

A Neural Network Approach to Evaluate Fold Recognition Results

D. Juan,¹ O. Graña,¹ F. Pazos,¹ P. Fariselli,² R. Casadio,² and A. Valencia,^{1,*}

¹Protein Design Group, National Center for Biotechnology, CNB-CSIC, Campus Universidad Autónoma, Cantoblanco, Madrid, M-28049, Spain.

²CIRB Biocomputing Unit and Laboratory of Biophysics, Department of Biology, University of Bologna, Bologna 40126, Italy.

ABSTRACT Fold recognition techniques assist the exploration of protein structures, and web-based servers are part of the standard set of tools used in the analysis of biochemical problems. Despite their success, current methods are only able to predict the correct fold in a relatively small number of cases. We propose an approach that improves the selection of correct folds from among the results of two methods implemented as web servers (SAMT99 and 3DPSSM). Our approach is based on the training of a system of neural networks with models generated by the servers and a set of associated characteristics such as the quality of the sequence-structure alignment, distribution of sequence features (sequence-conserved positions and apolar residues), and compactness of the resulting models. Our results show that it is possible to detect adequate folds to model 80% of the sequences with a high level of confidence. The improvements achieved by taking into account sequence characteristics open the door to future improvements by directly including such factors in the step of model generation. This approach has been implemented as an automatic system LIBELLULA, available as a public web server at <http://www.pdg.cnb.uam.es/servers/libellula.html>. *Proteins* 2003;50:600–608. © 2003 Wiley-Liss, Inc.

Key words: protein structure prediction; threading; back-propagation; public web server; LIBELLULA

INTRODUCTION

The increasing amount of data generated by genome sequencing requires new methods of assigning function to the new sequences. Structural information is of primary importance for functional annotation of uncharacterized proteins. Efforts and investments in high throughput X-ray and NMR methods need to be paralleled by the development of additional computational tools for structural proteomics.

An efficient alternative to the systematic prediction of protein structures is homology modeling in which an experimentally determined protein structure is used to iteratively derive the structure of a sequence-similar target protein. However, this technique cannot be applied unless the template of the solved structure has a sequence similarity to the target of at least 30%.^{1–4} At present, only

a fraction of the known proteomes can be modeled by this procedure.⁵

The best alternative for the remaining proteins are threading methods.^{6–17} In this case, the first step is to recognize possible folds that can accommodate the query sequence. This process, known as “fold recognition,” can be applied by using techniques based on information derived from known protein structures¹⁶ or based on specialized sequence comparison methods with hidden Markov models.¹⁵ Once a suitable template is identified, the query sequence and the template sequence are aligned with each other, a step known as threading. The coordinates of the template protein are used to build the structural model of the query protein; the explicit model, based on the initial sequence-structure alignment, is optimized with alignment and three-dimensional (3D) model-building techniques. Although fold recognition and threading can be differentiated, in practice different programs tend to combine them in a single procedure.

Here we present an approach that improves the selection of correct folds from fold recognition results given by two methods. The first, SAMT99, uses an iterative hidden Markov model-based method for finding proteins similar to the target sequence^{15,18}; the second, 3DPSSM, is based on 1D and 3D sequence profiles coupled with secondary structure and solvation potential information.¹⁶ The two methods are representative of the two classes of sequence- and structure-based fold recognition methods described above. Both methods can process large volumes of data from genome sequencing and are available to the public as web servers. Furthermore, both servers were evaluated during the recent CAFASP2 experiment.¹⁹ Most of the targets (21 of 26) used in the evaluation were difficult to predict because they did not have any detectable sequence similarity with proteins of known structure. If correctly

Abbreviation: NN, neural network.

D. Juan and O. Graña contributed equally to this work.

F. Pazos's present address is ALMA Bioinformatica, Centro Empresarial Euronova, Ronda de Poniente, 4, 2nd floor, Unit H, Tres Cantos, Madrid, 28760, Spain.

*Correspondence to: A. Valencia, Protein Design Group, National Center for Biotechnology, CNB-CSIC, Campus Universidad Autónoma, Cantoblanco, Madrid, M-28049, Spain. E-mail: valencia@cnb.uam.es

Received 2 August 2002; Accepted 10 October 2002

predicted residues are defined as those placed <3.5 Å from their correct position in the known structure, the results indicate that 3DPSSM and SAMT99 have average predictions of 23% and 12%, respectively, of folds correctly recognized, scoring among the most powerful tools for the derivation of protein models. The CAFASP2 evaluation shows that there is still room for improvement.

We propose the use of additional information to improve the first step of fold recognition. We trained a back-propagation neural network (NN) system for each server, to distinguish between correct and incorrect models. NNs have been shown previously to be useful in the context of fold recognition, and a number of methods use NNs as part of their architecture.^{11,17,20} In our method, we trained a back-propagation NN for each server with nine features, including data on the sequence-structure alignment, shape of the initial explicit model, and distribution in the model of conserved and apolar residues. The score of the threading servers was also an input.

One clear difference with other methods that use NNs as part of their methodology is that our approach uses all this additional information and not only the alignments that were successfully combined with statistical potentials in GenTHREADER.¹⁷ To the difference of the so-called meta-servers, which use NNs to combine the output of various threading servers (i.e., Pcons²⁰), our method includes information additional to the servers results (which is not the case in Pcons) with the results of a single server (SAMT99 or 3DPSSM). By doing that, our method can be considered as a filter of the results of a server with the combination of information not explicitly taken into account by the initial threading methods. The information of the initial threading servers is only combined at the level of the output, and this combination is done according to the scores of the NNs system. For the contrary, the metaservers use the NN engine to combine the results of other servers directly.

Our results show that additional information filtered by NNs can significantly improve the ability of the threading methods to distinguish correct from incorrect models. We also found that basic information, such as compactness of the protein folds and distribution of conserved residues, is not used explicitly in current threading methods.

