods (clustering models and deriving recurring constraints from various models[21–23]). The idea of a "consensus" prediction has also been incorporated into FR methods. For example, fully automated FR servers (e.g., Refs. 5,8, and 9) that internally run a number of different (somewhat incompatible) methods to derive a consensus prediction have been developed. These methods ranked among the best performing servers in the latest CASP4 and CAFASP2 experiments.[24,25] Another particularly successful example of the derivation of consensus predictions was the semiautomated CASP4 participant named CAFASP-CONSENSUS,[25] which derived consensus predictions by analyzing the predictions of the FR servers that participated in the parallel CAFASP2 experiment.[25,26] The CAFASP-CONSENSUS performed better than any of the CAFASP servers[25] and ranked seventh among all other human predictors of CASP.[24] This finding showed that a fully automated meta-predictor, using the predictions of a number of independent servers, could be very powerful. Following CAFASP, a method named *pcons* was developed,[27] which fully automated the procedures used by the CAFASP-CONSENSUS.

In this article, we describe a novel, fully automatic, highly sensitive and specific meta-predictor method named 3D-SHOTGUN, which tries to overcome a number of the above-mentioned limitations of current FR methods and further emulates some of the successful strategies that human expert predictors often apply. The method presented here is reminiscent of the so-called cooperative algorithms of computer vision, in which local operations appear to cooperate to form global order.[28] Such algorithms are also reminiscent of several cooperative phenomena observed in physics, as well as in certain brain processes.

3D-SHOTGUN derives a prediction from information culled from the top predicted models of a number of independent FR methods. The goal of our method is to arrive at an "expert" decision, taking into account evidence from a number of different, unreliable agents.[29] 3D-SHOTGUN does not simply select a model from among the individual models, but it generates more complete and accurate hybrid models. 3D-SHOTGUN first collects the top n models of m different, previously developed FR methods, referred to as the "initial models." Subsequently, 3D-SHOTGUN applies the following steps: (i) assembly, (ii) confidence assignment, and (iii) selection (see Methods). In the assembly step, each of the initial models is used as a framework on which to build the new hybrid, assembled model, using the recurring structural similarities detected among the initial models. In the confidence assignment step, scores are assigned to each of the assembled models on the basis of the initial scores and of the structural similarities of the assembled models. In the selection step, 10 representative assembled models are chosen and reported in decreasing order, based on their assigned scores. To test the performance of the new method, we applied it to an unbiased set of 77 newly released protein structures which share no sequence similarity with proteins previously released.

METHODS

The Initial Models

Initial models are generated independently by any available method, serving as input to the 3D-SHOTGUN meta-predictor. In general, the top n hits of each of m different methods are used. $M_{i,j}$ denotes the model at rank j produced by method i. $s_{i,j}$ denotes the score assigned to $M_{i,j}$. Two versions of the 3D-SHOTGUN method have been developed. The first, named "SHGU," considers the top five models of each of the five components of the *inbgu* server.[9] SHGU is available to the community as an additional option of the *inbgu* server at http://www.cs.bgu.ac.il/~bioinbgu. The second version of 3D-SHOTGUN, named "3DS3," considers the top five models of three different, independent servers: ffas,[7] 3dpsm,[8] and the *inbgu* consensus[9] itself (i.e., not the new SHGU results). 3DS3 is available at the *bioinfo* meta-server[30] at http://bioinfo.pl/meta. The three servers chosen for 3DS3 were selected because they are well-tested, available servers and because of their proven performance in CASP4[24] and CAFASP2.[25] It should, however, be emphasized that 3D-SHOTGUN is independent of the choice of initial model-generating methods.

Assembly Step

This step is conceptually similar to the "cut and paste" procedure used in homology modeling when multiple

parents (templates) are available (e.g., Refs. 18–20,31, and 32). The main difference in homology modeling is that the parents are usually globally structurally similar (all are homologues to the target), whereas here, assembly can be carried out by using globally dissimilar FR models, sharing only partial substructures with one of the models. In our current implementation, the assembly is carried out by using a purely geometric, tallying procedure. In what follows we outline the steps of the procedure; a detailed description is included in the Appendix.

