Cascaded multiple classifiers for secondary structure prediction [In Process Citation]

M Ouali and RD King

Protein Sci. 2000 9: 1162-1176
Access the most recent version at doi:10.1110/ps.9.6.1162

Protein Science (2000), *9*:1162–1176. Cambridge University Press. Printed in the USA.

Cascaded multiple classifiers for secondary structure prediction

MOHAMMED OUALI AND ROSS D. KING

Department of Computer Science, University of Wales, Aberystwyth Penglais, Aberystwyth, Ceredigion SY23 3DB, Wales, United Kingdom

(RECEIVED August 30, 1999; FINAL REVISION February 21, 2000; ACCEPTED March 30, 2000)

Abstract

We describe a new classifier for protein secondary structure prediction that is formed by cascading together different types of classifiers using neural networks and linear discrimination. The new classifier achieves an accuracy of 76.7% (assessed by a rigorous full Jack-knife procedure) on a new nonredundant dataset of 496 nonhomologous sequences (obtained from G.J. Barton and J.A. Cuff). This database was especially designed to train and test protein secondary structure prediction methods, and it uses a more stringent definition of homologous sequence than in previous studies. We show that it is possible to design classifiers that can highly discriminate the three classes (H, E, C) with an accuracy of up to 78% for β-strands, using only a local window and resampling techniques. This indicates that the importance of long-range interactions for the prediction of β-strands has been probably previously overestimated.

Keywords: neural network; prediction; protein; secondary structure; statistics

Although the protein folding process may require catalysts such as chaperonins (Hubbard & Sander, 1991), it is widely accepted that the three-dimensional (3D) structure of a protein is related to its sequence of amino acids (Epstein et al., 1963; Anfinsen, 1973; Ewbank & Creighton, 1992; Baldwin & Rose, 1999). This implies that it is possible to predict protein structure from sequence with high accuracy. The most general and reliable way of obtaining structural information from protein sequence data is to predict secondary structure. The aim of secondary structure prediction is to extract the maximum information from the primary sequence in the absence of a known 3D structure or a homologous sequence of known structure. With the increasing number of amino acid sequences generated by large-scale sequencing projects, and the continuing shortfall in crystallized homologous structure, the need for reliable structural prediction methods becomes ever greater.

Many approaches have been proposed to tackle this problem, and they can be approximately grouped into those using simple linear statistics either on residues or physicochemical properties or even both (Robson & Pain, 1971; Chou & Fasman, 1974; Lim, 1974; Robson & Suzuki, 1976; Garnier et al., 1978; Cohen et al., 1983; Ptitsyn & Finkelstein, 1983; Gibrat et al., 1987; King & Sternberg, 1996; Avbelj & Fele, 1998); those using symbolic machine learning (King & Sternberg, 1990; Muggleton et al., 1992); and those using sophisticated nonlinear statistical methods for prediction, which are often based either on neural networks exploiting patterns of residues and/or physicochemical properties (Qian & Sejnowski, 1988; Holley & Karplus, 1989; Kneller et al., 1990; Rost & Sander, 1993; Riis & Krogh, 1996; Kawabata & Doi, 1997; Baldi et al., 1999; Jones, 1999) or on k-nearest-neighbor methods (Biou et al., 1988; Zhang & Chou, 1992; Yi & Lander, 1993; Geourjon & Deleage, 1994; Salamov & Solovyev, 1995, 1997; Frishman & Argos, 1996, 1997; Levin, 1997). A fair comparative assessment of these different methods turns out to be difficult, as they use different datasets for the learning process and different secondary structure assignments (Cuff & Barton, 1999). However, a number of authors have designed methods with accuracies above the threshold of 70% accuracy taking advantage from multiple sequence alignments (Rost & Sander, 1993; Salamov & Solovyev, 1995, 1997; King & Sternberg, 1996; Levin, 1997) or selected pairwise alignment fragments (Frishman & Argos, 1997). These accuracies have been confirmed in the series of CASP blind trials (http://PredictionCenter.llnl.gov/).

In this paper, we present the results of an in-depth analysis of the performance of a new classifier for protein secondary structure prediction Prof (Protein forecasting). Prof is formed by cascading (in multiple stages) different types of classifiers using neural networks and linear discrimination. To generate the different classifiers, we have used both GOR formalism-based methods extended by linear and quadratic discriminations (Garnier et al., 1978, 1996; Gibrat et al., 1987), and neural network-based methods (Qian & Sejnowski, 1988; Rost & Sander, 1993). The theoretical foundation for Prof comes from basic probability theory, which states that all of the evidence relevant to a prediction should be used in making that prediction (Jaynes, 1994). This means that it should be possible to improve predictions by combining different algorithms or the same one trained in different ways or on different sets, as

Reprint requests to: Mohammed Ouali, Department of Computer Science, University of Wales, Aberystwyth Penglais, Aberystwyth, Ceredigion SY23 3DB, Wales, United Kingdom; e-mail: mho@aber.ac.uk.

long as the classifiers produce noncorrelated errors (i.e., if the produced errors do not all correlate with each other).

Prof represents a compromise between classifiers having different properties and achieves a global accuracy per residue of 76.7% on our nonhomologous data set, using a full jack-knife testing procedure (leave-one-out cross-validation).

We analyze the performance of each classifier and compare them with and without the use of evolutionary information (multiple alignments). We show that it is possible to obtain classifiers with global accuracies at better than 75% and capable of predicting β-strands with an accuracy per residue of better than 77–78% (with α-helix predicted at better than 79% and coils at better than 71%). While it has long been argued that the lower accuracy for β-strands was mainly due to the fact that all secondary structure methods do not take into account long-range interactions, and some attempts have been published using a double window for β-strands predictions to overcome this difficulty (Krogh & Riis, 1996; Frishman & Argos, 1997). Our results indicate that the importance of long-range interactions for the prediction of β-strands has been probably overestimated up to now.

Results and discussion

Assessment of secondary structure classifiers without using evolutionary information/GOR methods vs. single neural networks

Table 1 shows the evaluation of five different GOR methods (Garnier et al., 1978, 1996; Gibrat et al., 1987) and their combinations using linear and quadratic discriminations. To the best of our knowledge, this is the first time that an exhaustive comparison on the same database of all the GOR algorithms has been published. Surprisingly, a GOR I algorithm that uses probabilities to perform the classification task exhibits a higher estimated Q3 per residue than both GORIII and GORIV. This result is confirmed by the analysis of the Matthews' correlation coefficients. We found that GORIV, on our database, has an estimated accuracy per residue of 61.3%, while the authors give an estimate of 64.4%. We confirm the estimate of Cuff and Barton (1999), who show a reduction of the accuracy by 4% using a similar procedure to three states reduction from DSSP (Kabsch & Sander, 1983). This result underlines the difficulties of comparing different methods from different papers, and the importance of the reduction protocol. The measurements of the accuracy per proteins instead of per residue confirm these observations (data not shown), although the Sov (segment overlap measure) (Table 1) for GORIV is globally the same as for GORI. The Sov measures for the GOR III are particularly poor, and in all cases the global Sov does not exceed 60%, implying a lack of correlation in the prediction of adjacent residues at this stage. The addition of pair information (information a residue carries about another residue's secondary structure that does depend on the other residue's type) and the so-called pair–pair information does not increase the global Q3. The principal effect of using the probabilities to make a decision, rather than simply taking the state having the highest information value, is that the prediction then reflects the proportion of the three states (H, E, C) in the database. When the decision is taken on the information basis, β-strands are better predicted and a decrease of the Q_c is observed. The reason that the use of probabilities can lead to a different answer from the information is explained by Figure 1. This shows that with the same algorithm it is possible to design two very different classifiers. This is a key observation in the formation of multiple classifier combinations for improving secondary structure prediction.

The accuracy of GOR methodologies can also be improved by using simple linear discrimination. The vector used consists of the three information values of each classifier using only information and the two probabilities (probabilities for α-helix and β-strand) for the classifiers using probabilities (Table 1). A gain of more than 2% for the Q_3 is observed over GORI using probabilities. That this combination produces a better classifier is also clearly shown by the examination of the Matthews' correlation coefficients. A quadratic discrimination was performed on the results of the linear

Table 1. *Statistical analysis of the different GOR methods and neural network method without the use of multiple alignment*[a]

Method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	Sov (% (p))	Sov_H (% (p))	Sov_E (% (p))	Sov_C (% (p))
GOR I (Information)	60.7	64.7	57.8	58.9	0.420	0.371	0.408	56.4 ± 12.1	57.1 ± 25.8	66.9 ± 22.0	53.1 ± 14.9
GOR I (Probability)	62.3	65.0	37.0	72.4	0.422	0.360	0.406	56.7 ± 12.5	59.0 ± 26.1	52.1 ± 26.7	59.5 ± 14.7
GOR III (Information)	59.0	68.8	56.6	52.5	0.416	0.347	0.383	43.2 ± 11.4	49.6 ± 24.4	59.5 ± 23.4	40.9 ± 13.6
GOR III (Probability)	61.1	70.2	42.3	63.0	0.420	0.348	0.395	45.1 ± 11.3	51.7 ± 24.3	50.8 ± 25.8	47.20 ± 14.7
GOR IV	61.3	69.3	43.9	63.4	0.463	0.315	0.387	56.9 ± 12.2	62.2 ± 25.3	56.8 ± 25.3	53.5 ± 15.5
GOR (linear reg.)	64.3	64.7	41.8	75.0	0.467	0.388	0.432	57.0 ± 11.6	57.4 ± 25.7	56.1 ± 25.6	61.6 ± 14.3
GOR (quadratic reg.)	62.3	71.3	54.8	58.8	0.464	0.403	0.391	57.5 ± 13.2	59.9 ± 25.7	62.5 ± 25.2	54.3 ± 15.3
Neural network (u)	65.3	65.9	44.6	75.0	0.494	0.399	0.446	55.6 ± 12.4	56.9 ± 25.6	55.5 ± 26.2	60.6 ± 14.8
Neural network (b)	64.0	65.4	62.8	63.4	0.491	0.412	0.445	56.5 ± 12.8	57.3 ± 25.7	66.5 ± 22.9	56.7 ± 15.3