Our method, called LIBELLULA, is fully automatic, and it is available as a public web server at <http://www.pdg.cnb.uam.es/servers/libellula.html>. LIBELLULA has been inscribed in CASP5 (<http://predictioncenter.llnl.gov/casp5/Casp5.html>) and CAFASP3 (<http://www.cs.bgu.ac.il/~dfischer/CAFASP3/>) and also in the continuous evaluation system "EVA".²¹

MATERIALS AND METHODS

Definitions

In what follows, query proteins are the sequences submitted to the servers: Template folds refers to the folds returned by the servers for the constructions of the models. Implicit models are the translation of the query sequences into the corresponding template folds according to the alignments provided by the servers, also called sequence-

structure alignment. Finally, explicit models are the 3D models that would result from the modeling of the structures according to the sequence-structure alignment of the implicit models.

By fold recognition we characterize the first step of the process, in which the possible templates are proposed by the servers for a given query sequence, whereas threading is the second phase in which the sequence-structure alignments are constructed.

We use "Correct" and "Incorrect" to qualify the predicted folds on the basis of their comparison with the corresponding real structures, performed by computing the z-score parameter of the FSSP database.²²

We use the terms "Accurate" and "Inaccurate" to qualify the predictions given by the threading servers or by the corresponding NNs. This includes accurate and inaccurate predictions of correct and incorrect folds. In many respects, it is interesting to predict accurately correct and incorrect folds for a sequence.

The initial set of 796 protein chains used for training, testing, and validating each NN was obtained from the structurally nonredundant set of the TOPITS fold library,¹¹ and it is available at <http://www.pdg.cnb.uam.es/servers/libellulaTrainingSet.txt>. This set contains complete protein chains with lengths running from 21 to 905 amino acids, with a mean length of 207 amino acids.

From all the templates returned by the servers, we removed those that had $>25\%$ of identity with the query proteins (in the case of 3DPSSM, because of the high number of models sent by the server, we also rejected all the models resulting in a server score >1 , following the server indications). By this procedure we collected a set of 344 query proteins and 1199 templates from the 3DPSSM server and a set of 357 query proteins and 966 templates from SAMT99.

The NN performance was scored with a cross-validation procedure on the eight subsets obtained by randomly dividing each of the two data sets collected from the two servers.

Structure of the NN

The NN architecture comprises an input layer with nine neurons, a hidden layer with five neurons and two neurons as output units (Fig. 1). The topology and specifications were selected after testing the predictive power of several network configurations. We used the SNN package²³ to develop the networks (NN) applied to the servers.

Input information for NN

The nine input neurons were fed with the following features for each template, given a query sequence:

1. The score of the threading server (in terms of e-value for SAMT99 and 3DPSSM).
2. Closeness of conserved residues. Conserved residues are defined as those completely invariant in the corresponding HSSP alignments,³ available at "ftp://ftp.embl-heidelberg.de/pub/databases/protein_extras/hssp/".

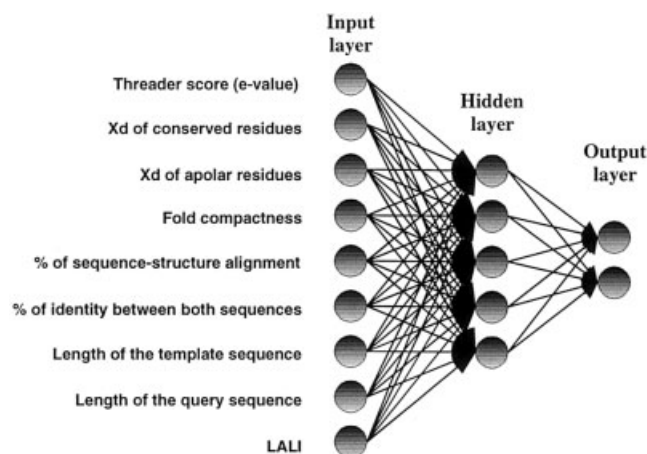


Fig. 1. Architecture of the NN. There are nine input neurons (the parameters for these inputs are written along with them), five hidden neurons in one hidden layer, and an output layer of two neurons that codified for “1–0” (correct model) or “0–1” (incorrect model).

We calculated the distribution of distances between all pairs of conserved residues and compared it to the distribution of distances for all other pairs of residues. The difference between these two distributions was calculated for every template resulting of the prediction of a server for each query. The comparison was based on the weighted harmonic average previously described as “Xd”.²⁴

$$Xd = \sum_{i=1}^{n^2} \frac{P_{ic} - P_{ia}}{d_i * n} \quad (1)$$

where n is the number of distance bins (there are 15 equally distributed bins from 0 to 60 Å). d_i is the upper limit for each bin (normalized to 60). P_{ic} is the percentage of conserved pairs with distance between d_i and d_{i-1} . P_{ia} is the same percentage for all pairs of positions. Defined in this way, $Xd = 0$ indicates no separation between the two distance populations, $Xd > 0$ indicates positive cases in which the population of conserved pairs is shifted to shorter distances with respect to the population of all pairs.

3. Closeness of apolar residues. A calculation similar to that described above was conducted for the apolar residues (Ala, Leu, Ile, Val, Met, Trp, and Phe). We demonstrated previously that conserved and apolar residues contain information about protein structures that is complementary to that contained in some threading methods.²⁵
4. Fold compactness is evaluated from the χ^2 test of the distribution of distances for the implicit models and the original template structures.

$$\chi^2 = \sum_i \frac{(N_i - n_i)^2}{n_i} \quad (2)$$

where N_i is the number of distances between residues within each bin (defined as in the Xd) for the template protein and n_i is the same for the query protein.