Each of the initial models is considered one at a time. For each model *M,* regions of structural similarity to each of the other initial models are first identified. Each initial model that shares some structural similarity with model *M* is superimposed upon *M* (using only the shared similarity to compute the transformations), thereby generating a multiple structural alignment. To build the assembled model, coordinates for each residue in the multiple alignment are selected by using a voting scheme (see Fig. 1 below). For each residue *i* and each model *N* containing *i,* the number of other models that include *i* in their superposition upon *N* are counted. The coordinates of residue *i* in the assembled model are copied from a model having the highest count. In the case where residue *i* is present in *M* and there is a tie, the coordinates are taken from *M.* This procedure thus builds hybrid, assembled models containing the most frequent structural features of the models in the multiple alignment, which often include coordinates for more residues than any of the initial models. The rationale of this strategy is that recurring structural features obtained by independent agents are more likely to be correct, because there are more ways to be wrong than there are to be right. Thus, a recurring feature is less likely to represent a recurrent error. It should be noted that the multiple structural alignment for each model *M* can contain structurally different models sharing only a few SSEs in common with *M* and that the resulting assembled model is a truly hybrid model, a composite of structural features from the various models in the multiple alignment (see the hypothetical example in Results for an illustration of the assembly process).

Confidence Assignment Step

In the current implementation, the following procedure is applied: each of the assembled models is assigned a score s', computed using (i) the original scores (normalized to similar scales) of the initial models (i.e., the $s_{i,j}$ values) and (ii) the structural similarity between them. More specifically, the score assigned to the assembled model $M_{i,j}$ is computed as $s'_{i,j} = \Sigma_{k,l} s_{k,l} \times sim(M_{i,j}, M_{k,l})$, where the indices k and l run over all models and $sim(M_{i,j}, M_{k,l})$ is the MaxSub similarity score[33] (see below) between the assembled models $M_{i,j}$ and $M_{k,l}$. The rationale of this step is to assign scores to the hybrid models that reflect the recurrent structural features detected among the various models.

Selection Step

For each cluster of highly similar assembled models only one representative model is reported. The 10 representative assembled models with the highest s' scores are reported as the output of the 3D-SHOTGUN method, sorted according to the decreasing order of their s' scores. The output includes full documentation of the assembly process and of the structural similarities among the various models.

Test Set

The new method's performance was evaluated by using a test set of 77 target proteins (listed in the legend of Table I), which correspond to LiveBench[12,13] targets released between May and October 2001. The LiveBench program provides a continuous evaluation of prediction servers, where on a weekly basis, new entries in the Protein Data Bank[34] with no sequence similarity to previously released proteins are selected as prediction targets. Thus, the selection of these targets ensures that the programs are tested on an unbiased set of newly released proteins.

Assessment Criteria

To detect structural similarities between models and to assess the correctness of the predictions, the MaxSub method[33] extensively used in the CAFASP and LiveBench experiments, was used. Given two models, or a model and a native structure, MaxSub searches for the largest subset of "well-predicted" residues (i.e., superimpose well; default: 3.5- Å, but this is a user-adjustable parameter) and produces a similarity score between 0.0 (random or no similarity) and 1.0 (perfect similarity).

Performance Measures

Three measures of performance were computed: sensitivity, overall quality of the models, and specificity. Only the highest scoring prediction was considered per target and per method. Hereafter, we refer to this highest scoring prediction as the "rank-1" prediction. These measures are similar to those used in the LiveBench experiment. Sensitivity counts the number of targets for which a correct (positive MaxSub score) rank-1 model was obtained. Overall quality is the sum of the MaxSub scores of each of the rank-1 models in the 77 test proteins. Specificity is measured as the number of correct rank-1 models identified with scores higher than the first to fifth false positives (see Ref. 13 for details).