[a] Q_3 is the accuracy per residue (see Materials and methods); Q_H, Q_E, and Q_C are the accuracies for α-helix, β-strand, and coil, respectively. C_H, C_E, and C_C are the Matthews' correlation coefficients for α-helix, β-strand, and coil, respectively. The Sov is the averaged segment overlaps measure per protein for the three states. Sov_H , Sov_E, and Sov_C are the averaged segment overlap per protein for α-helix, β-strand, and coil, respectively; the corresponding standard deviations are shown. This table summarizes the statistics for the different GOR methods and the neural-network methods without the use of multiple alignment. GOR I (information) is the GOR I algorithm using only the three computed information values for the decision process. GOR I (probability) is the GOR I algorithm with an explicit computation of the probability of each class (the decision is taken on the basis of the highest probability). Same for GOR III (information) and GOR III (probability). GOR (linear reg.) represents a combination of the five GOR algorithms using linear discrimination. GOR (quadratic reg.) is a quadratic discrimination over the GOR (linear reg.) algorithm using a window of seven residues. Neural network (u) is the network trained in an unbalanced. Neural network (b) states for the network trained in a balanced way.

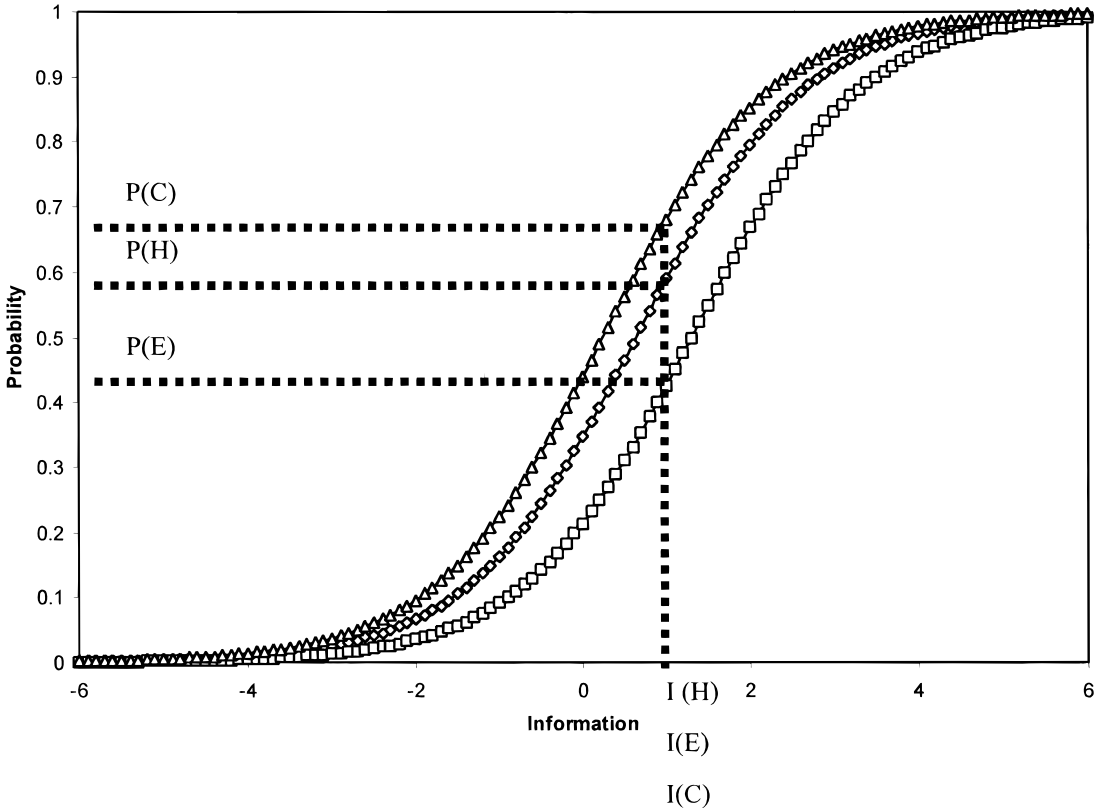

Fig. 1. Computed curves displaying the probability vs. the information values using Equation 9. The curve on the left side shows the relationship between the information values for the coil and the corresponding probability, the middle one for α-helix and right one for β-strands. Each curve depends on the prior probability of the considered class. For a same of information value (in the area of the slopes): the probability for coil will be higher than the probability for α-helix and β-strand, the probability of α-helix will be higher than the probability of β-strand. Due to the observed shifts, using the probability instead of the information will favor the prediction of first the C state over the E state, second of the H state over the E states, third of the C state over the H state. Therefore, this will lead mainly to an underprediction of the E state.

discrimination using a window of seven residues (the components of the vector are the probabilities for α-helix and β-strand). The result is an improvement over the prediction of H and E states with respect to the five methods. It was not possible to improve the global accuracy using quadratic discrimination. We used linear and quadratic discriminations to produce "new" classifiers. As it was possible to obtain an improvement over the GOR methods, we conclude that the errors produced by the different classifiers are not all correlated.

Table 1 also shows the evaluation of a single three-layered neural network trained in both a balanced and unbalanced way. The use of the unbalanced network is formally equivalent to the use of the observed class distribution as prior probabilities for each class (H, E, C) in the learning process: while the balanced network is equivalent to the use of uniform prior probabilities—in each epoch a random resampling is performed to achieve the redistribution of (1, 1, 1) for each class. The networks we used contained 13×21 input cells (20 residues + gaps), the hidden layer contained 30 cells, and the output had 3 cells. The neural network trained in an unbalanced way has an accuracy per residue of better than 65%, while the balanced one showed a decrease in the global accuracy of ~1%. All the methods that explicitly take into account the prior probability of occurrence for each class fail to accurately predict β-strands. The Matthews' correlation coefficients show that the neural network method is more accurate than any of the GOR algorithms when analyzed at the residue level, while at the segment level the performance was rather similar (Table 1). However, the Sov should not be used to assess the performance of a classifier, but rather to assess the quality and the usefulness of a prediction as the Sov can be improved by applying a second "structure-to-structure" network (Rost & Sander, 1993) or simple smoothing filters (King & Sternberg, 1996; Zimmerman & Gibrat, 1998). By using such a strategy, one can take into account (at least in part) the correlation between adjacent residues.

Assessment of GOR methods using evolutionary information (multiple sequence alignment)/first stage of our classifier

The alignment of homologous sequences provides additional information for predicting secondary structure. When dealing with statistical methods, the simplest way of using this extra information is to average the GOR information or probabilities over the aligned residues. This is equivalent to extending the GOR prediction algorithms to include homologous information (Zvelebil et al., 1987). All the proteins used in our multiple alignment were unique and had a minimum of 25% sequence identity with respect to the target sequence, insertions in the multiple alignments are ignored, and each sequence is predicted without any insertions, then the

average took place. Table 2 shows the analysis of this experiment. By using multiple alignments, it was possible to improve the accuracy of the different GOR algorithms by 4–5% over that of a single sequence. The best algorithm was still found to be the combination of all the GOR algorithms using linear discrimination: this method achieves a Q_3 per residue of 68.7% and a Q_3 per protein of 69% over the whole database (data not shown). α-Helices and β-strands are better discriminated as shown by the systematic improvement of the Matthews' correlation coefficients. This indicates that the use of multiple alignment diminishes the number of false positives and false negatives. The Sov is improved by 4–7% depending on the method used. The combined method using quadratic discrimination over a window of seven adjacent residues exhibits the highest value for Sov (more than 64%) as expected, since this kind of discrimination allows the correlation between adjacent residues to be taken into account.

This improvement using multiple aligned sequences agrees with the work of Zvelebil et al. (1987), who also found a mean improvement of 4% in accuracy on a set of 11 protein families. Levin et al. (1993) have obtained a mean improvement of 8% over seven protein families, using alignments obtained by spatial superposition of main-chain atoms in known tertiary protein structures, and they obtained using an automated procedure of multiple alignment an improvement of around 6.8%. It is difficult to draw firm statistical conclusions from this previous work (about the expected increase in accuracy obtained by using multiple alignments), but we recognize that our procedure is clearly far from optimal.

However, we will show that it is still possible to extract more information by exploiting the generation of multiple classifiers.

Generation of multiple neural network using evolutionary information, second stage of the classifier

We compared the combined GOR methods using linear discrimination and quadratic discrimination with neural networks. We combined the 7 GOR methods using small neural networks having 21 inputs over a window of 7 residues, a single hidden layer of 14 cells, and as usual 3 output cells. We learned the output of the different GOR methods, namely information and probabilities, without any normalization procedure. The chosen strategy was to learn only the residues (output of GOR) that exhibit no consensus in the prediction over the seven GOR methods, since the produced errors are uncorrelated. The residues for which a consensus existed between all the seven methods were simply passed through another similar network to produce an homogenate output. This was done in both a balanced and an unbalanced way. Interestingly, when the seven GOR methods agree each other, the global accuracy is 78% on the subset of residues with consensus, while the accuracy is only 55% on the subset of residues without consensus between the classifiers. Using such a procedure, it is possible to boost the GOR method to 71.4% (using the per-residue accuracy) for the unbalanced trained network and to 70% for the balanced one, which represents an improvement of 2% over linear discrimination and more than 5% over any individual GOR algorithm; the Sov is also improved (Table 3). The increase of the global accuracy is explained by the fact that the subset of residues without consensus is predicted correctly at 61% after the neural network step, which represents an improvement of 7% on this subset. Characteristically, the consensus subset always exhibits a global accuracy of 78%. This combination of GOR algorithms generates a classifier where β-strands and α-helices are better discriminated as shown by the Matthews' correlation coefficients.