5. Length of the template sequence.
 6. Length of the query sequence.
 7. Length of the sequence-structure alignment (implicit model) defined as the number of aligned residues between the query sequence and the template provided by the server.
 8. Fraction of the query sequence built in the implicit model, expressed as percentage of the query sequence aligned to the template in the implicit model.
- These last four parameters account for the possible similarity in length between the query sequence and the correspondent template, as indicators of the sequence to structure fitting.
9. Percentage of identity between the query and template sequences. It has been reported that even weak similarities can reinforce the signals provided by threading methods.²⁶

All the values were presented to the NN as normalized real values.

NN Output Information

The two output units of the NN represent the goodness of the implicit models, encoded in the form “1-0” for “correct models” and “0-1” for incorrect models. For the training of the NNs, the classification as “correct” or “incorrect” was based on the structural similarity between the target models and the query proteins (known in the case of the training sets). The structural similarity was measured from the Z-score parameter of the FSSP database,²² which allows simple, direct calculation. The division between correct and incorrect models corresponds to the initial intention of improving the “fold recognition” step, and it is not concerned with the structural similarity between the implicit threading model and the query structure, which corresponds to the threading step.

We classified models as correct or incorrect by using three cutoff values of structural similarity: 6, 8, and 10 (z-score values) as computed by FSSP.

Assigning Confidence Levels to the Predictions

The output values of the NNs were used to assess the confidence of the predictions. This confidence is defined as the difference between the values of the two output neurons, one for the prediction of correct and the other for the prediction of incorrect. For instance, on the hypothetical case of a correct model, the desired output would be 1/0. If the result provided by the two output neurons were 0.8 and 0.2, respectively, the model would have been predicted as a correct one, and the confidence of the prediction would have been calculated as $\text{int}(\text{abs}(0.8 - 0.2) * 10) = 6$.

In this way, it is possible to obtain independent confidence values for each model given by a server. The output of the system is the sorted list of models by the confidence values. It is important to remark that the predictions of the NNs for each threading method are completely independent, but the final results are combined on the basis of the confidence values of the NNs.

Training and Testing Procedures

The protein sets were divided randomly into eight equal-sized cross-validation subsets, six for training, one for testing, and one for validating the network. Thus, the predictor was composed of eight subnetworks for each protein set. The final score was the average of the performance of all the tests of each subnet. In each cross-validation process, the best network was selected on the basis of its performance on the test set (training independent proof), and finally, the accuracy of the network was evaluated on the validation set (training and selection-independent proofs).

A typical training process consisted of 100,000 cycles and took 3 h in a cluster of 12 workstations under Linux SuSE 7.1.

Evaluation of the Results

Accuracy is computed as:

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{number of total predictions}} \quad (3)$$

NN and threading programs were evaluated at different threshold values of the threading programs and confidence values. The discrimination between correct and incorrect models is carried out at different threshold values of structural similarity (Z-score of FSSP).

At a given NN confidence value, the reference value for the prediction is computed as follows:

$$\text{Reference value} = \frac{\text{number of data of the most abundant class}}{\text{total number of data}} \quad (4)$$

where the most abundant class corresponds to correct/incorrect template folds (depending on which of them contains more models). The reference value is the minimum threshold for useful NN predictions: it computes the performance of a fixed predictor if all the templates belong to a single class (correct/incorrect). Scoring procedures were performed independently for each result obtained from 3DPSSM and SAMT99 servers.

RESULTS

We evaluate the method used for generating fold recognition models and the NN predictions conducted on these models. Model correctness is deduced from the similarity of the known structure of the query protein and the proposed structure by fold recognition methods for that protein. A real example of the type of results obtained is given in Table I, for the query sequence P49158:acetyl-coenzyme A carboxylase carboxyl transferase subunit beta. Table I shows how the list of models recovered by the servers was reordered according to the LIBELLULA evaluation: one set with the good models in first place, and another one with the bad models in second place. Each set ordered by the confidence of the evaluation (in both cases, it runs from 0, the less reliable, to 9, the most reliable).

TABLE I. Information Managed During the Evaluation of P49158 Query Sequence

Server	Fold returned	E-value of the server	NN evaluation	NN confidence
SAMT99	1ccwB	0.28	Good model	5
3DPSSM	1gwz	1.39	Good model	1
3DPSSM	1ldcA	2.16	Good model	0
3DPSSM	lsmaA	2.4	Good model	0
SAMT99	1jvnA	0.7	Bad model	8
SAMT99	1cb7B	0.28	Bad model	5
3DPSSM	1ad5A	5.23	Bad model	4
3DPSSM	1tplA	5.35	Bad model	4
3DPSSM	1ltdA	7.13	Bad model	4
3DPSSM	1ej9A	8.02	Bad model	4
3DPSSM	2src	8.62	Bad model	4
3DPSSM	1nseA	2.76	Bad model	3
3DPSSM	1b3oB	3.28	Bad model	3
3DPSSM	1eceA	3.38	Bad model	3
3DPSSM	1avaA	3.79	Bad model	3
3DPSSM	1bvza	4.69	Bad model	3
3DPSSM	1edqa	4.92	Bad model	3
3DPSSM	2aaa	5.77	Bad model	3
3DPSSM	1qhoA	6.1	Bad model	3
3DPSSM	1nodA	6.46	Bad model	3
3DPSSM	8gep	7.85	Bad model	3
3DPSSM	1a4yA	2.51	Bad model	2

Example of the results of LIBELLULA for the query sequence P49158. The first column shows the server that produced the fold evaluated, the second column contains the fold proposed by the server, the third one corresponds to the e-value assigned by the threading method, the fourth column shows the evaluation of the model done by the NN and the last one contains the confidence of the evaluation.

The confidence values indicate the validity of the structures for model building, rather than the quality of the models themselves.

The predictions of the servers are classified as accurate predictions of correct fold, if the FSSP Z-score obtained for the target sequence structure and the template pair is better than a given threshold, and inaccurate predictions of correct fold for all the other cases. The first section of the results analyzes these predictions.