RESULTS AND DISCUSSION

We first illustrate how 3D-SHOTGUN works by using a hypothetical target as an example. Subsequently, we apply the two versions of 3D-SHOTGUN, SHGU and 3DS3, to the set of 77 test sequences and compare their performance with that of previous methods.

A Hypothetical Example

We use a hypothetical protein to provide a detailed example of the application of 3D-SHOTGUN. The hypothetical protein is represented as a string of "structural features" (ABCDEFG), where each "feature" may indicate a secondary structure element or an individual residue (the actual assembly is carried out at the residue level).

Native Structure:	A	B	C	D	E	F	G	(unknown at assembly)
Initial model M_1:	A	b	C	d	E	F	-	(4 native-like features)
Initial model M_2:	A	B	C	d	-	f	g	(3 native-like features)
Initial model M_3:	A	b	c	D	E	-	g'	(3 native-like features)
Hybrid assembly for M_1:	A_1	B_2	c_3	D_3	E_3	F_1	g'_3	(5 native-like features)

Fig. 1. Illustration of the 3D-SHOTGUN assembly process. Structural features are indicated by letters (ABCDEFG). Native-like structural features of the models are represented in uppercase, and non-native-like features in lowercase. The dashes represent missing features (i.e., gaps that often exist in fold-recognition models). The assembly for model M_1 is illustrated. Only two of the other 24 initial models had some structural similarity with M_1: M_2 and M_3. The multiple structural alignment of the three models is depicted. Models M_2 and M_3 are superimposed upon M_1 by using only the shared similarities to M_1 to compute the transformations (M_2 shares features A, C, and d, whereas M_3 shares features A, b, and E). Using only the shared similarities to M_1 to compute the transformations is critical because the models may contain a number of incompatible features (e.g., b and B or g and g'; the apostrophe indicates that g and g' represent different, non-superimposable features) which, if taken into account, can produce a poor superposition. Notice that the three initial models had at most four native-like features, but taken together, six native-like features were predicted. An ideal hybrid model assembly would include these six features. The last line in the figure depicts the assembled model generated for M_1. The models from which the coordinates of the assembled model were taken are indicated as subindices. As can be seen, the assembled model contains coordinates for all the residues of the multiple alignment, resulting in a more complete model. For simplicity, in this example the multiple alignment has only three models. However, in general, the coordinates of the assembled models are extracted from any of the models in the multiple alignment (see Fig. 2). This assembled model is a hybrid containing the most frequent structural features appearing in the initial models. Five features correspond to native-like features (A, B, D, E, and F), whereas two correspond to non-native-like ones (c and g'). Notice that the voting process uses the submatrix V'_{M1}, which stores the results of the comparison of each model with the other 24 models (see the Appendix); thus, majority in the multiple alignment does not necessary imply that this feature will be selected. For example, the hybrid model chose the B feature of model M_2, even though the b features of M_1 and M_3 are a majority in the multiple alignment. This can occur if the counts for feature B of M_2 in V'_{M1} are larger (or equal) than the corresponding features of M_1 and M_3 (i.e., among all the initial models, B is more frequent than b). The opposite occurred for feature c. Because this assembled model inherited more correct regions from each of the initial models than incorrect ones, it is more similar to the native structure than the individual models are.

The assembly step (in the SHGU version) uses as initial models the top five hits of each of the five *inbgu* components and considers each of the 25 models one at a time. Here we describe the assembly of one of the 25 models, denoted as M_1 (see Fig. 1). The assembly step first searches for structural similarities between M_1 and each of the other 24 initial models. Here we illustrate the assembly process in a simple form: only two other models were found to have some structural similarities to M_1: M_2 and M_3. Notice that a full superposition of M_1 with either of these two models may have a large root-mean-square deviation (RMSD), because the models can be globally dissimilar, sharing only partial features. A multiple structural alignment based on M_1 is then generated (using only the shared similarities to compute the transformations). The assembled model is constructed by using the (transformed) coordinates of the different models in the multiple structural alignment (see Fig. 1). As described above, a voting scheme is applied on a residue basis to determine which model's coordinates to import into the assembled model. Thus, the assembled model is a hybrid composite of the recurring structural features of different models, including more residues than any of the initial models.