Another simple and direct way of using multiple aligned sequences when dealing with neural networks is to compute the corresponding profile. We compute the profile first by explicitly counting the gaps (profile 1) and second by ignoring the gaps (profile 2). The architecture of these networks is the same as the one used for single sequences. This produces different classifiers whose characteristics are shown in Table 3. Their accuracies per residue are at ~71%, which represents an improvement of 5% over the neural networks using only single sequences, as in the case of GOR.

Recently, at the CASP3 meeting (third meeting on the critical assessment of techniques for protein structure prediction) (http://PredictionCenter.llnl.gov/casp3/Casp3.html), D. Jones used the profile generated by PSI-BLAST to design a set of networks that performed particularly well (Jones, 1999). This procedure has the following basic advantages: more distant sequences are found; the probability of each residue at a specific position is computed using a more rigorous statistical approach; and each sequence is properly weighted with respect to the amount of information it carries (Altschul et al., 1997). This way of using multiple alignments is a step forward. We therefore also made use of PSI-BLAST profiles in an analogous manner to the work of D. Jones. The NR database was filtered to remove segment with low complexity (Jones, 1999). For direct comparison, we used the same architecture for the neural network as D. Jones, namely 17×20 input cells and 75 cells for the hidden layer used with three outputs cells (however, this architecture only produced a small difference on global accuracy from our standard architecture). This network was trained in a

Table 2. *Statistical analysis of the different GOR algorithms using multiple alignment*[a]

Method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	Sov (% (p))	Sov_H (% (p))	Sov_E (% (p))	Sov_C (% (p))
GOR I (Information)	65.3	69.2	61.3	64.3	0.499	0.440	0.461	61.0 ± 13.6	61.3 ± 26.6	71.6 ± 22.8	57.4 ± 15.9
GOR I (Probability)	66.3	68.8	36.2	79.0	0.499	0.415	0.462	59.8 ± 13.2	63.1 ± 27.2	52.0 ± 27.9	63.0 ± 14.1
GOR III (Information)	64.4	75.7	62.0	56.7	0.505	0.437	0.445	50.3 ± 13.3	57.7 ± 24.8	66.5 ± 23.3	45.4 ± 15.2
GOR III (Probability)	65.8	76.0	42.7	69.0	0.497	0.421	0.459	52.3 ± 13.2	60.4 ± 25.2	53.4 ± 26.0	53.2 ± 15.8
GOR IV	65.4	74.7	42.3	69.3	0.528	0.376	0.438	60.8 ± 13.0	68.1 ± 25.7	57.1 ± 25.9	57.2 ± 16.1
GOR (linear reg.)	68.7	68.2	47.2	79.5	0.552	0.463	0.487	62.7 ± 13.3	63.1 ± 26.9	61.1 ± 25.5	65.5 ± 14.4
GOR (quadratic reg.)	68.0	73.8	61.5	66.6	0.566	0.481	0.465	64.5 ± 13.4	66.7 ± 26.3	68.3 ± 24.3	61.7 ± 14.9

[a] Same nomenclature as Table 1 after the use of multiple alignment.

Table 3. *Statistical analysis of all the classifiers forming the second stage of Prof* [a]

Method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	Sov (% (p))	Sov_H (% (p))	Sov_E (% (p))	Sov_C (% (p))
NN-GOR (u)	71.4	71.7	56.8	78.2	0.610	0.516	0.515	66.4 ± 14.4	66.8 ± 27.4	67.2 ± 24.8	67.6 ± 15.4
NN-GOR (b)	69.8	74.7	69.0	66.3	0.599	0.516	0.499	64.1 ± 13.7	65.8 ± 25.9	72.1 ± 22.1	61.5 ± 15.2
NN profile 1 (u)	70.6	70.2	55.0	78.5	0.585	0.500	0.515	64.1 ± 13.4	64.5 ± 25.9	66.0 ± 24.7	66.6 ± 14.8
NN profile 1 (b)	69.1	72.1	67.6	67.6	0.585	0.496	0.497	60.9 ± 14.7	63.3 ± 25.9	70.7 ± 22.1	59.0 ± 16.5
NN profile 2 (u)	70.2	71.5	55.0	77.5	0.590	0.503	0.515	62.7 ± 13.9	63.7 ± 26.9	64.3 ± 25.2	65.0 ± 15.1
NN profile 2 (b)	69.3	72.3	68.6	67.3	0.587	0.502	0.502	62.6 ± 13.9	63.4 ± 26.0	72.4 ± 21.9	60.9 ± 16.0
NN profile-blast (u)	73.6	75.8	60.9	78.0	0.650	0.564	0.530	62.9 ± 14.5	65.5 ± 26.7	67.6 ± 24.9	64.4 ± 16.1
NN profile-blast (b)	72.5	76.6	74.6	68.4	0.651	0.571	0.524	62.4 ± 14.5	65.8 ± 25.9	74.8 ± 21.6	59.6 ± 16.2

[a] Same nomenclature as Table 1 for the statistics. All the classifiers make use of multiple aligned sequences. NN-GOR (u) states for the combination of the seven GOR methods after the use of multiple alignment by a neural network trained in an unbalanced way. NN-GOR (b) states for the same combination with a neural network trained in a balanced way. NN profile 1 states for the neural networks taking as input the profile computed with gaps, which means that the profile is computed by treating gaps as a simple residue. NN profile 2 states for the networks taking as input a profile without gaps (Rost & Sander, 1993). NN profile psi-blast states for the networks taking as input the profile derived from PSI-BLAST. (u) and (b) states always for the way of training: unbalanced and balanced, respectively.

balanced and unbalanced way to generate classifiers with different properties. We obtained two classifiers whose accuracies per residues are 73.6 and 72.5%, respectively, which represents an improvement of 2% over NN-GOR and 2 to 3% over the neural network using a standard profile (profile 1 or 2) (Table 3). It is also an improvement of more than 7 to 8% over the neural network using only single sequences. The Matthews' correlation coefficients also show that the PSI-BLAST profile carries more information, as the three coefficients are improved by 1.5 to 5%. α-Helices and β-strands are also better discriminated. The Sov measurements indicate that at this stage the "Psi-Blast" networks do not perform better than the other profile-networks and that NN-GOR networks are even better from this point of view: this is to be excepted since a better Sov requires a second step of filtering or regularization. Our work confirms the results of D. Jones and shows that such classifiers have different properties, but the gap between "standard profiles" and "psi-blast profiles" is not as wide as expected from the Q_3 point of view as previously suggested at the CASP3 meeting. However, more information does seem to be extracted from the PSI-BLAST profile by the networks. This is explained partly by the fact that the learning process occurs over the profiles and that somehow we are learning directly from the multiple alignment, which was not the case, for example, when we used a simple average over the multiple alignment with the GOR algorithms.

The method of Jones using the PSI-BLAST profile has an average estimated accuracy per residue of 76.5%, based on a benchmark of 187 unique protein folds with full cross validation. By chain, the mean Q_3 score is 76.0% with a standard deviation of 7.8% (http://globin.bio.warwick.ac.uk/psipred/psipred_info.html). To achieve this result, the author used a large database of 1,887 proteins where the threshold for sequence identity is 95%. Those proteins form the "N-level" in the CATH database (Orengo et al., 1997). For cross validation, he used fold similarity and sequence identity to exclude chains from that list—chains that share a common domain fold (i.e., have a domain with identical CATH numbers) or that have a sequence identity superior to 25% to the test protein chains are excluded from the training set (Jones, 1999). The use of only the PSI-BLAST profile does not alone produce as high an accuracy as 76.5%, as we have demonstrated on our own database that was constructed using a strict homology cutoff (similarity score (SD) less than 5; see Materials and methods). We therefore speculate that the PSI-PRED method of Jones (1999) obtains its high accuracy by exploiting the extra information available in homologous tertiary structures. Indeed, the fact that two homologous proteins share the same fold does not imply that the secondary structure of the two homologues are strictly similar; on the contrary, differences are expect to be observed. As stated in the introduction, it is a basic rule in statistics that all relevant information should be used in predictions. PSI-PRED is the first prediction method to exploit multiple homologous tertiary structures. Previous methods (and our approach) have avoided using such data because of the danger of biasing toward folds with many structures. We therefore believe that PSI-PRED obtains high accuracy by use of this new source of information (3.8 times more structures): while Prof obtains high accuracy by more efficient use of data. If this is true, it may be possible to combine such approaches to produce a method with higher accuracy than either PSI-PRED or Prof.

