

Improved fold recognition by using the Pcons consensus approach.

Huisheng Fang , Björn Wallner , Jesper Lundström , Christer von Wowern
and
Arne Elofsson *

July 12, 2001

§ To whom correspondence should be addressed

Fax: +46-8-15 8057

Tel: +46-8-16 1553

Email: arne@sbcsu.se

Running Title:

Consensus fold recognition.

Abbreviations:

Keywords: fold recognition, threading, benchmark, web-servers, livebench, CASP, CAFASP.

*Stockholm Bioinformatics Center, Stockholm University, SE-106 91 Stockholm, Sweden E-mail: arne@sbcsu.se

Abstract

In the CASP and CAFASP processes it has been shown that manual experts are better to predict the fold of an unknown protein than fully automated methods. The best manual predictions seems to be performed by authors using a wide-range of different methods, and the most obvious similarity between them is that they have worked on fold recognition for years. Several of these experts also develop methods, however these methods do not perform as well as the experts them self. What are these secrets that the manual experts possess, but are not able to put into a computer?

Here we show that one such secret is the use of a “consensus” approach in fold recognition. By using several different methods, the same method with different parameters or searching using several homologous sequences a “consensus” prediction can be made. The consensus analysis can also be done using only a single sequence and a single method, by searching for similar hits among the top-scoring hits. In contrast most automatic methods do only use a single sequence, a single set of parameters and do not use the top-scoring hits to search for “consensus” predictions.

Here, we will describe a new method for fold recognition, Pcons, that utilizes the “consensus analysis” to improve automatic fold recognition. We will describe the process behind the development of Pcons. Starting with the use of a “semi-automatic” method in CASP4, and the later development of the fully automated Pcons method. We will show some results from large scale benchmarking that shows the advantages Pcons. Finally we will describe some recent development that has improved the performance of Pcons further.

1 Introduction.

As the genome projects proceed, we are presented with an exponentially increasing number of protein sequences, but with only a very limited knowledge of their structure or function. Since the experimental determination of both, the structure or the function is a nontrivial task, the quickest way to gain some understanding of these proteins and their genes is by relating them to proteins or genes with known properties.

From other examples in this book it is clear that there are many different fold recognition methods. Different methods are based on single sequences (Smith & Waterman, 1981; Needleman & Wunsch, 1970), multiple sequence alignments or profiles (Gribskov *et al.*, 1987; Altschul *et al.*, 1997; Karplus *et al.*, 1998; Rychlewski *et al.*, 2000), and predicted (Fischer & Eisenberg, 1996; Rost *et al.*, 1997; Rice & Eisenberg, 1997; Kelley *et al.*, 2000) or experimentally determined (Jones *et al.*, 1992) structures. It is not clear what features are most important and

how they should best be combined. However, what is noticed is that it seems to be of great importance with detailed choices of parameters to get the best performance.

Several different methods to benchmark the performance of these methods has been developed, including large scale benchmarks (Abagyan & Batalov, 1997; Park *et al.*, 1997; Park *et al.*, 1998; Lindahl & Elofsson, 2000), blind-predictions (Moult *et al.*, 1997; Fischer *et al.*, 1999; Fischer *et al.*, 2001) and automatic benchmarking of all newly solved protein structures (Bujnicki *et al.*, 2001*a*; Bujnicki *et al.*, 2001*b*). Several groups have also benchmarked the alignment quality for different fold recognition methods (Domingues *et al.*, 2000; Bujnicki *et al.*, 2001*a*; Sauder *et al.*, 2000; Elofsson, 2000).

The large scale benchmarked were based on databases of structurally related proteins, such as SCOP (Murzin *et al.*, 1995) and CATH (Orengo *et al.*, 1997). In these benchmarks it was concluded that the use of fraction identity as a measure of similarity between two proteins should be abandoned (Abagyan & Batalov, 1997). It is much better to use statistical measures, such as E- and P-values used in BLAST (Altschul *et al.*, 1997) and FASTA (Pearson & Lipman, 1988). In these studies the performance of several different methods was compared and methods that use multiple sequence information perform better than methods using only single sequence information (Park *et al.*, 1998; Elofsson, 2000). Different studies gave slightly different results on how to utilize the multiple sequence information best. However, it was clear that PSI-BLAST (Altschul *et al.*, 1997) performed very well and was one of the computational most efficient ways to use multiple sequence information.