Figure 1 shows that the assembled model inherited correct regions from each of the initial models in the multiple structural alignment, as well as others that are incorrect. In this hypothetical, but realistic, example (see Fig. 2), 3D-SHOTGUN selected more correct than incorrect regions from the initial models. Consequently, the similarity of the assembled model to the native structure is higher than that of the initial models.

The assembly step produces assembled models for each of the initial models. In the confidence assignment step, each assembled model is assigned a score (see Methods), and the top scoring models (after clustering) are reported.

Large-Scale Evaluation

We evaluated the 3D-SHOTGUN method using the 77 protein test set. For each test protein, we applied both versions of 3D-SHOTGUN: SHGU and 3DS3. For SHGU, the top five models of the five *inbgu* components were collected directly from the stored results in the *inbgu* server, available at http://www.cs.bgu.ac.il/~bioinbgu. For 3DS3, the top five models of 3dpsm, ffas and *inbgu* were collected from the LiveBench web-pages at http://bioinfo.pl/LiveBench. We compare the performance of SHGU with that of the *inbgu*'s consensus. We also compare the performance of 3DS3 with that of SHGU. To measure the added value of 3DS3 over the individual servers, we compare its performance with that of the best individual server (referred to below as "best").

Tables I and II summarize and compare the performances of the various methods on the 77 targets of the test set. The supplementary table lists for each target, the