The second section analyzes the predictions of the NN; in this case, the predictions are classified as correct or incorrect models depending on NN output values. In the case of the NN, both predictions are qualified by a confidence value that quantifies the tendency observed by the NN. The results are analyzed separately for predictions of correct and incorrect models at different levels of confidence.

The third section analyzes how the server predictions can be improved by further analyzing the fold templates with NNs. In this case, the results are analyzed separately for different e-values of the servers, because the possibility of improving the results depends critically on the values of the servers. In this section, the accuracy of the predictions is analyzed for predictions of both correct and incorrect templates.

For the following it is important to notice that we have two types of accurate NN predictions: accurate predictions of a template to be correct and accurate predictions of a

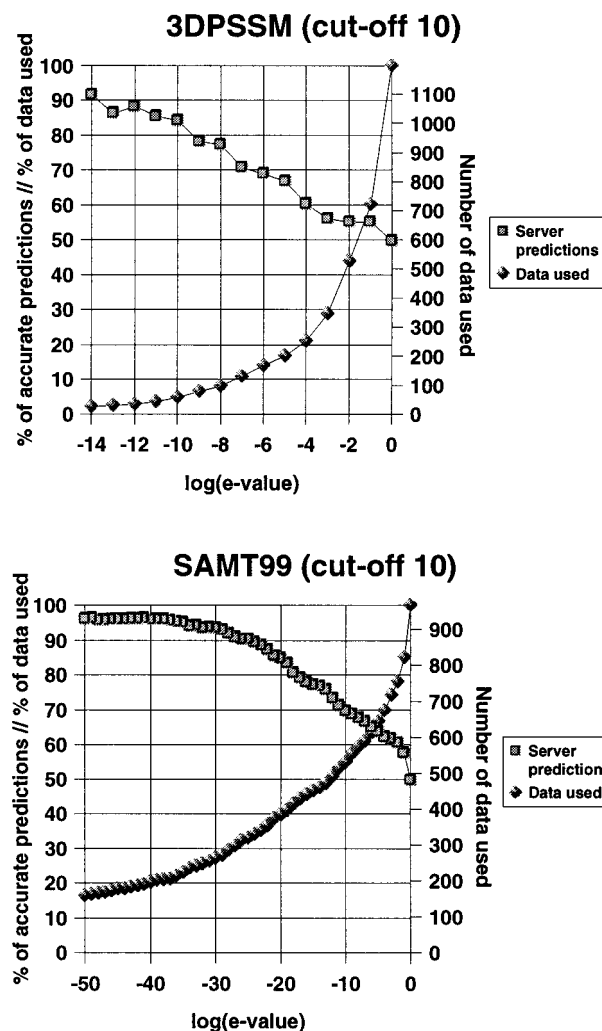


Fig. 2. Accuracy of the threading predictions. In these incremental plots an estimation of fold recognition predictions by 3DPSSM (top) and SAMT99 (bottom) are shown. In both cases, the percentage of accurate predictions was obtained as the ratio of the number of correct predictions to the number of total predictions multiplied by 100. In this case, we use the same FSSP Z-score cutoff as in the NN evaluations (≥ 10 corresponds to a correct template; all the rest are incorrect templates). The percentage of data used along the graphs corresponds to the number of templates with an e-value equal to or lower than the one appearing in the x axis.

template to be incorrect. Both cases are complementary and equally important to discriminate correct from incorrect templates.

Performance of the Servers on the Test Sets

The evaluation of the comparison was extracted from the FSSP database, where the similarities between protein structures are expressed in terms of Z-scores. To simplify the evaluation of the methods, folds are considered correct if they have a Z-score better than a given threshold (6, 8, or 10). The predictions of the servers are divided in accurate predictions of correct fold, if the FSSP Z-score obtained for the target sequence structure and the template pair is better than a given threshold and inaccurate ones for all the other cases.

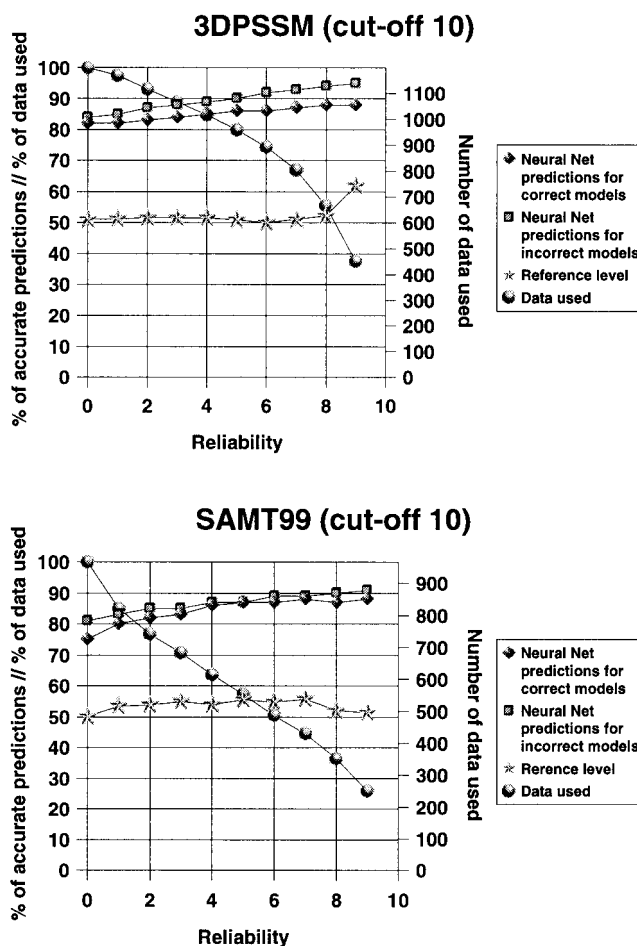


Fig. 3. Accuracy of the correct and incorrect predictions based on NNs at different levels of confidence. In these incremental graphs, percentages of NN accurate predictions for correct and incorrect folds, calculated as the ratio of the number of accurate predictions of a class (correct or incorrect templates) to the number of total cases of this class are shown versus the NN reliability estimated as the difference between desirable and obtained NN outputs for the validation set of proteins. Each graph corresponds to an independent experiment, providing an independent NN to classify templates coming from each threading program. The reference level for each server is also shown.