It was also observed that different methods perform best at different levels of relationship (Elofsson, 2000). For proteins that only share the same fold, i.e. does not have a common ancestor, it is better to use methods that completely ignore the sequence, while for protein that are more closely related it is better to utilize the sequence information.

In later studies other fold recognition methods were also included in large scale benchmarks. Here it was observed that several fold recognition methods perform better than PSI-BLAST, when detecting distantly related proteins (Elofsson, 2000; Bujnicki *et al.*, 2001*a*; Bujnicki *et al.*, 2001*b*). Several, but not all, of these methods use structural information. The structural information can be included in several different ways. Inbgu (Fischer, 2000) and 3D-PSSM (Kelley *et al.*, 2000) use predicted secondary structures, 3D-PSSM and fugue (Shi *et al.*, 2001) use structural alignments and genTHREADER uses a threading potential. In addition, at least two methods that only use sequence information, FFAS (Rychlewski *et al.*, 2000) and Sam-T99 (Karplus *et al.*, 1998), seems to perform better than PSI-BLAST. It was also noted that no method today can reliably distinguish between weak correct hits and wrong hits (Bujnicki *et al.*, 2001*a*).

In the LiveBench studies it was observed that a correct often is obtained from a one method (Bu-

jnicki *et al.*, 2001a). Further, some studies have also focused on the quality of the generated alignment (Domingues *et al.*, 2000; Bujnicki *et al.*, 2001a; Sauder *et al.*, 2000; Elofsson, 2000). An important conclusion from these studies is that for different targets the best predictions are often made by different methods. It is quite common for a single method and pairs of distantly related proteins, that the optimum choice of alignment parameters differs from case to case. Thus, when evaluating the accuracy of a structure prediction protocol on a large set, it is quite clear that its performance could be increased if different, best suited approaches could be applied in appropriate cases.

The exact choice of parameters such as gap-penalties is of great importance for the performance of a method. Therefore, the development of better prediction methods is an art that only a few groups master.

2 Why are manual predictions better ?

How can these observations be used for making automatic fold recognition prediction as good as predictions by experts ? To answer these questions we need to try to understand what knowledge that experts use and that is not used by the servers (developed by them and others). We believe that there are three important contributions by the experts. These are biological knowledge, structural verifications and consensus analysis. Our recent studies indicate that of these three factors the last one is of most importance and that incorporating it into an automated method provides a significant improvement.

2.1 Biological knowledge

If a protein is known to be a DNA-binding protein, any high scoring hit to a DNA-binding domain would get the attention of an expert. Due to the current limitations in computer-readable classifications of protein functions this knowledge is hard to automatize. However, some attempts have been done. The SAWTED algorithm (MacCallum *et al.*, 2000) searches for related keywords in SWISS-PROT (Bairoch & Apweiler, 1996) between two proteins. SAWTED is utilized by 3D-PSSM and was shown to increase its performance by a few percentage (MacCallum, personal communication). It is difficult to judge how much biological knowledge actually improves fold recognition, but certainly in some cases it can be important.

2.2 Structural analysis.

As well as biological knowledge can be important, manual experts can also use structural knowledge. For instance it is known that secondary structure prediction algorithms are better

at predicting secondary structures than fold recognition methods. Therefore, if a model clearly have four helices a hit to an immunoglobulin can easily be disregarded.

Structural information can be used either directly in the alignment algorithm, as in Inbgu, or as a post-processing filter, as in genTHREADER (Jones, 1999). The difference is that when used in the alignment algorithm a different alignment is obtained, but when using as a post-processing method only the score of a particular alignment is influenced by the structural information. Two different types of information has successfully been used in fold recognition methods, predicted secondary structures and residue contact information.

Intuitively it seems as if a post-processing filter might be most useful to deselect false positives. This also seems to be correct as genTHREADER, the only method that use a post-processing filter, has a very good specificity (Bujnicki *et al.*, 2001*b*). The inclusion of structural information in the alignment procedure might instead improve the alignment of distantly related proteins.

2.3 Consensus analysis

A common trick used by fold recognition experts is to use what could be referred to as a consensus analysis. Here, not only one prediction for each target is considered. Instead models from different predictions, with similar scores, using different parameters, from different methods or for homologous sequences, are taken into account. In contrary an automatic fold recognition method returns a list of hits and when the performance is measured only the single highest scoring hit is used. Until the introduction of Pcons we are not aware of any method that utilize this type of information.