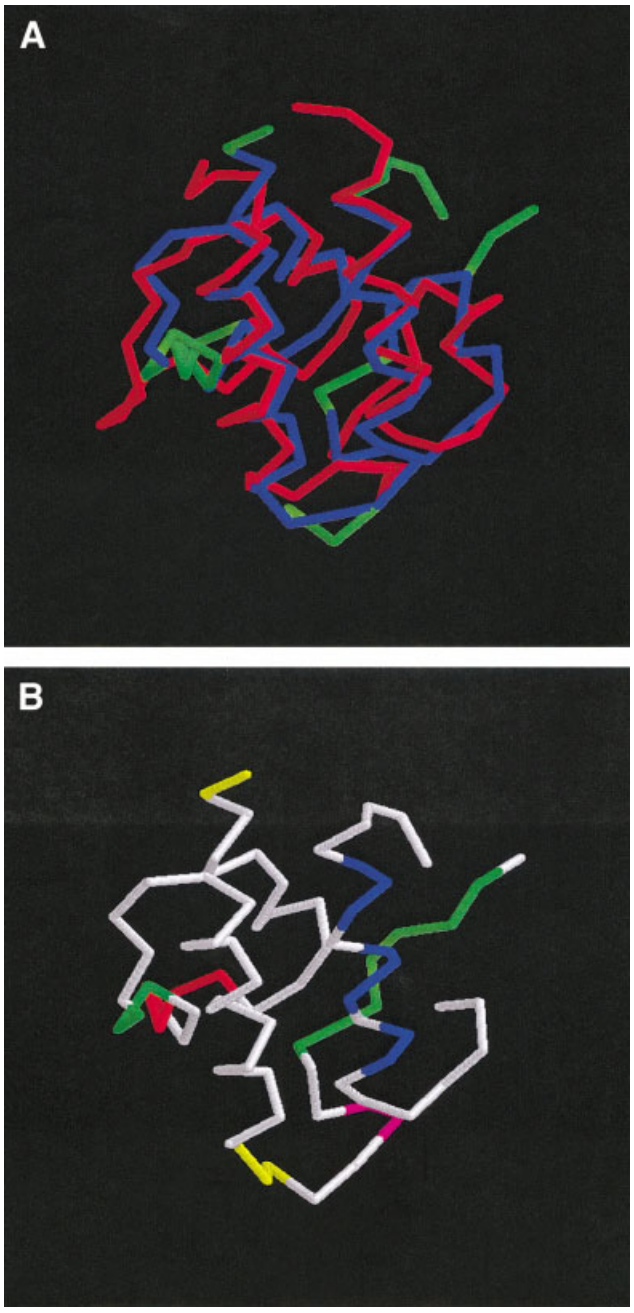

Fig. 2. 3D-SHOTGUN produces hybrid models by assembling regions from the initial models. Shown here is the predicted hybrid model for one of the targets in the 77 test set: PDB 1dv5. **a:** MaxSub superposition of the hybrid model (blue and yellow) with the native structure (red). Blue represents regions in the hybrid model identified by MaxSub as well predicted. **b:** The segments from the various initial models that compose the hybrid model. Most of the hybrid is composed from one single model: the rank-2 model from 3D-PSSM (shown in white).

MaxSub scores of the rank-1 predictions of each method. The first row of Table I shows the performance of the *inbgu* server alone. The *inbgu* server produced correct (MaxSub score > 0) models for 53% (41 of 77) of the targets, with an overall quality of 9.49. However, only 27 of the correct models had scores higher than the first false positive. Allowing for 3 errors, *inbgu* had 31 correct predictions; its average specificity was 30.2. The second row lists the "best" performance of the three individual servers: *inbgu,* ffas, and 3dpsm. The performances of ffas and 3dpsm (not shown) were very similar to that of *inbgu,* confirming the findings of previous experiments.[12,13,25] It has also been observed that each of the servers is able to produce correct predictions for targets where the other servers fail.[12] There were 51 targets for which at least one of these 3 servers had a correct rank-1 prediction. In 27 targets, the 3 servers had correct rank-1 predictions, in 11 targets only two servers had correct predictions, whereas in 13 targets, only one of the servers produced a correct rank-1 prediction.

The second part of Table I shows the performance of the new SHGU and 3DS3 methods. SHGU has a slight but significantly better sensitivity than *inbgu* (10% better; see Table II). A similar improvement was observed in SHGU's specificity. However, the major improvement is observed in the overall quality of the models produced (23% higher). This finding shows that the 3D-SHOTGUN method, when applied to the same data as that used by the *inbgu* method (i.e., the SHGU version), shows a better overall performance in all our three measures.

3DS3 produced 46 correct rank-1 models (60% of the targets) with an overall quality 15% higher and a specificity 17% higher than those of "best." 3DS3 predicted 30 correct models with scores higher than its first false positive and had an average specificity of 35.2. Of the 46 correct predictions, 27 (of 27) corresponded to those also correctly predicted by all three servers, 10 (of 11) corresponded to those correct in two of the three servers, and 8 (of 13) corresponded to those correct in only one of the servers. In addition, 3DS3 produced one correct rank-1 prediction for a target where none of the servers had a correct rank-1 prediction. This result shows the value of considering recurrent structural information from the top models of various methods.

Figure 2 depicts the rank-1 model produced by 3DS3s for one of the 77 test targets. This hybrid model was assembled from eight different initial models: 4 from 3dpsm, 2 from ffas, and 2 from inbgu. The initial models were obtained by the individual servers from five different parent structures, none having any significant sequence similarity to the target. Of the eight initial models, only five contributed their coordinates into the assembled model. The hybrid model includes coordinates for all 80 residues, more than any of the initial models. In addition, the quality of this hybrid model was higher than that of all the initial models. A full superposition of the model with the native structure results in an RMSD of 3.6 Å. MaxSub identified a subset of 64 residues that superimpose well (<3.5 Å) upon the native structure, resulting in an RMSD of 1.8 Å. The quality of this model as assessed by MaxSub is 0.63, a quality that corresponds to a relatively accurate prediction.