The performance of the two servers on the selected databases are shown by plotting the number of predicted models and accuracy of the prediction as a function of the e-values (Fig. 2). Only a small fraction of models are predicted with high scores (low e-values) with a high accuracy by both servers. Concomitantly, most of the templates are found in the region of worst scores (high e-values), where both servers are performing with a moderate level of accuracy (Fig. 2). Now the question arises: can we improve the prediction in the region characterized by high e-values? With this aim, we implemented an NN-based filter, which uses the server's scores and additional information to filter out the servers' outputs.

NN Filtering

Filtering of the models generated by the two fold recognition servers with NNs is shown by plotting the results at

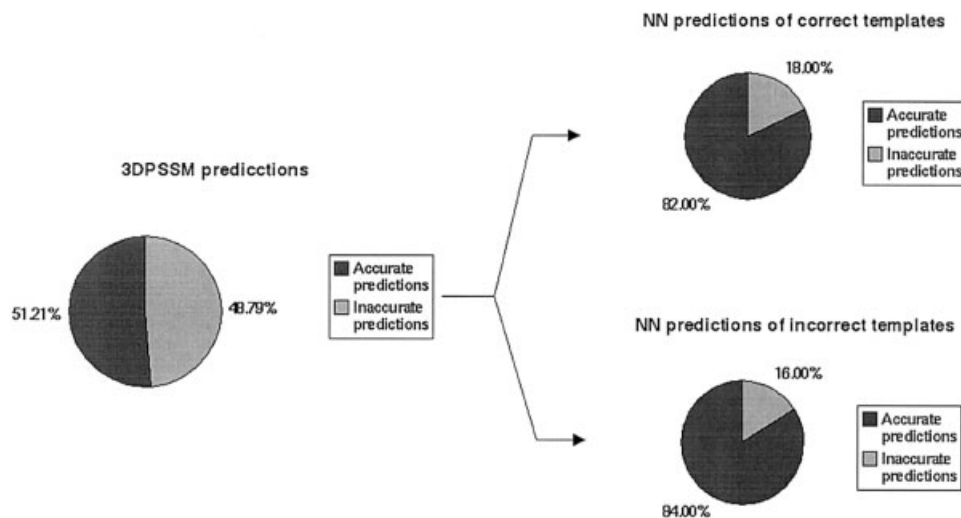


Fig. 4. Server (3DPSSM) and NN predictions evaluation. This figure illustrates the difference between the server (3DPSSM) and the NN: although the server gives a list of “correct” templates sorted by their accuracy estimation (left pie chart), the network can separate this information into two classes: correct and incorrect templates (right pie charts) based on a given FSSP Z-score threshold (10 in this case). The same cutoff for FSSP Z-score was used to estimate the predictions in the server case. “Accurate predictions” refers to those templates classified as correct for a real Z-score of FSSP equal to or higher than the cutoff value, and incorrect ones for the rest. This scheme was also applied for SAMT99 with similar results.

increasing values of confidence levels of the NN (Fig. 3). In this case, it is clear that high-confidence regions account for more accurate predictions (of correct and incorrect models) than low ones. In the case of SAMT99 at 0 value of confidence (all predictions included), LIBELLULA performs with an accuracy of 75% (accurate predictions for correct templates over total predictions) and 81% for incorrect templates (accurate predictions of incorrect templates). At a confidence value of 9, it scores 88% for correct and 91% for incorrect templates, indicating that at increasing reliability levels the discriminating capability of LIBELLULA improves. This is also the case for the NN filter of 3DPSSM, which increases accuracy from 82 to 88% and from 84 to 95% for correct and noncorrect templates with the increase in the confidence level.

The NN accuracy values are higher than the NN reference level of the initial predictions of the server, which shows the improvement provided by the method. It should also be noticed that the levels of prediction accuracy are similar for correct and incorrect models, an observation that confirms the consistency of the method (Fig. 4).

Analyzing the Combined Results of the NN and Threading Servers

In Figure 5, it is shown that when the servers have great confidence in the predictions (low e-values), the NN results are comparable with that of the original method. In this e-value range, however, the number of detected models is very low. In contrast, it is when the scores of the servers are relatively bad (high e-values) that the NN significantly improves the starting method performance. Actually, this is not only the most interesting region (e-value $> 1E-24$ for SAMT99 and e-value $> 1E-11$ for 3DPSSM) for the application of the NN, but it is also the most populated (65% of total templates for SAMT99 and 95% for 3DPSSM).

In Figure 5, it is clear that there is a correlation between the e-value of the servers and the level of their prediction accuracy. The levels of false positives of both servers shown here are probably lower than the ones obtained with large sets of real proteins (CAFASP type of evaluations), because these evaluations are performed by using known folds that the servers already have in their databases. Although we filtered all models with $>25\%$ of sequence identity, that is a postfiltering that may not avoid completely the internal use of this information by the servers during their calculations.

Evaluation of the Results at Different Levels of Structural Similarity

The results described in the section above were generated by considering as threshold for separating correct and incorrect values a cutoff of 10 for the Z-score of FSSP. The consequences of varying this threshold are analyzed in Figure 6. For both servers, NN accuracies are not very dependent on the level of structural similarity used. These results were expected because in each case NN is trained to recognize templates with a certain minimum of similarity to the target protein. This behavior shows that the NN training procedure is a robust method to discriminate templates with different levels of similarity to a target protein.