What is an expert actually doing when he examines several hits and why could it be used to increase the performance? One obvious feature is that using several parameters increases the possibility to create at least one good model. A method can create several predictions for each target-template pair and then use the one with the highest (normalized) score. Some groups have already used a simple form of a consensus predictors, where several models are created for each sequence-template pair. The Inbgu method performs five alignments using combinations of single sequence and profile data (Fischer, 2000). The 3D-PSSM method performs 3 alignments for each sequence-template pair (Kelley *et al.*, 2000). In both Inbgu and 3D-PSSM all alignments are made using predicted secondary structure information for the query sequence and the experimentally determined secondary structure of the template protein. The alignments of Inbgu are made using either single sequence or multiple sequence information of the query and the template. In 3D-PSSM two alignments are made using the query sequence and two different template profiles, one being derived from a superfamily-wide structural alignment, the third

alignment uses the template sequence and a profile obtained from the query sequence. For each query-template pair these methods choose one alignment, 3D-PSSM chooses the highest scoring one, while Inbgu takes also the rank into account. However, these method still only consider the different templates individually, while manual experts often examine multiple hits. Therefore, if hits 2 to 9 all are Tim-barrels but the first hit is something else the manual expert would guess Tim-barrel structure while these automatic method would not.

Here we present describe the process that lead to the development of a consensus method, that tries to mimic the work of an expert, Pcons. We show recent results from the LiveBench process that benchmarks the performance of Pcons and other fold recognition methods (Bujnicki *et al.*, 2001*a*; Bujnicki *et al.*, 2001*b*). Finally we will describe some current attempts to increase the performance of Pcons.

3 Consensus predictions in CASP4.

During the CASP4 process it was realized that the meta-server, described in another chapter, gave all participants in CASP4 the possibility to easily use a large set of fold recognition methods easily. Several of the top-performing groups in CASP4 utilized the results from the meta-server. However, they also used manual knowledge and other methods. In contrast, together with Daniel Fisher and Leszek Rychlewski, we wanted to examine if an automatic consensus prediction would perform better than the single servers and possible as good as the manual predictors.

During CAFASP we were not able to create a fully automated consensus server, therefore we used a semi-automatic procedure to perform the consensus predictions. These were submitted to CASP as the CAFASP-CONSENSUS semi-automatic predictions. Since the end of CAFASP we have developed the fully automatic consensus predictor Pcons. The process of both the automatic and manual consensus predictions can be described in three steps:

- First predictions are obtained from a set of web-servers using the meta-server (Bujnicki *et al.*, 2001*a*).
- The next step of the consensus procedure is to detect structural similarities between high scoring predictions from the different servers. In the manual procedure the structural comparison is done by listing the SCOP folds for each prediction. In the automated version this is done by comparing the structure of the produced models.
- The second step is to identify related predictions. In the manual predictions this was simply done by listing the number of predictions for a particular SCOP fold. In the automated predictions, the number of other models that were similar to a particular model is listed.
- In the third and last step the goal is to select one model out of all the models from the

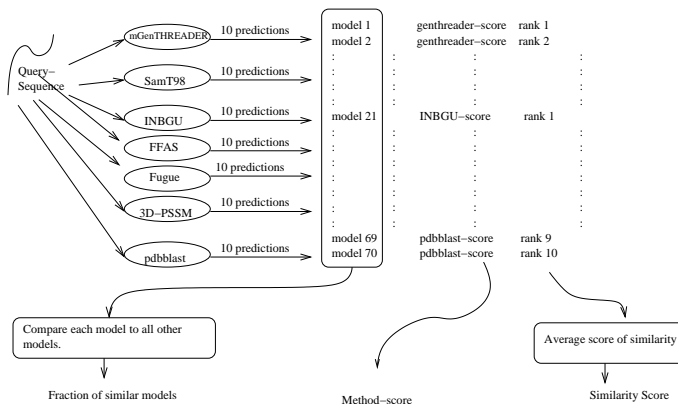


Figure 1: Description of the NN-all neural networks used in this study. In the top figure, the generation of the inputs to NN-all is shown. First up to 70 models are collected from 7 different web servers. The structure of these models are compared to the structure of all other models and templates. Three data points are fed into the network, the score, the fraction similar models and the fraction similar templates. A separate network is trained for each server. For each model obtained from one server the log of LGscore2 is predicted.

servers. In the automated consensus predictor this is done by a neural networks that combines the particular score of a model with the information about the number of other similar models. For the manual consensus predictions we used the same information as an input to our human neural networks.