TABLE I. 3D-SHOTGUN Performance on the 77[a] Targets Test Set

Method	Sensitivity[b]	Overall[c] quality	Specificity[d] Average	No. correct with scores > first n false positives $n = 1$	2	3	4	5
Previous servers								
inbgu	41	9.49	30.2	27	28	31	32	32
best[e]	44	10.45	30.2	27	28	31	32	33
New meta-predictors								
SHGU	45	11.68	32.8	30	31	33	34	36
3DS3	46	12.08	35.2	30	32	35	38	41

[a]The 77 test proteins are: 1i1j 1hyn 1i9b 1i4t 1f08 1f0i 1i9z 1g8o 1i9s 1huf 1g8g 1ii7 1iho 1j77 1j98 1hf2 1im3 1i74 1f42 1ja5 1gd5 1i82 1e1a 1f47 1f45 1h9f 1j9y 1h9e 1j79 1hs6 1j8s 1f7t 1feu 1hq0 1h4u 1e44 1jfy 1ijy 1h5p 1htj 1jfi 1jc5 1ijx 1i52 1i5n 1h8m 1h6g 1jey 1h7y 1e6f 1ipg 1jeq 1f35 1hi9 1j7u 1e5x 1i8n 1jb3 1jb0 1jb0 1e5s 1ihr 1jb0 1eb7 1dv5 1jb0 1g7o 1gc5 1jbi 1i8t 1i9g 1jma 1jx3 1h6o 1j7v 1im4 1jmt.
[b]Sensitivity: number of correct (positive MaxSub score) rank-1 models (out of 77).
[c]Overall quality: sum of the MaxSub scores of each of the 77 rank-1 models.
[d]Specificity: computed as in the LiveBench experiment (see Methods). The average specificity is the mean of the five numbers shown in the following columns.
[e]Each of the performance numbers shown in "best" corresponds to the largest number from all three individual servers: inbgu, ffas, and 3dpsm.

TABLE II. 3D-SHOTGUN Improvement Over Previous Methods

Method 1	vs.	Method 2	Sensitivity (%)	Overall[a] quality (%)	Average specificity (%)
SHGU	vs.	inbgu	10%	23%	9%
3DS3	vs.	inbgu	12%	28%	17%
3DS3	vs.	SHGU	2%	3%	7%
3DS3	vs.	best	5%	15%	17%

[a]Legend as in Table II.

The cooperative character of 3D-SHOTGUN allows for extraction of recurring structural features from the initial models, features that are more likely to be correct than those appearing less frequently. Cooperation also allows for a better ranking of the alternative assembled models, because the evidence is gathered from different, reinforcing sources. Cooperation also allows for a higher specificity, which has been identified as one of the major limitations of current FR methods. This significant improvement in specificity may be due to the fact that (partially) correct predictions often recur among the initial models, but they either have below-confidence scores or do not appear as the rank-1 prediction. The cooperative approach of 3D-SHOTGUN exploits this fact and is able to assign more reliable scores.

One of the limitations of 3D-SHOTGUN is its dependency on the initial model-generating methods. If these err recurrently, such errors will inevitably be bequeathed to the generated models. To obtain a correct model, at least some correct structural features must appear in a number of the initial models, albeit not necessarily at rank-1. However, this limitation is also a positive feature—the better the individual methods become, the better the meta-prediction will be. As long as a meta-predictor performs better than the individual methods, the meta-predictor is fulfilling its goal. Another drawback of 3D-SHOTGUN is that the price paid for producing more complete models is that more can go wrong. Indeed, the assembled models can contain segments that either collide or are too far apart from the rest of the protein. For the most difficult targets, where little structural recurrence exists among the initial models, collisions and fragmentation represent undesirable features of the current implementation of 3D-SHOTGUN. However, such models usually receive relatively low-confidence scores, indicating their low reliability. Indeed, most of the higher scoring models suffer from little overlap or fragmentation (see Fig. 2). Improvements are required to correct or filter some of the wrong assemblies that are currently produced. A "refinement" step aimed at generating physically valid (and in a later stage, full-atom) models is being developed. Another aspect of ongoing work is the application of machine learning techniques to the confidence assignment step. This will allow for the assignment of different weights to the contribution of each model in the derivation of the scores.