Increasing the threshold is equivalent to considering fewer folds as correct frameworks for the generation of models. In this case, it is obvious that the predictions are more difficult to make and require more precision of the methods. This requirement extends equally to the threading servers and the additional NN analysis.

In contrast, decreasing the threshold implies accepting more folds as correct frameworks. At the limit, if all the folds are considered correct, the number of accurate predic-

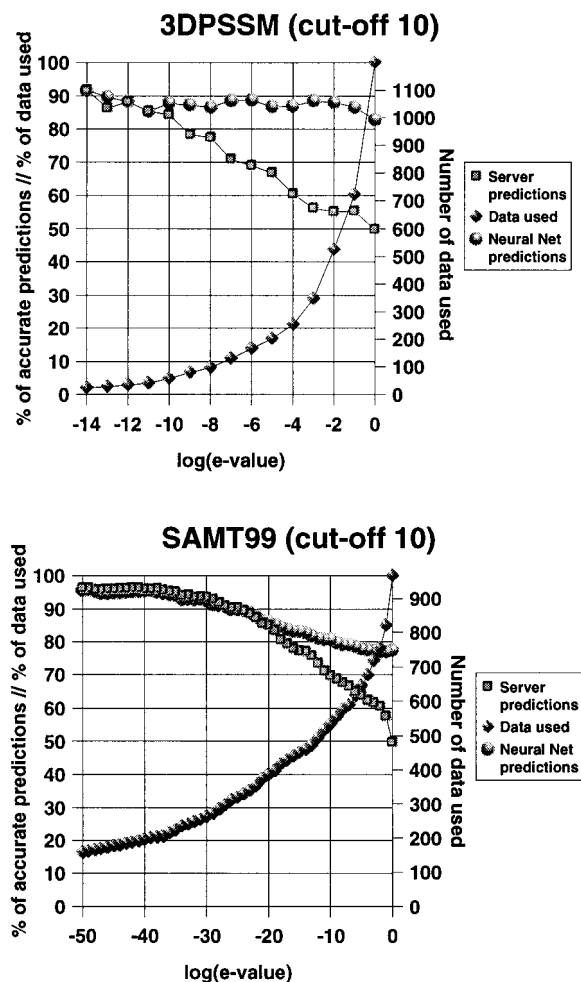


Fig. 5. Result of the application of the NN to the threading outputs analyzed at different scores of the threading programs. In this incremental plot, circles correspond to NN predictions (including correct and incorrect models), squares refer to server predictions, and diamonds correspond to the amount of data used. This graph is constructed in the same way as Figure 3.

tion of correct models will be equal to the threshold set for the servers (Fig. 6), and the predictions of the server coincides with that of NNs. In the light of these considerations, the threshold of 10 for the Z-score of FSSP used for the work described above seems to be a reasonable choice with a good balance of correct and incorrect folds (596 correct folds of 1199 total templates for 3DPSSM and 481 correct folds of 966 total templates for SAMT99).

Evaluation of the Sensitivity and Specificity of the NNs

The sensitivity (true positives/(true positives + false negatives)) and the specificity (true negatives/(true negatives + false positives)) have been represented under Receiver Operating Characteristic (ROC) curves (Fig. 7). They were calculated for the NNs under a cutoff of 10, according to the more restrictive Z-score of FSSP. Rates were obtained for correct and incorrect models and for the NNs of both servers. In every case, as appear in the ROC

curves, even with high sensitivity, the false-positive rate (1-specificity) remains low, showing again the consistency of the method.

DISCUSSION

The quick derivation of protein models is a key resource for Structural Proteomics, Pharmacology, Molecular Biology, and Biochemistry. During the last few years, a number of valuable resources have been created for the derivation of protein models on the web, in the form of threading servers. Still, there are many areas in which the predictions of these servers can be improved, especially, in the structure of the possible models and the integration of sequence information.

We have shown that sequence-derived information, such as completely conserved residues, apolar residues, and tree-determinant residues, is important for protein modeling²⁵ and protein-protein docking.²⁴

Here we used two servers, among the best in the recent CAFASP2 evaluation²⁰ and reevaluated the models with NNs trained to recognize correct and incorrect ones. The results show that the servers are highly accurate when the scores are good (low e-values), accounting for a small number of models (75% of accuracy for -50% of models in SAMT99 and -10% in 3DPSSM) and far less accurate when the scores are bad (high e-values). Unfortunately, in many practical applications of threading, most of the models do not have high e-values.²⁷⁻²⁹

It is in this region of low scores that filtering of the models with NNs becomes a valuable tool. Our results show that it is possible to discriminate between correct and incorrect folds by using NN filtering. These results suggest interesting applications. The first is the direct implementation of the NN as a postserver filter able to rescore the server results with additional information that can help to distinguish additional correct and incorrect folds. It may also be possible to implement the NN directly in the threading servers. Following the first approach, we implemented a system LIBELLULA (<http://www.pdg.cnb.uam.es/servers/libellula.html>), which is participating in the CAFASP3 and CASP5 experiments and also included in EVA,²¹ providing an open way to estimate its accuracy and utility.

Some limitations of the method can be found in cases in which the answer depends on a parameter not included in the input of our system; therefore, it will be difficult to perform an acceptable evaluation. Another drawback is that LIBELLULA is based entirely on the models proposed by SAMT99 and 3DPSSM and it cannot provide any evaluation in the case that those servers do not return any valuable models.

As a future development, it would be interesting to analyze the different contribution of each one of the input information, and subsets of them, to the overall system performances. This analysis could be interesting for future improvements of this and other fold recognition methods.