Combining all the generated neural network from stage 2 to achieve higher accuracy/third stage of the classifier

The third stage of our classifier consists of combining the eight different classifiers generated from stage 2 (Fig. 2). To do this, we use both simple linear discrimination and a neural network trained in a balanced and an unbalanced way. The linear discrimination uses a vector whose dimension is 24 (three output values by classifiers), and no correlations between adjacent residues are introduced. The architecture of the network used at this stage consists of 24×13 input cells (we use a window of 13 residues to predict the central residue); the hidden layer is made of 40 cells and it has three outputs. By using such a strategy, we are again able to produce a set of highly accurate and different classifiers all better than 75% (accuracy per residue). We obtained an improvement of 2.6% at this stage with respect to stage 2 achieving even a per-residue accuracy of 76.2% for the best classifier at this level. The properties of each classifier are analyzed in Table 4. The Matthews' correlation coefficients show an improvement of 4 to 5% uniformly over the three states. The Sov measurements show an im-

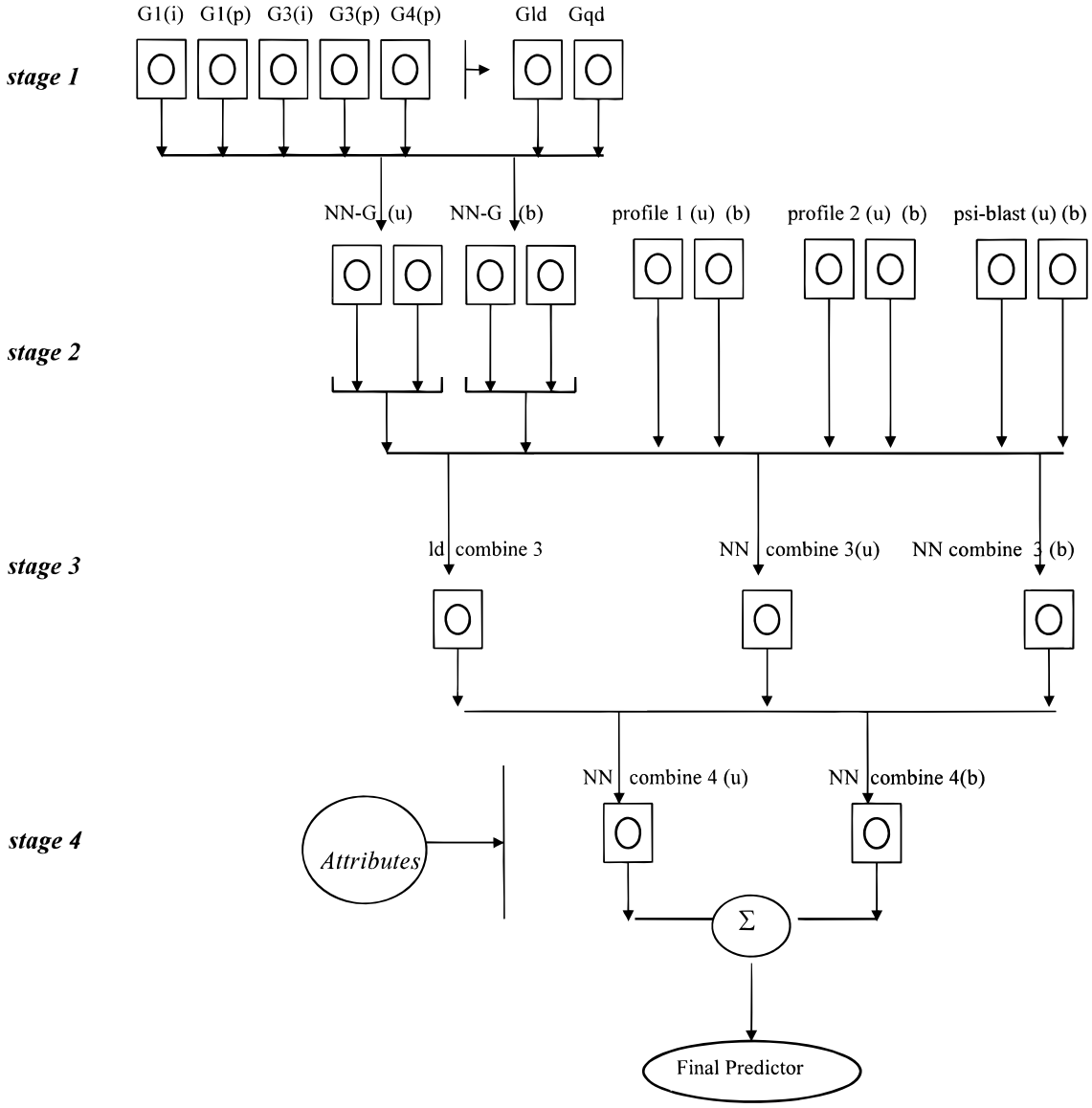

Fig. 2. Architecture of the cascaded multiple classifier Prof. G stands for GOR; (i) for information; (p) for probability; NN for neural network; ld for linear discrimination; qd for quadratic discrimination; (u) for neural networks trained in an unbalanced way; (b) for neural networks trained in a balanced way. Stage 1 is constituted by GOR algorithms. Stage 2 contains the combination of GOR algorithms using neural networks (NN-G) and also neural networks using different profiles (profile 1 and 2), PSI-BLAST profiles (psi-blast). Stage 3 uses outputs from stage 2 combined by linear discrimination (ld combine 3) and neural networks (NN combine 3). Stage 4 uses outputs from stage 3 and the set of attributes (see text) to produce new classifiers using neural networks. These networks are then averaged.

provement of ~6%, which is to be expected, since this step can be also seen as a generalization of the second level of prediction in PHD (the level structure to structure) (Rost & Sander, 1993). Furthermore, the network trained in a balanced way achieves a global accuracy of only 75.1% but the accuracies for α-helix and β-strands are 79.6 and 77%, respectively; the ability of discriminating between the three states is high as indicated by the Matthews' coefficients.

This result undermines the argument that β-strands are poorly predicted mainly because the stabilization of such a structure requires long-range interactions (to form β-sheets) that cannot be captured using a single local window (Frishman & Argos, 1996; Garnier et al., 1996). We have shown that it is possible to predict with high accuracy β-strands using a single window and resampling techniques, confirming the earlier results of Rost and Sander (1993). Our results suggest that perhaps the lower accuracy for β-strands is due mostly to the way the data are represented and their frequency distribution.

Adding attributes to the classifiers/fourth and fifth stages of the classifier

King and Sternberg (1996) showed that it was possible to boost the GOR I algorithm using further attributes combined by linear dis-

Table 4. *Statistical analysis of stages 3, 4, and 5 of Prof* [a]

Method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	Sov (% (p))	Sov_H (% (p))	Sov_E (% (p))	Sov_C (% (p))
Combine stage 3 (linear)	75.7	75.7	61.9	82.3	0.686	0.599	0.568	69.6 ± 14.0	69.5 ± 27.7	71.0 ± 24.2	70.7 ± 14.7
Combine stage 3 NN (u)	76.2	77.7	64.9	80.5	0.699	0.608	0.571	72.6 ± 13.5	72.8 ± 27.6	72.9 ± 24.9	71.3 ± 14.6
Combine stage 3 NN (b)	75.1	79.6	77.0	70.6	0.699	0.604	0.558	71.7 ± 13.6	73.2 ± 26.9	77.7 ± 22.0	67.7 ± 15.3
Combine stage 4 NN (u)	76.8	78.1	66.7	80.7	0.709	0.626	0.576	73.5 ± 13.6	71.0 ± 29.9	72.8 ± 27.2	72.2 ± 14.4
Combine stage 4 NN (b)	75.7	79.7	78.1	71.5	0.710	0.619	0.561	72.4 ± 14.0	71.0 ± 29.8	77.4 ± 24.4	68.0 ± 15.0
Average stage 5	76.7	78.8	71.6	77.6	0.710	0.629	0.574	73.7 ± 13.9	71.1 ± 29.9	75.6 ± 26.0	71.1 ± 15.0

[a] Same nomenclature as Table 1 for the statistics. Combine stage 3 (linear) is the classifier obtained by combining the eight neural networks from stage 2 (see Fig. 2) using linear discrimination. Combine stage 3 NN (u) and (b) state for the neural networks using as input the output of the eight networks of stage 2 trained, respectively, in an unbalanced and balanced way. Combine stage 4 NN (u) and (b) are the networks combining the three methods of stage 3 (linear discrimination and 2 networks) taking as input on one hand the output of stage 3 and, on the other hand, the computed attributes (moment of hydrophobicity assuming an α-helix and a β-strand, fraction of residues H, E, Q, D, R in the sequence, fraction of predicted α-helix and β-strand). Average stage 5 represents the final classifier obtained by averaging the two classifiers of stage 4.

crimination; they also obtained better balanced predictions by using these attributes. In this work, we add to the three outputs of the three classifiers of the previous stage the following selected attributes (we selected only the attributes that gave improvements): the moment of hydrophobicity (Eisenberg, 1984) is computed for each residue over a central window of seven under the assumption that these residues are in α-helix; the moment of hydrophobicity assuming a β-strand conformation; we add also the fraction of the following residues H, E, Q, D, R, as well as the fraction of α-helix and β-strand (computed from the averaged three classifiers of the third stage). The architecture of the used networks are 13×18 input cells (window of 13 and 18 values), the hidden layer contains 30 cells, and we have 3 output cells. The network has been trained both in a balanced and unbalanced way. Results are presented in Table 4. An improvement of 0.7% is observed on the global accuracies. The accuracy for the unbalanced trained network is close to 77% while the balanced one is very close to 76% and exhibits even higher accuracy for β-strand and α-helix. The prediction of the β-strand and α-helix population is improved by 1 to 2% over the accuracies as well as the Matthews' coefficients. We therefore conclude that these attributes aid in the discrimination of the three classes. The Sov measurements are also improved (Table 4). Finally, we average the two classifiers of the fourth stage, which constitutes the final classifier that we call Prof. We then obtain a classifier that has an estimated global accuracy of around 77%; it predicts the α-helix at 79%, the β-strand at 71.6%, and coils at 77.6%. This represents a compromise between the balanced and the unbalanced way of training a neural network. This is the first time, to the best of our knowledge, that a classifier predicts β-strand with such high accuracy (the statistical analysis of the classifier is shown in Table 4).