In CASP4 we detected three scenarios: The first scenario consists of trivial predictions, where most methods predict the same fold with significant scores. In this cases we picked one of these predictions more or less randomly. The second scenario was when no servers gave any significant hits, but that a particular SCOP fold was clearly most frequent among the top hits. In this case we selected one prediction from this fold. In the last scenario there were no fold selected significantly more frequently than others. Here, we tried to use additional information. It was noted that in the first two scenarios almost always the most frequent fold was almost always the correct one.

According to the official ranking the CAFASP-CONSENSUS predictor performed better than all other automatic methods, and only six manual groups managed to perform better, although all of the consensus data and our predictions were publicly available. Post-predicting the CASP4 targets showed that the automatic Pcons predictor performed equally well as the best individual methods, but not significantly better.

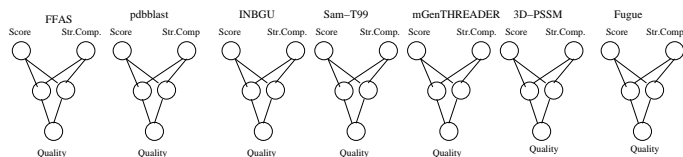


Figure 2: Description of the neural networks used by Pcons. The structure of these models, and the related templates are compared to the structure of all other models and templates. Two type of data points are given to the network, the score and the fraction of similar models. A separate network is trained for each server. For each model obtained from one server the log of LGscore2 is predicted by a first layer neural network.

4 Pcons

The Pcons approach presented here differs significantly from earlier consensus predictors. It follows the approach described above for the semi automatic CAFASP-CONSENSUS predictor used in CASP4. Pcons follows the following steps, see figure 1 and 2.

First a set of publicly available protein fold recognition web servers is utilized to produce input data. This data is converted into possible models for the query target.

Secondly all collected models are being compared used structural superposition and evaluation algorithms.

Thirdly similarity between models plus the particular score for this model is used to predict the quality of the model.

Following the protocol sketched above there are several possible choices that an automated consensus predictor can be implemented. In the following sections we will discuss how some of these choices affects the performance of Pcons.

4.1 Collection of publicly available models.

The current version of Pcons utilizes 7 servers (pdbblast, FFAS, Inbgu, mGenTHREADER, SAM-T99, 3D-PSSM and fugue). The top 10 hits from each server were converted into models by the meta-server. To create the optimal consensus predictor it is necessary to use the optimal number of models from the best servers. We have used the servers that performed best in LiveBench (Bujnicki *et al.*, 2001b) and some limited experimentation. The conversion uses the simplest possible method, i.e. using the backbone coordinates from the template. The resulting model might include gaps and if there are insertion there will be some residues missing. In a future we plan to replace this process by using a homology modelling method to build a full

model of the query protein for each predicted model. This will allow the use of additional terms that can be used as an input to the network.

4.2 Structural comparison.

The main idea about the consensus predictor is to use the observation that the correct fold often does not occur alone among the top hits. For instance if many of the top hits are immunoglobulins it is quite likely that the query protein actually is an immunoglobulin. But, what is the best method to examine how many immunoglobulins are found among the top hits ? One possible method could be to use one of the databases over protein folds, such as FSSP, SCOP or CATH. However, this involves several problems. These databases all use different domain definitions, while the different servers could use any of these or another set of domain definitions. Another problem is that these protein fold databases are updated less frequently than some of the servers. To avoid these problems we use structural comparisons of the models in Pcons.

Structural comparisons can be done in two different ways. Either the structure of the template or the structure of the models can be used. Structural comparisons of the models can be done either by taking the alignment into account (alignment dependent) or ignoring this (alignment independent), while the template comparison needs to use alignment independent structural superpositions. In our first studies we used both template and model comparisons (Lundström *et al.*, 2001) in an alignment independent way. However, the structural alignment method is too time-consuming for a server versions. Therefore, all versions of Pcons discussed in this study only utilizes alignment dependent model comparisons. In our experience the difference in performance between the different structural superposition methods is not very significant.