Furthermore, given the adaptability of NNs, it may be possible to improve the results by including additional input information such as the following: (i) tree-determi-

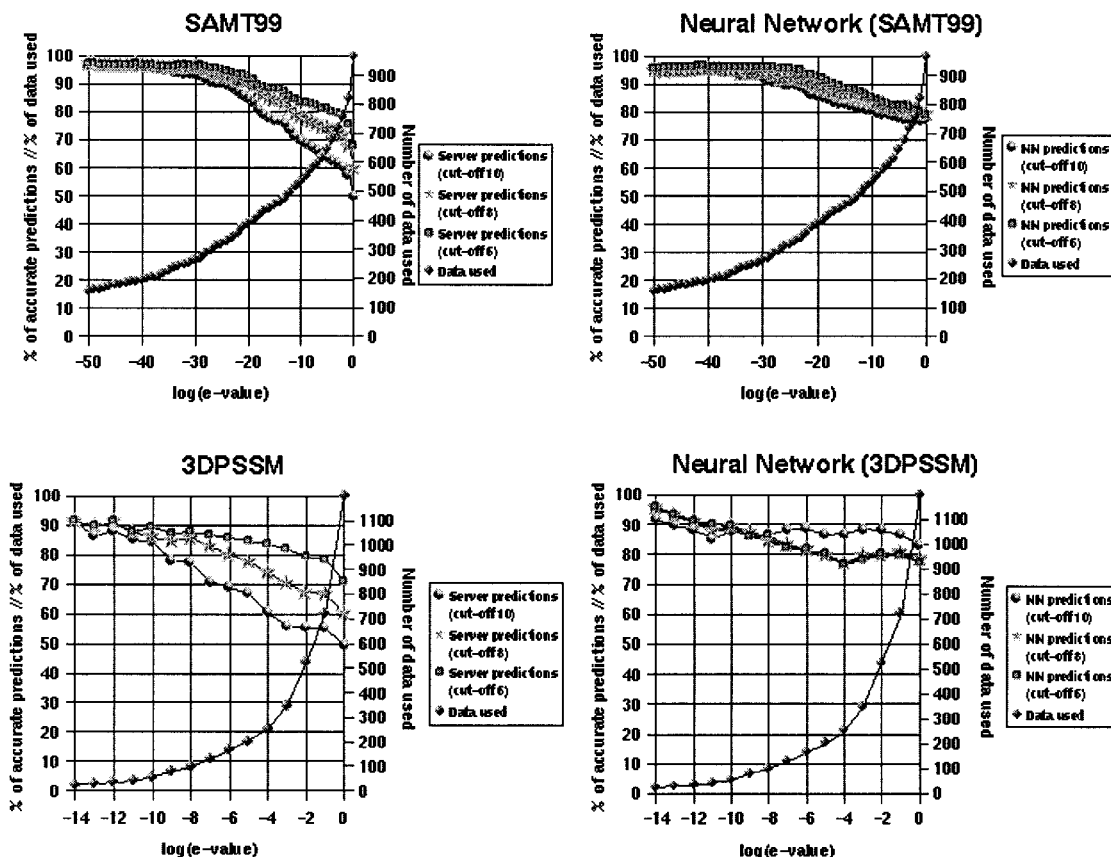


Fig. 6. Analysis of the threading and NN results at different levels of structural similarity between the known structure of the query and target proteins. These graphs illustrate the behavior of threading programs and their respective NNs predictions at different cutoffs of Z-score of FSSP.

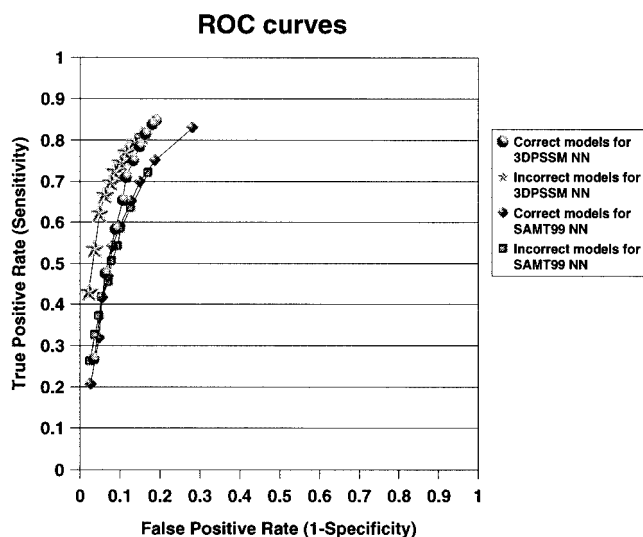


Fig. 7. Evaluation of the sensitivity and specificity of the NNs. ROC curves for a cutoff of FSSP Z-score of 10, for correct and incorrect models and for the NNs of both servers. Each point in the curves represents the value for a reliability of the NNs, from 9 to 0.

nant residues^{30,31} and correlated mutations,^{32–34} (ii) the structural quality of the models (not only a Z-score threshold), and (iii) details of the alignments to supplement the results provided by the evaluation of the folds. This, together with the addition of new servers and the combination of their predictions, will improve the results of this approach.

ACKNOWLEDGMENTS

We thank Dr. Burkhard Rost (Columbia University) for useful discussion about the training of the system. We are in debt with Lawrence Kelley and Mike Sternberg for the facilities using the 3DPSSM server, and also with Kevin Karplus for the facilities using the SAMT99 server. We acknowledge continuous support and interesting discussions with the Protein Design Group at CNB-CSIC. This work was supported in part by grants from the Spanish Ministry of Science and Technology. R.C. and P.F. acknowledge a grant from the Ministero della Università e della Ricerca Scientifica e Tecnologica (MURST) for the project “Hydrolases from Thermophiles: Structure, Function and Homologous and Heterologous Expression” and of a grant for a target project in Biotechnology of the Italian Centro Nazionale delle Ricerche (CNR), both delivered to R.C. The project was also initially supported by a grant for joint collaboration between Italy and Spain.