Proteins can be classified into four structural classes (Zhang & Chou, 1992; Rost, 1996). We analyzed Prof using this classification. The final classifier has an accuracy per protein of 79.6% and a Sov of 76.3% on the all-α family (helix $\geq$ 45%, strand $<$ 5%), on the all-β family (strand $\geq$ 45%, helix $<$ 5%); the algorithm has an accuracy of 76% with a Sov of 77.8%. For the α/β family (helix $\geq$ 30%, strand $\geq$ 20%), the accuracy per protein is also 76% and the Sov is 76.5%. All the other proteins have an averaged accuracy per protein of 75.5% with an Sov of 72.3%. A tool for assisting in tertiary structure prediction should allow the user to choose between the three final classifiers. The distribution of the accuracy per residue and the Sov (per protein) show that there are still a small number of proteins that are poorly predicted (Fig. 3).

About the influence of alternative eight to three states decompositions (from DSSP) on Prof

DSSP provides an eight states assignment of secondary structure (Kabsch & Sander, 1983). However, all the available prediction methods are normally trained to predict three states (H, E, C). It has been argued recently that the way of decomposing these eight states could have a dramatic effect on the accuracy of a method (Cuff & Barton, 1999). We then have tested Prof using different decomposition methods. The results are presented in detail in Table 5. Our goal is not to argue about the best way of decomposing the eight states of DSSP into three states, as we think that all these methods are defensible from a structural point of view. Instead, our goal is to give a complete view of the performance of Prof using different definitions.

In this paper, as stated in Materials and methods, we have used the following conservative mapping to train the method: H, I, and G states from DSSP are translated as α-helix (H), E is translated as β-strands (E), and the remainder is translated as coil (C). Using this mapping achieves an accuracy of 76.7%.

However, some authors used the following decomposition: E and B as (E), G and H as (H), and the rest as (C) (Cuff & Barton, 1999). This decomposition (Method A) treats isolated β-bridges as part of a β-sheet (E). This increases the proportion of state (E). One has to keep in mind at this stage that Prof has not been trained with this decomposition so a decrease of the accuracy is obviously to be expected. Nonetheless, with this method Prof still achieves an accuracy per residue of 76%. This is a decrease of 0.7% with respect to our previous estimation. The β-strand are still predicted with an accuracy per residue of 68.4% instead of 71.6%; this is a decrease of 3.2% while the increase of the total population of the state (E) is of 6.1%. This level of accuracy in the (E) states is still the highest accuracy ever reported to the best of our knowledge.

Rost and Sander (1993) have used another decomposition (Method B). In this method, H, G, and I are translated in (H), E is translated into (E), B is translated in (EE), and BB is translated in (CCC). The remainder is translated into (C). This represents an increase in the (E) population of 6.8%. Prof achieves an accuracy of 76% per protein and 75.8% per residue. This is a decrease of 0.9% over our estimated accuracy of Prof. But Prof still exhibits a high accuracy for the prediction of β-strands. One has to remark as well that with these two decomposition methods, no decrease of the accuracy of (H) and (C) states is observed.

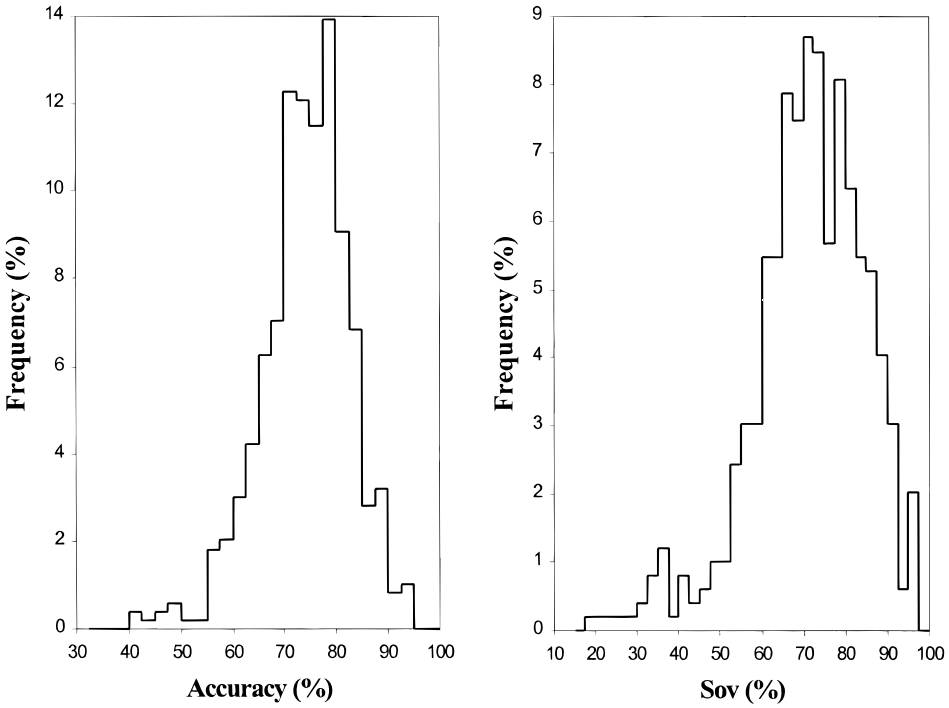

Fig. 3. Distribution of the Q_3 per protein and the Sov of the cascaded multiple classifier.

Frishman and Argos (1997) have used another decompositon (Method C). They translated E as (E), H as (H), and the rest into (C), including EE and HHHH. Table 5 shows the results. With this decomposition, Prof achieves an accuracy per residue of 77.9% and an accuracy per protein of 77.8%. An increase of the prediction of the (H), (E) states is shown as expected since short helices and short β-strands (EE) are difficult to predict partly because they are less stable. This represents an improvement of 1% over our method of decomposition.

Salamov and Solovyev (1995) used the following decomposition (Method D): GGGHHHH are translated into HHHHHHH, B and G are redefined as (C), E are translated as (E), and H as (H), the remainder as (C). Using this decomposition method, Prof achieves an accuracy per residue of 77.8 and 77.7% per protein. β-Strands are predicted with the same accuracy as with our decomposition while α-helices are better predicted. This represents an improvement of 1% over our decomposition.

These experiments using different decomposition methods give a better idea of the performance of Prof. Furthermore, it shows that our algorithm is very stable with respect to the decomposition method since a variation of only $\pm 1\%$ is observed over the global accuracy. The β-strands state (E) is obviously the most sensitive to the way of translating the β-bridges (B), but at the residue level the fluctuations are $\pm 3\%$.

Table 5. *Statistical analysis of the performance of Prof using different three states decompositions*[a]

Decomposition method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_C (%)	C_H	C_E	C_C	Sov (% (p))	Sov_H (% (p))	Sov_E (% (p))	Sov_C (% (p))
Our method	76.7	78.8	71.6	77.6	0.710	0.629	0.574	73.7 ± 13.9	71.1 ± 29.9	75.6 ± 26.0	71.1 ± 15.0
Method A	76.0	78.7	68.4	77.7	0.709	0.613	0.561	72.8 ± 13.5	71.1 ± 30.0	67.5 ± 29.5	70.0 ± 13.9
Method B	75.8	78.7	68.1	77.7	0.709	0.613	0.559	73.1 ± 13.6	71.1 ± 29.9	68.4 ± 28.9	70.8 ± 14.1
Method C	77.9	84.5	73.4	75.6	0.735	0.630	0.595	72.5 ± 15.5	81.4 ± 25.5	81.3 ± 22.7	67.6 ± 17.8
Method D	77.8	84.5	71.5	76.2	0.735	0.628	0.592	74.0 ± 14.3	81.4 ± 25.5	75.5 ± 26.0	69.9 ± 16.4

[a]Same nomenclature as Table 1 for the statistic per residue. We used the following decomposition methods:

Our method: H, G, I → (H) ; E → (E) ; the remainder → (C)
Method A: H, G → (H); E and B → (E); the remainder → (C)
Method B: H, G, I → (H); E → (E) but B → (EE), B B → (CCC); the remainder → (C)
Method C: H → (H); E → (E); the remainder → (C) including EE and HHHH
Method D: GGGHHHHH → HHHHHHH; B, GGG → (C); H → (H); E → (E).

Test on two independent test sets

In developing a new prediction method, there is always a danger of overfitting. To guard against this, we have made rigorous use of leave-one-out cross validation, which represents the best way of assessing a prediction method. In addition, we have tested our classifier on two independent test sets.

The first dataset is formed by 23 proteins coming from CASP3 (http://PredictionCenter.llnl.gov/). These are the proteins classified by the organizers in 1999 as protein with no homologous sequences of known. We emphasize the fact that this set has not been included in the training set in any manner and constitutes a truly new set of proteins for Prof.

The dataset from CASP3 consists of 3,484 residues: 1,093 in α-helix conformation, 851 in β-strand, and 1,540 in coil. It represents a complete independent set of proteins in which our classifier shows an accuracy per residue of 76.0% and an accuracy per protein of 76.8% with a standard deviation of 10.5%; the Sov is 75.1% with a standard deviation of 16.1%. The accuracy per residue for the α-helix state is 71.3%, 75.3% for the β-strand state, and 79.5% for the coil state. This result is in good agreement with our estimated accuracy and Sov. We take this result on the CASP3 dataset only as a supplementary argument supporting our results with leave-one-out cross validation.

The second dataset considered was generated by James Cuff and Geoff Barton at the European Bioinformatics Institute (EBI) using the same procedure as the training data but on an updated release of the Protein Data Bank (PDB). The dataset consists of all nonhomologous domains added to PDB since formation of the training set in 1996. This dataset consists of 405 domains. There are 81,911 residues: 28,277 in α-helix conformation, 18,591 in β-strand, and 35,043 in coil. This dataset is large enough to identify differences between prediction methods, but it is possible that a few domains were used to train one or more of the prediction methods. On this dataset, our classifier has an estimate accuracy per residue of 77.2% and an accuracy per protein of 77.1% with a standard deviation of 8.7%. The Sov is 75.1% with a standard deviation of 13.9%. In our current available implementation of Prof, the accuracy per residue for the α-helix state is 73.4%, while the accuracy for β-strand state is 75.3%. The accuracy per residue for the coil state achieves 81.2%. This constitutes a supplementary argument supporting our estimation of the performances of Prof.