Besides the decision to use alignment dependent structural superpositions it is also possible to choose a measure for the similarity between two models. One simple measure of similarity could be rmsd. However it is possible that the models are only partly correct and then other measures are better. We choose to use the same type of measures as used for the evaluation of the model quality, see below.

4.3 Prediction of quality of the models.

Pcons predicts the quality and accuracy of all collected models, and if several servers predict one particular fold, Pcons will assign a high score to it. Pcons also differs from most earlier methods in the way how correct predictions are defined. Pcons is trained to predict the quality of a model while most other methods are optimized to detect if a correct fold is recognized. This might be advantageous as it is not trivial to uniquely define folds (Hadley & Jones, 1999), and even if the correct fold is found, the alignment could potentially be wrong. The problems

with domain definitions and updating of fold recognition methods, described above, makes it very difficult to train Pcons to predict the correct fold. Therefore Pcons is trained to predict the quality of the model.

Several different measures have been developed to measure the quality of a protein model. These can be divided into four groups. Global, such as rmsd, that uses the whole model to measure the quality. Alignment dependent that uses the most similar segment between the model and the correct structure and the model. Alignment independent measures are identical to alignment dependent measures but allow a shift in the alignment. Finally there are template based measures. We have recently reviewed these measures and concluded that “alignment dependent” measures were best at identifying the best models, while “alignment independent measures” are better at detecting fold recognition (Cristobal *et al.*, 2001).

In the first Pcons version we used the alignment independent measure LGscore2 to evaluate model accuracy (Cristobal *et al.*, 2001). In the more current version of Pcons we have used an updated version of the alignment dependent LGscore. The updated version provides better statistics for short segments. Since the first Pcons was released we have used several different measures of the quality, including MaxSub (Siew *et al.*, 2000), new versions of LGscore and touch (Bujnicki *et al.*, 2001*b*). Here we observed that using the LGscore we obtained the best correlation between the predicted and the real quality measures. However, the difference was quite small. It was also observed if Pcons was trained to predict a particular measure it performed slightly better using that measure for the evaluation. Surprisingly, it was of a greater importance what scoring was used for the structural comparisons than what was used as the goal for the predictions. The best predictions of MaxSub quality was obtained if MaxSub was used as the measure of structural similarity, etc.

5 Performance of Pcons.

As we have claimed above Pcons performs better than any single automatic method and the performance might even rival manual experts. Here, we will describe some different tests done to analyze the performance of different fold recognition servers. These are based on the LiveBench benchmarking system (Bujnicki *et al.*, 2001*a*; Bujnicki *et al.*, 2001*b*). For simplicity we have mainly used MaxSub as the evaluation method, but the results are similar using any evaluation method.

5.1 Performance in LiveBench-2.

LiveBench-2 is based on a large set of 203 proteins. Each week all new structures in PDB that does not show any significant sequence similarity to an already known protein is collected. A

Table I: Number of correct first ranked models by servers and consensus predictors.

Method	Easy (44)	Hard (155)	All (199)
FFAS	36	37	75
pdbblast	38	19	57
inbgu	39	39	83
mGenTHREADER	39	35	79
3D-PSSM	40	50	93
Sam-T99	31	29	61
fugue	35	36	74
Pcons	38	56	99
Pcons-II	42	65	107
Pcons-II-MLR	42	75	117

meta-server, submits these sequence to all participating servers, collects the predictions and analysis the results automatically. Pcons-I was trained on data from LiveBench-1 and was thus included in LiveBench-2. Results have been divided into easy and hard target, depending on the best score obtained by pdbblast. The LiveBench process is continuously proceeding and can be found at <http://bioinfo.pl/livebench/>.

There are several possible methods to analyse the results from LiveBench. In the complete analysis several methods has been used and only results that are consistent throughout all methods are considered significant. For simplicity, here we choose to analyse the results using MaxSub (Siew *et al.*, 2000). In table I it can be seen that Pcons-I does not detect significantly more correct hits for the easy targets, while for the hard targets the number of correct hits is increased by about 10% from 50 (for 3D-PSSM) to 56. This indicates a small but significant improvement for the hard targets.

Another important feature of an automatic fold recognition method is to be able to distinguish between correct and incorrect hits. In figure 3 we plot the number of correct predictions at a given number of incorrect predictions. It can be seen that the consensus predictor identifies significantly more correct hits than any single server. None of the individual servers detect more than the “easy” targets before a number of false predictions are detected, while Pcons detects more than 20 of the “hard” targets before any significant number of false positive predictions.