REFERENCES

1. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. *EMBO J* 1986;5:823–826.
2. Doolittle RF. Of URFs and ORFs: a primer on how to analyze derived amino acid sequences. Mill Valley, CA: University Science Books; 1986.
3. Sander C, Schneider R. Database of homology-derived structures and the structural meaning of sequence alignment. *Proteins* 1991;9:56–68.
4. Rost B. Twilight zone of protein sequence alignment. *Protein Eng* 1999;Feb 12:85–94.
5. Iliopoulos I, Tsoka S, Andrade MA, Janssen P, Audit B, Tramontano A, Valencia A, Leroy C, Sander C, Ouzounis CA. Genome sequences and great expectations. *Genome Biol* 2000;2: interactions0001.1–0001.3.
6. Jones DT, Taylor WR, Thornton JM. A new approach to protein fold recognition. *Nature* 1992;358:86–89.
7. Bryant SH, Altschul SF. Statics of sequence-structure threading. *Curr Opin Struct Biol* 1995;5:236–244.
8. Sippl MJ. Knowledge-based potentials for proteins. *Curr Opin Struct Biol* 1995;5:229–235.
9. Bowie JU, Luethy R, Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 1991;253:164–170.
10. Casari G, Sippl MJ. Structure-derived hydrophobic potential. Hydrophobic potential derived from X-ray structures of globular proteins is able to identify native folds. *J Mol Biol* 1992;224:725–732.
11. Rost B. TOPITS: threading one-dimensional predictions into three-dimensional structures. In: Rawlings CD, Altman R, Hunter L, Lengauer T, S. Wodak, editors. Third International Conference on Intelligent Systems for Molecular Biology. Cambridge, England: Menlo Park, CA: AAAI Press; 1995:314–321.
12. Fisher D, Eisenberg D. Protein fold recognition using sequence-derived predictions. *Protein Sci* 1996;5:947–955.
13. Fischer D. Hybrid fold recognition: combining sequence-derived properties with evolutionary information. Pacific Symposium Biocomputing. Altman RB, Dunker AK, Hunter Lauderdale K, Klein TE, editors. Hawaii, USA. Hawaii: World Scientific; 2000: 119–130.
14. Ouzounis C, Sander C, Scharf M, Schneider R. Prediction of protein structure by evaluation of sequence-structure fitness: aligning sequences to contact profiles derived from three-dimensional structures. *J Mol Biol* 1993;232:805–825.
15. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 1998;14:846–856.
16. Kelley LA, MacCallum RM, Sternberg MJ. Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J Mol Biol* 2000;299:499–520.
17. Jones DT. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol* 1999;287: 797–815.
18. Hughey R, Karplus K, Krogh A. SAM: sequence alignment and modeling software system, version 3. Technical report UCSC-CRL-99-11, 1999. University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064. Available from <http://www.cse.ucsc.edu/research/compbio/sam.html>.
19. Fischer D, Elofsson A, Rychlewski L, Pazos F, Valencia A, Rost B, Ortiz AR, Dunbrack RL Jr. CAFASP2: the second critical assessment of fully automated structure prediction methods. *Proteins* 2001;45:171–183.
20. Lundstrom J, Rychlewski L, Bujnicki J, Elofsson A. Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci* 2001;11:2354–2362.
21. Eyrich VA, Marti-Renom MA, Przybylski D, Madhusudhan MS, Fiser A, Pazos F, Valencia A, Sali A, Rost B. EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics* 2001;12:1242–1243.
22. Holm L, Sander C. Searching protein structure databases has come of age. *Proteins* 1994;19:165–173.
23. Zell A, Mache N, Sommer T, Korb T. The SNNS neural network simulator. *GWAI-91,15. Fachtagung Für Künstliche Intelligenz* 15.-20. Bonn, Informatik-Fachberichte 285, Springer; 1991:254–263.
24. Pazos F, Helmer-Citterich M, Ausiello G, Valencia A. Correlated mutations contain information about protein-protein interaction. *J Mol Biol* 1997;272:1–13.
25. Olmea O, Rost B, Valencia A. Effective use of sequence correlation and conservation in fold recognition. *J Mol Biol* 1999;293:1221–1239.
26. Domingues FS, Koppensteiner WA, Jaritz M, Prlc A, Weichenberger C, Wiederstein M, Floeckner H, Lackner P, Sippl MJ. Sustained performance of knowledge-based potentials in fold recognition. *Proteins* 1999;37:112–120.
27. Pons T, Chinea G, Olmea O, Beldarrain A, Roca H, Padron G, Valencia A. Structural model of Dex protein from *Penicillium minioluteum* and its implications in the mechanism of catalysis. *Proteins* 1998;31:345–454.
28. Pons T, Olmea O, Chinea G, Beldarrain A, Marquez G, Acosta N, Rodríguez I, Valencia A. Structural model for family 32 of glycosyl-hydrolase enzymes. *Proteins* 1998;33:383–395.
29. Devos D, Garmendia J, de Lorenzo V, Valencia A. Deciphering the action of aromatic effectors on the prokaryotic enhancer binding protein XylR through a fold recognition approach of its N-terminal domain. *Environ Microbiology* 2002;24:29–41.
30. Casari G, Sander C, Valencia A. A method to predict functional residues in proteins. *Nat Struct Biol* 1995;2:171–178.
31. del Sol A, Pazos F, Valencia A. Automatic methods for predicting functionally important residues. *J Mol Biol* Submitted for publication.
32. Göbel U, Sander C, Schneider R, Valencia A. Correlated mutations and residue contacts in proteins. *Proteins* 1994;18:309–317.
33. Pazos F, Olmea O, Valencia A. A graphical interface for correlated mutations and other structure prediction methods. *Comput Appl Biosci* 1997;13:319–321.
34. Olmea O, Valencia A. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold Des* 1997;2:S25–S32.