Conclusion

For the protein secondary structure prediction, we have reassessed, rigorously and completely, various GOR methods and simple three-layered neural networks with and without the use of multiple alignments. We have shown how it is possible to improve secondary structure prediction by exploiting the production of uncorrelated errors from different kinds of predictors. Using this insight, we have designed a cascaded multiple classifier for prediction that takes advantage of these various methods. The accuracy per residue of this method is 77%. This accuracy has been also reassessed using three different state reductions. However, to achieve such a high accuracy, we have had to use a combination of complicated nonlinear statistical methods. This has reduced the insight into the folding process provided by the method. Nevertheless, we have demonstrated that it is possible to design classifiers with both high global accuracy and high accuracy on β-strands using only sequence information with a local window. This suggests that the importance of long-range interactions for this class was previously overestimated. We consider that our algorithm represents an improvement in the field of secondary structure prediction.

Materials and methods

Data

We use a set of 496 nonhomologous domains. (The database can be freely obtained by academics upon request to Geoffrey J. Barton (http://barton.ebi.ac.uk).) This dataset is based on the one developed by Cuff and Barton (1999), and it is almost a proper superset of a training set of 126 domains used to originally train PHD (Rost & Sander, 1993) and DSC (King & Sternberg, 1996). The definition of homology used is now stricter than used to train PHD and DSC. Cuff and Barton (1999) did not use the percentage of identity to derive their nonredundant database; rather they used a more rigorous method consisting on the computation of the similarity score *SD* (Feng et al., 1985; Barton & Sternberg, 1987):

$$SD = \frac{V - \langle x \rangle}{\sigma} \quad (1)$$

where V is the score for the alignment of two sequences A and B by a standard dynamic programming algorithm (Needleman & Wunsch, 1970). The order of amino acid in both sequences A and B is randomized and realigned. This is re-performed n times (n is typically equal to 100). The average score $\langle x \rangle$ as well as the root-mean-square σ are computed. According to the authors (Cuff & Barton, 1999), there is no pair of domain proteins in the database with an *SD* score $\geq$5. This represents a much more stringent definition of similarity than simply taking all of the proteins that share less than 25% of identity to each other. Furthermore, the 5 *SD* cutoff used to derive the database is more stringent than scores used in all previous studies of secondary structure prediction (Cuff & Barton, 1999).

The database contains 82,847 residues: there are 28,678 in helix conformation, 17,741 in beta-strand, and 36,428 in coil. Secondary structure was assigned using the DSSP program (Kabsch & Sander, 1983). Cuff and Barton (1999) have shown that the exact mapping of DSSP output to three states secondary structure can have a significant effect on the resulting estimated accuracy. Therefore, we have used the following conservative mapping to train the method: H, I, G states from DSSP are translated as α-helix (H), E is translated as β-strands (E), and the remainder is translated as coil (C).

Generating the multiple sequence alignments

We used the BLAST program with the default parameters (Altschul et al., 1990) with the BLOSUM62 matrix (Henikoff & Henikoff, 1992) to search for homologous sequences on the NR protein database (release of April 17, 1998) containing 299,576 sequences. The BLAST output was then filtered by the program TRIMMER (obtained from M. Saqi). This is an implementation of the Needleman and Wunsch (1970) algorithm and permits the performance of a global alignment between the target sequence and the homologous sequences found by BLAST. We select all the sequences sharing between 25 and 97% of homology and which sizes lie between the thresholds of 70 and 150% of the size of the target sequence.

The similar protein sequences are then aligned using the program CLUSTALW (version 1.7) with default parameters (Thompson et al., 1994). This conservative procedure is the one that is used currently on the DSC server (http://www.icnet.uk/bmm/dsc/dsc_form_align.html) and is very close to the strategy used by PHD (http://www.embl-heidelberg.de/predictprotein/ppDoPredDef.html).

During the CASP3 meeting (1998), it was shown that improvement could be achieve using PSI-BLAST (Altschul et al., 1997) derived sequence profiles (Jones, 1998) (http://globin.bio.warwick.ac.uk/psipred_info.html). We explored this new idea by making use of the profile matrices generated automatically by PSI-BLAST. The PSI-BLAST iterative procedure is more sensitive than the corresponding BLAST program in the sense that it can detect weaker but truly related sequences with respect to the query.

Prediction measurements

We have used several measures of prediction success. We computed the standard per residue Q_3 accuracy that is defined as the number of residues correctly predicted divided by the total number of residues. This measures the expected accuracy of an unknown residue. We also measured the Q_3 per protein. The prediction accuracies for the three types of secondary structure (H, E, C) were computed. We define Q_H as the total number of α-helix correctly predicted divided by the total number of α-helix. We define in the same manner Q_E for β-strands and Q_C for coils. We computed the Mathews' correlation coefficient as well for each state (Mathews, 1975):

$$C_i = \frac{p_i n_i - u_i o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}} \quad \text{with } i \in (\text{H,E,C}) \tag{2}$$

where

p_i = number of residues correctly positively predicted to structure i,

n_i = number of residues correctly negatively predicted,

u_i = number of false negatives, and

o_i = number of false positives.

More recently, it has been proposed (Rost et al., 1994; Zemla et al., 1999) to use the Sov or segment overlap measure as a complement to the standard per residue accuracy. The aim of Sov is to assess "in a more realistic" manner the quality of a prediction. This is done by taking into account the type and position of secondary structure segment, the natural variation of segment boundaries among families of homologous proteins, and the ambiguity at the end of each segment. The quality of match of each segment pair is taken as a ratio of the overlap of the two segments ($minov(S_{obs}, S_{pred})$) and the total extent of that pair ($maxov(S_{obs}, S_{pred})$). The definition allows this ratio to be improved by extending the overlap by the value $\delta(S_{obs}, S_{pred})$. In the following formula, $S(i)$ denotes a pair of overlapping segments (S_{obs}, S_{pred}) in conformation $i \in$ (H, E, C), $S'(i)$ denotes the set of all segment S_{obs} for which there is no overlapping segment S_{pred} in state i (for further details see Zemla et al., 1999). To make these computations, we used the program SOV written by A. Zemla and freely available from the web site: (http://PredictionCenter.llnl.gov/local/ss_eval/sspred_evaluation.html).

$$Sov(i) = \frac{1}{N(i)} \sum_{S(i)} \left[\frac{minov(S_{obs}, S_{pred}) + \delta(S_{obs}, S_{pred})}{maxov(S_{obs}, S_{pred})} \times len(S_{obs}) \right]$$

$$N(i) = \sum_{S(i)} len(S_{obs}) + s \sum_{S'(i)} len(S_{obs})$$

$$\delta(S_{obs}, S_{pred}) = min(maxov(S_{obs}, S_{pred}) - minov(S_{obs}, S_{pred}); minov(S_{obs}, S_{pred}); int(len(S_{obs})/2); int(len(S_{pred})/2)). \tag{3}$$

The measures of success were estimated using *a leave-one-out cross-validation procedure* (full Jack-knife), which is less biased than a simple n-fold cross validation. As we are using a cascaded classifier, each stage was carefully tested by Jack-knife to avoid any overfitting. This means that when assessing the prediction for protein X belonging to a set S, all the classifiers of the cascade learn on the subset S without X.

Furthermore, we used two different test sets to assess our classifier, the first one comes from the CASP3 competition and contains 23 proteins. This dataset consists of 3,484 residues. The second one was generated by J. Cuff and G. Barton at the European Bioinformatics Institute (EBI) using the same procedure as the training data but on an updated release of the PDB. This dataset consists of all nonhomologous domains added to the PDB since the formation of the training set in 1996. This dataset consists of 405 domains.

Predicting secondary structure using GOR methods and fundamentals

All GOR methods (Garnier et al., 1978, 1996; Gibrat et al., 1987) are based on the idea of treating the primary sequence R and the sequence of secondary structure S as two messages related by a translation process. This translation process is examined using information theory (Shannon & Weaver, 1949) and simple Bayesian statistics. By definition, the information function can be written as follows:

$$I(S_j; R_j) = \ln\left(\frac{P(S_j/R_j)}{P(S_j)}\right) \tag{4}$$

where

ln = natural logarithm,

S_j = one of the three conformations or classes (H, E, C),

R_j = one of the 20 amino acids at position j,

$P(S_j/R_j)$ = conditional probability for observing a conformation S_j having a residue R_j,

$P(S_j)$ = prior probability of having a conformation S_j.

All these quantities are directly computable from the database.

Applying the Bayes rule and the definition of a probability, it follows that

$$P(S_j/R_j) = \frac{f(S_j,R_j)}{f(R_j)} \quad \text{and} \quad P(S_j) = \frac{f(S_j)}{N} \tag{5}$$

where f are frequencies and N is the total number of residues in the database.

In theory, the conformation of any residue should depend on the whole sequence. In practice, the authors of GOR (Robson & Pain, 1971; Garnier et al., 1978, 1996; Gibrat et al., 1987) take into account only the local sequence around the residue of interest; namely, they used a window of 17 residues. This means that to predict the residue R_j they use all the residues from R_{j-8} to R_{j+8}; beyond these residues the information decreases (Robson & Suzuki, 1976). This window is moved over the whole sequence. In fact, they compute for each of the three states (H, E, C) the following information difference, which has to be interpreted as the discriminant function between S_j and $\overline{S}_j$:

$$I(\Delta S_j; R_{j-8},\ldots,R_{j+8}) = I(S_j; R_{j-8},\ldots,R_{j+8}) - I(\overline{S}_j; R_{j-8},\ldots,R_{j+8}) \tag{6}$$

where $\overline{S}_j$ is the complement of state S_j; for example, if S_j is C then $\overline{S}_j$ is (H and E).