It is not only important to predict the correct fold it is also important to produce a high quality model for a query sequence. One method to measure the quality of the models is to use the sum of the MaxSub scores for the best model according to each server. In table II it can be seen that for the hard targets Pcons produce better models than any single server (18.7 vs 16.1 for 3D-PSSM), while there is not a significant improvement for the easy targets (14.4 vs 14.4 for 3D-PSSM).

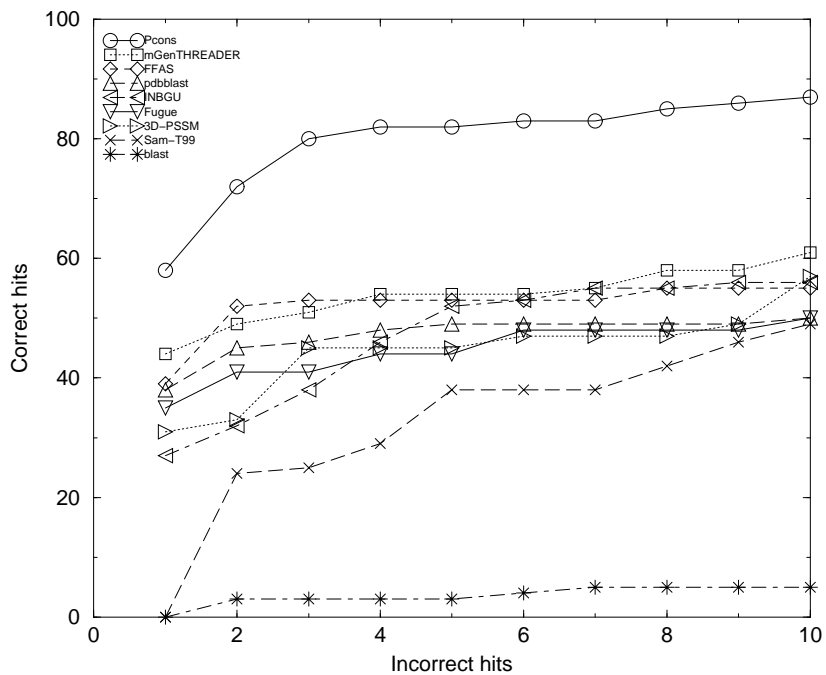


Figure 3: Cumulative plot of correct versus incorrect models in LiveBench-2. Correct and incorrect models are defined by using a combination of several different measures. The X-axis reports the number of incorrect models while the y-axis indicated number of correct models.

Table II: Sum of MaxSub scores for individual servers.

Method	Easy	Hard	All
FFAS	14.0	12.4	26.4
pdblast	14.5	5.7	20.2
inbgu	13.3	14.1	27.3
mGenTHREADER	13.0	13.4	26.4
3D-PSSM	14.4	16.1	30.6
Sam-T99	12.4	8.6	21.0
fugue	13.6	12.6	26.1
Pcons	14.4	18.7	33.1
Pcons-II	16.5	20.1	36.5
Pcons-II-MLR	15.6	22.6	38.2

Table III: Number of correct first ranked models by each server and network for the LiveBench-1 set.

Method	Easy (30)	Hard (95)	All (125)
GenTHREADER	23	13	36
Sam-T98	22	16	38
FFAS	28	14	42
Inbgu	23	21	44
3D-PSSM	22	21	43
pdblast	28	10	38
NN-score	28	20	48
NN-noscore	28	30	58
NN-model	28	29	57

5.2 Why does Pcons perform better ?

It is not obvious that Pcons should perform better than the best single protein structure predictor, however it does. In the introduction of this chapter we claimed that the main improvement was due to the “consensus analysis” used in Pcons but not by the other servers. However, it is also possible that the improvement is obtained from some other type of information.