3D-SHOTGUN can be applied to any set of methods. Indeed, a third (score-independent) version of the method, named 3DS5, extends 3DS3 to include any two additional servers. 3DS5, as well as SHGU and 3DS3, are being evaluated in the ongoing LiveBench-6 experiment and have already shown superior performance in the LiveBench-4 and CAFASP3 (http://www.cs.bgu.ac.il/~dfischer/

CAFASP3) experiments, where they have ranked at the very top.

CONCLUSIONS

We have presented a new, cooperative, FR method named 3D-SHOTGUN that produces a meta-prediction using as input the top models generated by a number of independent methods. 3D-SHOTGUN first assembles hybrid models from the initial models and then assigns scores to each of the assembled models by using the original models' scores and the structural similarities between them. These two features result in a higher sensitivity and a significantly higher specificity than those obtained by the individual servers. In addition, the average quality of the correct models is also significantly higher.

Despite the limitations of the current implementation of 3D-SHOTGUN described above, it has proved to be a very powerful approach, with much room for future enhancement. One of the most exciting challenges for bioinformatics is to be able to automate as many valuable human practices as possible. We believe 3D-SHOTGUN is a step forward in this direction. As our knowledge of protein structure expands, we expect that sensitive and specific methods, such as 3D-SHOTGUN, will become an important complement to experimental structural genomics projects.

ACKNOWLEDGMENTS

3DS3 and 3DS5 could not exist without the availability of the initial model-generating servers; our thanks to their developers. Thanks to Arne Elofsson, Leszek Rychlewski, Amir Mishali, and Eyal Shimony for discussions and to A.M. and E.S. for critically reading this manuscript. Special thanks to LR for the availability of his meta-server at http://bioinfo.pl/meta.

Parts of this research were funded by a grant from Israel's Defence Ministery and by grant No. 9900032 from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel.