In GOR I (Garnier et al., 1978), the following approximation is used for the computation of the information difference:

$$I(\Delta S_j; R_{j-8},\ldots,R_{j+8}) \approx \sum_{m=j-8}^{m=j+8} I(\Delta S_j, R_{j+m}) = \sum_{m=j-8}^{m=j+8} \left(\ln\left(\frac{f(S_j,R_{j+m})}{f(\overline{S}_j,R_{j+m})}\right) + \ln\left(\frac{f(S_j)}{f(\overline{S}_j)}\right)\right) \tag{7}$$

where $f(S_j,R_{j+m})$ is the frequency of conformation S at position j when there is a residue R at position $j+m$. In this approximation, only the so-called directional information (information a residue carries about another residue's secondary structure that does not depend on the other residue's type) is taken into account.

In GOR III (Gibrat et al., 1987), another approximation is used for the computation of this function:

$$I(\Delta S_j; R_{j-8},\ldots,R_{j+8}) \approx \sum_{m=j-8}^{m=j+8} I(\Delta S_j, R_{j+m}/R_j) = \sum_{m=j-8}^{m=j+8} \left(\ln\left(\frac{f(S_j,R_{j+m},R_j)}{f(\overline{S}_j,R_{j+m},R_j)}\right) + \ln\left(\frac{f(S_j,R_j)}{f(\overline{S}_j,R_j)}\right)\right). \tag{8}$$

All the information measures were estimated directly from frequencies, since the sample size is large enough to preclude the need for a Bayesian estimation method (as initially recommended) (Robson & Suzuki, 1976). $f(S_j,R_{j+m},R_j)$ is the frequency of conformation S at position j when there is a residue R at position j and R' at $j+m$. In this approximation, the pair information is taken into account (information a residue carries about residue's secondary structure that does depend on the other residue's type).

For each residue in the protein, three functions I are computed, one for each of the three states (H, E, C).

There are two ways to predict the structure of a residue: predict the conformation having the highest difference information function or compute the probability that the residue is in a state S_i = (H, E, C) from the information value as follows:

$$p(S_i/X) = \frac{1}{1+\frac{f(\overline{S}_i)}{f(S_i)}\,e^{-(I(\Delta S_i,X))}} \quad \text{with } X = (R_{i-8},\ldots,R_{i+8}) \text{ and } S_i \in (\text{H, E, C}). \tag{9}$$

We emphasize that these two ways of assigning the secondary structure result in two different classifiers, because in one case we do not take into account the prior probability that a residue has the conformation S_i, while in the second case we do. Figure 1 gives an example of this.

In GOR IV (Garnier et al., 1996), the authors use yet another approximation to take into account all the possible pairs formed by each residue in the window. The following approximation is used:

$$\ln\left(\frac{P(S_j,X)}{P(\overline{S}_j,X)}\right) \approx \frac{2}{17}\sum_{\substack{m=j-8\\ n>m}}^{m=j+8} \ln\left(\frac{f(S_j,R_{j+m},R_{j+n})}{f(\overline{S}_j,R_{j+m},R_{j+n})}\right) - \frac{15}{17}\sum_{m=j-8}^{m=j+8} \ln\left(\frac{f(S_j,R_{j+m})}{f(\overline{S}_j,R_{j+m})}\right). \tag{10}$$

Here the computation of the probabilities $P(S_j,X)$ are straightforward from this equation. We have performed a reassessment of these methods in this paper to study the advantages of each method.

Linear discrimination

Fisher's linear discriminant function dates back to the 1930s. A dataset with p attributes (such as input values or some function of the input values) and q possible classes is divided by $q-1$ $(p-1)$-dimensional hyperplanes in such a way as to maximize the number of data points classified correctly (Weiss & Kulikowski, 1991). A quadratic cost function is optimized to choose the "best" hyperplanes. For two categories, the linear discriminant can be expressed as a multiple regression. For more than two categories, a linear discriminant for each class is used. Equal covariance matrices for the different categories are assumed as well as a Gaussian distribution of the variables.

$$F(x) = (m_1^T - m_2^T)V^{-1}x + \frac{m_2^T V^{-1} m_2 - m_1^T V^{-1} m_1}{2} + \ln\left(\frac{p(c_1)}{p(c_2)}\right) \tag{11}$$

where

x = vector of the attributes,

m_1, m_2 = vectors of means of the attributes for classes 1 and 2, respectively,

V^{-1} = inverse of the covariance matrix for the pooled population 1 and 2,

$p(c_1)$, $p(c_2)$ = prior probabilities for an element belonging either to class 1 or class 2, respectively,

$F(x)$ = linear discriminant function between class 1 and 2, and

Classes 1 and 2 = S_j and $\overline{S}_j$, respectively.

(The vector of attributes can typically be the set of probabilities and Information functions computed using the different GOR methods.)

Quadratic discrimination

Quadratic discriminant functions are similar to linear ones, except that the boundary can be a "hypercurve" rather than a hyperplane. No assumption of equal covariance matrices is made, which means that the algorithm should be robust for cases where the classes have different covariances (Weiss & Kulikowski, 1991).

$$F(x) = \frac{1}{2} x^T (V_2^{-1} - V_1^{-1}) x + (m_1^T V_1^{-1} - m_2^T V_2^{-1}) x + \left(\ln \frac{|V_2|^{1/2}}{|V_1|^{1/2}} + \ln \left(\frac{p(c_1)}{p(c_2)} \right) + \frac{m_2^T V_2^{-1} m_2 - m_1^T V_1^{-1} m_1}{2} \right). \quad (12)$$

This is different from the linear discrimination in that the two populations to be discriminated are not pooled: the inverse of covariance matrices V_1^{-1} and V_2^{-1} are assumed to be different. Also, a Gaussian distribution is assumed that leads to the previous quadratic form. $F(x)$ represents the quadratic discrimination function. We used our own implementation of linear discrimination and quadratic discrimination for learning.

In both cases, linear and quadratic discrimination, the probability that an element (described by the vector of attributes x) belongs to class 1 rather than class 2 is computed as

$$p(c1/x) = \frac{1}{1 + e^{(-F(x))}}. \quad (13)$$

These two standard methods of discrimination are used here for the combination of outputs from different versions of GOR or from different neural networks to see if any improvements on the global accuracy can be achieved.

Neural networks

A neural network learning system is a network of nonlinear processing units that have adjustable weights (Fig. 4). We used standard three-layered fully connected feedforward networks with the backpropagation with momentum learning rule used (Press et al., 1986) to avoid oscillation problems, which are common with the regular backpropagation algorithm when the error surface has a very narrow minimum area. The width of the gradient steps was set to 0.05 and the momentum term was 0.2 (Rost & Sander, 1993). The initial weights of the neural nets were chosen randomly in the range of [−0.01, 0.01]. The learning process consists of altering the weights of the connections between units in response to a teaching signal that provides information about the correct classification in input terms. The difference between the actual output and the desired output is minimized (the sum of squares error).

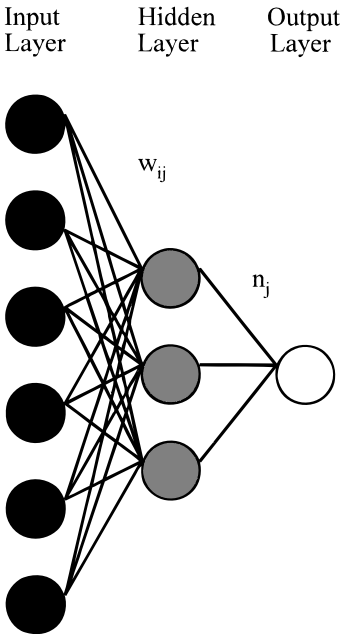

Fig. 4. Architecture of the three-layered feed-forward network used. Formal neurons are drawn as circles; weights are represented by line connecting the neurons.

For a three-layered neural network, the discriminant function $F(x)$ representing one single output can be written as follows:

$$F(x) = Sigm\left(\sum_j n_j Sigm\left(\sum_i w_{i,j} x_i - m_j \right) - m_{out} \right) \quad \text{with } Sigm(y) = \frac{1}{1 + e^{-y}} \quad (14)$$

where

x = a set of vector of attributes (input signal),

n_j = hidden-to-output weights,

$w_{i,j}$ = input-to-hidden weights,

m_j = hidden layer bias values, and

m_{out} = output neuron's bias.

The target outputs are coded as (1 0 0) for α-helices, (0 1 0) for β-strands, and (0 0 1) for coil states. All the neural networks have been trained on a set of 445 proteins, and 50 proteins are used to detect convergence. When convergence is achieved (typically less

than 40 steps of minimization), we predict the protein that has been left out. We use a simple "winner take all" strategy for the classification (Rost & Sander, 1993). It has been shown that the network outputs can be interpreted as estimated probabilities of correct prediction, and therefore they can indicate which residues are predicted with high confidence (Riis & Krogh, 1996).

To generate the neural network architecture and the learning process, we make use of the SNNS program version 4.2 freely available from the ftp site, ftp.informatik.uni-stuttgart.de (Zell et al., 1998).

We use neural-network classifiers in four different ways:

1. We learn simple sequences using the same coding procedure as Qian and Sejnowski (1988) and Rost and Sander (1993).

2. We learn sequence profiles generated from our multiple alignment with and without taking gaps into account. For each residue the frequency of occurrence is computed. Each of these 21 real numbers then represents a basic cell of the input layer (20 residues + 1 cell for the gaps).