During the development of Pcons we developed several versions of Pcons. These include: NN-score, that only used the scores from the servers as its input, NN-noscore, that only used the “consensus analysis” as its input and, NN-model which was the final version of Pcons using both scores and network (Lundström *et al.*, 2001). In table III we can see that for the easy targets the three different Pcons implementations perform equally well while for the hard targets the performance of NN-score perform worse than the other two. NN-score is the only Pcons predictor that does not use the “consensus analysis”. In fact NN-score does not perform significantly better than the best single method. This indicates that the consensus analysis is the most important contribution to the improvements obtained by Pcons. Once this type of information is included into the individual servers it is very likely that a consensus predictor does not perform significantly better than the individual servers. Interestingly, even completely ignoring the scores, from the different methods, NN-noscore performs quite well. In the development of an MLR-based Pcons we also observed that the weights for the scores were very small, indicating again that the consensus analysis is the most important part of Pcons.

6 Pcons-II.

The first version of Pcons was developed using data from LiveBench-2 and benchmarked in the LiveBench-2 process. In LiveBench-2 it was concluded that several servers had increased their performance and that some new servers performed very well. Therefore we decided to create a new version of Pcons using the data from LiveBench-2.

6.1 Improvements using more servers.

The first generation of the Pcons server used predictions from six different servers. After some minor benchmarking we decided to use a set of seven servers for Pcons-2. One completely new server was included (fugue (Shi *et al.*, 2001), while two new servers were updated (GenTHREADER to mGenTHREADER and Sam-T98 to Sam-T99.). In tables I and II these results are present as Pcons-II. It can be seen that a small, but significant, improvement is obtained using these additional servers. About 5% more correct structures are identified. The improvement is noticeable both for easy and hard targets.

6.2 Speed-up of structural comparisons.

The time for running Pcons increases with the square of the number of models. This makes it roughly twice as time-consuming to use seven servers (490 comparisons) instead of five (250 comparisons). This made the response time from Pcons to be in too long to be acceptable. We have made a significant speed-up of the structural comparison algorithm used to calculate the LGscore. This was obtained by ending the structural comparison if a fragment had an rmsd larger than $(N+225)/45$, where N is the number of residues in the fragment. For more details look at <http://www.sbc.su.se/~arne/lgscore/>.

6.3 Using better statistics.

During the evaluation of CASP4 we noted that the LGscore did not give significant scores to short proteins. To deal with this problem we have re-calculated the measure of statistical significance which are the base for the LGscore measures. The original statistical measure was obtained from a study by Levitt and Gerstein (Levitt & Gerstein, 1998). However, the statistics was only calculated for proteins larger than 120 residues, while the fragments used by LGscore often are much shorter. Using this new measures gave no significant improvements but as the statistics is better we decided to use it anyhow. As mentioned above we have also tried to use other measures of the model quality.

6.4 Improvements using Linear Regression.

Although neural-networks are a powerful tool to detect patterns, they are not always ideal. One such problem is that they might easily be over-trained and thereby not perform ideally for unseen data. To avoid these potential problems we examined the possibility to use Multiple Linear Regressions instead of the neural networks. The results are shown in tables I and II as Pcons-II-MLR. It can be seen that about 15% more correct hard targets are found using Pcons-II-MLR than the standard Pcons-II method. When studying the terms obtained from the MLR it was shown that the terms for the different scores were small in comparison with the terms for the “consensus analysis”. The model-quality for the hard targets has also been improved. In LiveBench-3 the Pcons-II-MLR method will be used.

7 Summary

In this article we show that using a “consensus analysis” fold recognition methods can be improved significantly. Using this type of analysis we think that automatic fold recognition methods can challenge the performance of manual experts.