REFERENCES

1. Fischer D, Eisenberg D. Assigning folds to the proteins encoded by the genome of *Mycoplasma genitalium.* Proc Natl Acad Sci USA 1997;94:11929–11934.
2. Abbott A. Computer modellers seek out ten most wanted proteins (news). Nature 2001;409:4.
3. Fischer D, Baker D, Moult J. We need both computer models and experiments (correspondence). Nature 2001;409:558.
4. Baker D, Sali A. Protein structure prediction and structural genomics. Science 2001;294:93–95.
5. Jones DT. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. J Mol Biol 1999;287: 797–815.
6. Shi J, Blundell TL, Mizuguchi K. Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J Mol Biol 2001;310: 243–257.
7. Rychlewski L, Jaroszewski L, Li W, Godzik A. Comparison of sequence profiles. strategies for structural predictions using sequence information. Protein Sci 2000;9:232–241.
8. Kelley LA, MacCallum RM, Sternberg MJE. Recognition of remote protein homologies using three-dimensional information to generate a position specific scoring matrix in the program 3D-PSSM. RECOMB, 1999, p 218–225.
9. Fischer D. Combining sequence derived properties with evolutionary information. Proc Pacific Symposium on Biocomputing, Jan 2000, p 119–130.
10. Fischer D, Eisenberg D. Predicting structures for genome sequences. Curr Opin Struct Biol 1999;9:208–211.
11. CASP4. Critical Assessment of Protein Structure Prediction Methods (CASP), Round IV. Proteins 2001;Suppl 5. See also http:// Prediction center.lnl.gov.
12. Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. Livebench-1: continuous benchmarking of protein structure prediction servers. Protein Sci 2001;10:352–361.
13. Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. LiveBench-2: Large-scale automated evaluation of protein structure prediction servers. Proteins 2001; Suppl 5:184–191. See also http://bioinfo.pl/ LiveBench.
14. CASP3. Critical Assessment of Protein Structure Prediction Methods (CASP), Round III. Proteins 1999;Suppl 4. See also http:// Prediction center.lnl.gov.
15. Siew N, Fischer D. Convergent evolution of CAFASP and computer chess tournaments: CASP, Kasparov and CAFASP. IBM Systems J 2001;40(2):410–425.
16. Cuff JA, Barton GJ. Evaluation and improvement of multiple sequence methods for protein secondary structure. Proteins 1999; 34:508–519.
17. Greer J. Comparative model-building of the mammalina serine proteases. J Mol Biol 1981;153:1027–1042.
18. Havel T, Snow M. A new method for building protein conformations from sequence alignments with homologues of known structures. J Mol Biol 1991;217:1–7.
19. Blundell TL, Sibanda BL, Sternberg MJ, Thornton JM. Knowledge-based prediction of protein structures and the design of novel molecules. Nature 1987;326:347.
20. Sali A, Blundell T. Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 1993;234:779–815.
21. Simons KT, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J Mol Biol 1997;268:209–225.
22. Huang ES, Samudrala R, Ponder JW. Ab initio fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions. J Mol Biol 1999;290:267–281.
23. Kihara D, Lu H, Kolinski A, Skolnick J. TOUCHSTONE: an ab initio protein structure prediction method that uses threading-based tertiary restraints. Proc Natl Acad Sci USA 2001;98:10125–10130.
24. Sippl MJ, Lackner P, Domingues FS, Prlic A, Malik R, Andreeva A, Wiederstein M. Assessment of the CASP4 fold recognition category. Proteins 2001; Suppl 5:55–67.
25. Fischer D, Elofsson A, Pazos F, Valencia A, Rost B, Ortiz AR, Dunbrack RL Jr. CAFASP2: the second critical assessment of fully automated structure prediction methods. Proteins 2001;Suppl 5:171–183. Special Issue: See also http://www.cs.bgu.ac.il/ ~dfischer/CAFASP2.
26. Fischer D, Elofsson A, Rychlewski L. The 2000 Olympic Games of protein structure prediction. Protein Eng 2000;13:667–670.
27. Lundstrom J, Rychlewski L, Bujnicki J, Elofsson A. Pcons: a neural-network-based consensus predictor that improves fold recognition. Protein Sci 2001;10:2354–2362.
28. Marr D. Vision. New York: W.H. Freeman and Company; 1982.
29. Johnson G Jr, Santos E Jr. Generalizing knowledge representation rules for acquiring and validating uncertain knowledge. In Proceedings of the Thirteenth Int. Florida Artificial Intellignece Research Society Conference, Orlando, Florida, May 2000.
30. Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. Structure prediction meta server. Bioinformatics 2001;17:750–751.
31. Levitt M. Accurate modeling of protein conformation by automatic segment matching. J Mol Biol 1992;226:507–533.
32. Fidelis K, Stern P, Bacon D, Moult J. Comparison of systematic search and database methods for constructing segments of protein structure. Protein Eng 1994;7:953–960.
33. Siew N, Elofsson A, Rychlewski L, Fischer D. MaxSub: an automated measure for the assessment of protein structure prediction quality. Bioinformatics 2000;16:776–785.
34. Bernstein FC, Koetzle TF, Williams GJB, Meyer EF, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. The Protein

Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol 1977;112:535–542.

APPENDIX

Here a detailed description of the assembly step is presented.

Input: The target sequence and t initial models.
Output: t assembled models.
Variable definitions:
m: the number of model-generating methods
n: the initial models considered per method
t: $m \times n$
l: length of the target sequence

The Algorithm

1. Generate a vote matrix V with t rows and l columns, by carrying out an all against all (MaxSub) comparison of the t initial models. $V_{i,j}$ tallies the number of models N in which residue j is part of the MaxSub superposition of model N onto model i.
2. For each model M, build an assembled model:
   2.1. Let V'_M be a matrix containing the subset of the rows of V, which correspond to those models sharing some minimal structural similarity with M (here we use a minimum of 20 residues in the MaxSub superposition).
   2.2. Superimpose each of the models included in V'_M onto M (using only the shared similarities to compute the transformations).
   2.3. For each residue i in the sequence do: Let k be the model that corresponds to the row with the largest value at column i of V'_M. If $k \neq M$ and $V_{k,i} = V_{M,i}$, then let $k = M$. Assign to residue i of the current assembled model the coordinates of the corresponding residue of model k (after the superposition carried out at step 2.2).