References

- Abagyan RA, Batalov S. 1997. Do aligned sequences share the same fold ? *J. Mol. Biol.* 273:355–368.
- Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.* 25:3389–3402.
- Bairoch A, Apweiler R. 1996. The swiss-prot protein sequence data bank and its new supplement trembl. *Nucleic Acids Res.* 24:17–21.
- Bujnicki J, Elofsson A, Fischer D, Rychlewski L. 2001a. Livebench: continous benchmarking of protein structure prediction servers. *Protein Science* 10 (2):352–361.
- Bujnicki M, Elofsson A, Fischer D, Rychlewski L. 2001b. Livebench-2: large-scale automated evaluation of protein structure prediction servers. *submitted* -.
- Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A. 2001. How can the accuracy of a protein model be measured ? *BMC: bioinformatics submitted*.
- Domingues F, Lackner P, Andreeva A, Sippl MJ. 2000. Structure based evaluation of sequence comparison and fold recognition alignment accuracy. *J. Mol. Biol* 297 (4):1003–1013.
- Elofsson A. 2000. A study on how to best align protein sequences. *submitted submitted*.
- Fischer D. 2000. Hybrid fold recognition: combining sequence derived properties with evolutionary information. In *Pacific Symposium on Biocomputing*, (Altman R, Dunker A, Hunter L, Klien T, eds), vol. 5: pp. 116–.127, World Scientific.
- Fischer D, Barret C, Bryson K, Elofsson A, Godzik A, Jones D, Karplus K, Kelley L, MacCallum R, Pawowski K, Rost B, Rychlewski L, Sternberg M. 1999. Critical assessment of fully automated protein structure prediction methods. *Proteins structure function and genetics Suppl* 3:209–217.
- Fischer D, Eisenberg D. 1996. Protein fold recognition using sequence-derived predictions. *Protein Sci.* 5:947–955.
- Fischer D, Elofsson A, Rychlewski L, Pazos F, Valencia A, Rost B, Ortiz A, Dunbrack RL. 2001. Cafasp2: the critical assessment of fully automated structure prediction methods. *submitted* -.
- Gribskov M, McLachlan AD, Eisenberg D. 1987. Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci U S A* 84:4355–4358.

- Hadley C, Jones DT. 1999. A systematic comparison of protein structure classifications: scop, cath and fssp. *Structure* 7 (8):1099–1112.
- Jones D. 1999. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* 287 (4):797–815.
- Jones DT, Taylor WR, Thornton JM. 1992. A new approach to protein fold recognition. *Nature* 358:86–89.
- Karplus K, Barrett C, Hughey R. 1998. Hidden markov models for detecting remote protein homologies. *Bioinformatics* 14:846–856.
- Kelley L, MacCallum R, Sternberg M. 2000. Enhanced genome annotation using structural profiles in the program 3d-pssm. *J. Mol. Biol.* 299 (2):523–544.
- Levitt M, Gerstein M. 1998. A unified statistical framework for sequence comparison and structure comparison. *Proc Natl Acad Sci U S A* 95 (11):5913–20.
- Lindahl E, Elofsson A. 2000. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.* 295:613–625.
- Lundstrm J, Rychlewski L, Bujnicki J, Elofsson A. 2001. Pcons: a neural network based consensus predictor that improves fold recognition. *submitted* -.
- MacCallum R, Kelley LA, Sternberg M. 2000. Sawted: structure assignment with text description-enhanced detection of remote homologues with automated swiss-prot annotation comparisons. *Bioinformatics* 16 (3):125–129.
- Moult J, Hubbard T, Bryant SH, Fidelis K, Pedersen JT. 1997. Critical assesment of methods of proteins structure predictions (CASP): round II. *Proteins: Struct. Funct. Genet., Suppl* 1:2–6.
- Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247:536–540.
- Needleman SB, Wunsch CD. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48:443–453.
- Orengo CA, Michi AD, Jones S, Jones DT, Swindels MB, Thornton JM. 1997. Cath - a hierarchical classification of protein domain structures. *Structure* 5:1093–1108.
- Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T, C C. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol* 284:1201–1210.
- Park J, Teichmann SA, Hubbard T, Chothia C. 1997. Intermediate sequences increase the detection of homology between sequences. *J. Mol. Biol.* 273:249–254.

- Pearson WR, Lipman DJ. 1988. Improved tools for biological sequence analysis. *Proc. Natl. Acad. Sci. U.S.A.* 85:2444–2448.
- Rice D, Eisenberg D. 1997. A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *J. Mol. Biol.* 267:1026–1038.
- Rost B, Schneider R, Sander C. 1997. Protein fold recognition by prediction-based threading. *J. Mol. Biol.* 270:471–480.
- Rychlewski L, Jaroszewski L, Li W, Godzik A. 2000. Comparison of sequence profiles. strategies for structural predictions using sequence information. *Protein Sci.* 9 (2):232–241.
- Sauder J, Arthur J, RL Dunbrack J. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins structure function and genetics* 40:6–22.
- Shi J, Blundell T, Mizuguchi K. 2001. Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J Mol Biol.* 310 (1):243–257.
- Siew N, Elofsson A, Rychlewski L, Fischer D. 2000. Maxsub: an automated measure to assess the quality of protein structure predictions. *Bionformatics* 16 (9):776–785.
- Smith TF, Waterman MS. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147:195–197.