3. Following the idea of Jones (1998), we also used the profile computed by PSI-BLAST after three iterations. This produces a different profile, first, because it detects more related sequences with weak similarity, and second, because the probabilities of occurrence of an amino acid at a specific position are computed using more powerful statistics (Tatusov et al., 1994). This method uses the prior knowledge of amino acid relationships embodied in the substitution matrix (blosum62) to generate residue pseudocount frequencies, which are averaged with the observed frequencies to estimate the probability that a residue is at specific position in the query sequence (for more details see Tatusov et al., 1994; Altschul et al., 1997). Moreover, the different sequences are weighted accordingly to the amount of information they carry.

4. We use neural networks to combine outputs from different classifiers (i.e., different versions of GOR, different networks) to design more powerful predictors. By combining a set of different classifiers in this way, it is possible to obtain an enhanced predictor, only if the individual classifiers disagree with one another (Hansen & Salamon, 1990), which means that somehow the produced errors are uncorrelated.

We use a window of 13 for both the profiles and single sequences, which means that to predict a residue we take into account the 6 previous residues and the 6 following ones, the predicted residue being at the central position of the window. The window is shifted residue by residue through the protein. However, for comparison we use as Jones (1998) a window of 17 residues (we tried also a window of 13 residues and obtained very similar results) to learn the profiles generated by PSI-BLAST.

Theoretical foundations of the combining approach

The idea of combining multiple classifiers (such as neural networks) into a single superior predictor has these recent years received great research interest (Rost & Sander, 1993; Bishop, 1995; Rosen, 1996). This constitutes probably one of the most leading advance in machine learning over these last few years. There are many existing methods for generating multiple classifiers from a training dataset and many way for combining them (Dietterich, 1997). In this work, we used different background theories and resampling techniques to generate multiple classifiers. The question of why the combination of an ensemble of classifiers should a priori perform better can be intuitively answered by the fact that uncorrelated errors made by different classifiers can be removed by correctly combining them. Another question that arise immediately is: why shouldn't we be able to find a single classifier that performs as well as an ensemble?

This question can be answered in three points (Dietterich, 1997): (1) The training data may not provide sufficient information to choose a single best classifier, and instead different "hypotheses" appear to be equally accurate. (2) The chosen learning algorithm may not be able to solve correctly the search problem that we pose. For example, neural network algorithms employ local search methods. (3) Our hypothesis space may not contain the true function. Instead, this space may contain different approximations.

Given that we have trained an ensemble of classifiers, how should we combine their individual classification decisions? The existing methodologies can be subdivided into unweighted vote, weighted vote, gating network, and combination via stacking.

1. The simplest approach is to take an unweighted vote (Clemen, 1989). One refinement on simple majority vote is when each classifier can produce class probabilities. It is then possible to average these probabilities and choose the class having the highest probability. This is the strategy adopted by the program PHD (Rost & Sander, 1993).

2. Many different weighted voting methods have been developed for ensembles (Perrone & Cooper, 1993). For classification problems, weights are usually obtained by measuring the accuracy of each individual classifier C_i and constructing weights that are proportional to those accuracies (Ali & Pazzani, 1996).

3. Another approach for combining classifiers is to learn a gating network or a gating function that takes the input features vector x and produces as output the weights to be applied to compute the weighted vote of the classifiers (Jordan & Jacobs, 1994). The output of each classifier is a probability distribution over all the possible classes while the output of the gate is a probability distribution over the classifiers.

4. A procedure called "stacking" can be used. Having different classifiers trained on a set of training examples. The goal of stacking is to learn a good combining classifier. Wolper (1992) proposed the following scheme for learning using a form of leave-one-out cross validation. The output of each classifier obtained using the leave-one-out cross-validation procedure gives a new dataset of "level 2" examples. Now we can apply some other learning algorithm to this level 2 data to obtain a more accurate classification. Breiman (1996) applied this approach to combine different forms of linear regression with good results. Stacking can be used either to combine models or to improve a single model. In this paper, we have more particularly investigated a stacking method consisting of four levels and we show how this technique can be successfully use to improve the prediction.

Acknowledgments

Ross D. King and Mohammed Ouali were funded by the BBSRC/EPSRC Bioinformatics initiative grant BIF08765. Many thanks are due to Geoffrey Barton and James Cuff for kindly providing us with the database of non-

homologous sequences. We would also like to thank the organizers of CASP3 for collecting the new crystal structures and all the crystallographers who donated structures to CASP3. We thank Mansoor Saqi for providing us with his program TRIMMER and also Mike Sternberg for helpful discussions.

References

Ali K, Pazzani MJ. 1996. Error reduction through learning multiple descriptions. *Machine Learning 24(3)*:173–202.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol 215*:403–410.

Altschul SF, Madden TL, Schäffer 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.

Anfinsen CB. 1973. Principles that govern the folding of protein chains. *Science 181*:223–230.

Avbelj F, Fele L. 1998. Role of main-chain electrostatics, hydrophobic effect and side-chain conformational entropy in determining the secondary structure of proteins. *J Mol Biol 279*:665–684.

Baldi P, Brunak S, Frasconi P, Soda G, Pollastri G. 1999. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics 15(11)*: 937–946.

Baldwin RL, Rose GD. 1999. Is protein folding hierarchic? I. Local structure and peptide folding. *Trends Biochem Sci 24*:26–33.

Barton GJ, Sternberg MJE. 1987. A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons. *J Mol Biol 198*:327–337.

Biou V, Gibrat JF, Levin JM, Robson B, Garnier J. 1988. Secondary structure prediction: Combination of three methods. *Protein Eng 2*:185–191.

Bishop C. 1995. *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press.

Breiman L. 1996. Stacked regressions. *Machine Learning 24*:49–64.

Chou PY, Fasman GD. 1974. Prediction of protein conformation. *Biochemistry 13*:222–245.

Clemen RT. 1989. Combining forcats: A review and annotated bibliography. *Int J Forecasting 5*:559–583.

Cohen FE, Abarbanel RM, Kintz ID, Fletterick RJ. 1983. Secondary structure assignment for alpha/beta proteins by a combinatorial approach. *Biochemistry 22*:4894–4904.

Cuff JA, Barton GJ. 1999. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins 4*:508–519.

Dietterich TG. 1997. Machine learning research: Four current direction. http://www.cs.wisc.edu/~shavlik/cs760.html.

Eisenberg D. 1984. Three-dimensional structure of membrane and surface proteins. *Annu Rev Biochem 53*:595–623.

Epstein CJ, Golberger RF, Anfinsen CB. 1963. The genetic control of tertiary protein structure: Studies with model systems. *Cold Spring Harbor Symp Quant Biol 28*:439–449.

Ewbank JJ, Creighton TE. 1992. Protein folding by stages. *Curr Opin Struct Biol 2*:347–349.

Feng DF, Johnson MS, Doolittle RF. 1985. Aligning amino acid sequences: Comparison of commonly used methods. *J Mol Evol 21*:112–125.

Frishman D, Argos P. 1996. Incorporation of non local interactions in protein secondary structure prediction from the amino-acid sequence. *Protein Eng 9*:133–142.

Frishman D, Argos P. 1997. Seventy-five percent accuracy in protein secondary structure prediction. *Proteins 27*:329–335.

Garnier J, Gibrat JF, Robson B. 1996. GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol 266*:541–553.

Garnier J, Osguthorpe DJ, Robson B. 1978. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J Mol Biol 120*:97–120.

Geourjon C, Deleage G. 1994. SOPM: A self optimised prediction method for protein secondary structure prediction. *Protein Eng 7*:157–164.

Gibrat JF, Garnier J, Robson B. 1987. Further developments of protein secondary structure prediction using information theory. *J Mol Biol 198*:425–443.

Hansen L, Salamon P. 1990. Neural network ensembles. *IEEE Trans Pattern Analysis and Machine Intell 12*:993–1001.

Henikoff S, Henikoff JG. 1992. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA 89*:10915–10919.

Holley HL, Karplus M. 1989. Protein secondary structure prediction with a neural network. *Proc Nat Acad Sci USA 86*:152–156.

Hubbard TJP, Sander C. 1991. The role of heat shock and chaperone proteins in protein folding: Possible molecular mechanisms. *Protein Eng 4*:711–717.

Jaynes ET. 1994. Probability theory: The logic of science. http://omega.albany.edu:8008/JaynesBook.html.

Jones DT. 1998. Prediction of protein secondary structure at 77% accuracy based on PSI-BLAST derived sequence profiles. Third meeting on the critical assessment of techniques for protein structure prediction. Asilomar Conference center Pacific Grove, California.

Jones DT. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol 292*:195–202.

Jordan MI, Jacobs RA. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6(2)*:181–214.

Kabsch W, Sander C. 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers 22*:2577–2637.

Kawabata T, Doi J. 1997. Improvement of protein secondary structure prediction using binary word encoding. *Proteins 27*:36–46.

King RD, Sternberg JE. 1996. identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Sci 5*:2298–2310.

King RD, Sternberg MJE. 1990. Machine learning approach for the prediction of protein secondary structure. *J Mol Biol 216*:441–457.

Kneller DG, Cohen FE, Langridge R. 1990. Improvements in protein secondary structure prediction by an enhanced neural network. *J Mol Biol 214*:171–182.

Krogh A, Riis SK. 1996. Prediction of beta sheets in proteins. In: Touretzky DS, Moser MC, Hasselmo ME, eds. *Advances in Neural Information Processing System 8*. MIT Press.

Levin JM. 1997. Exploring the limits of nearest neighbor secondary structure prediction. *Protein Eng 10*:771–776.

Levin JM, Pascarella S, Argos P, Garnier J. 1993. Quantification of secondary structure prediction improvement using multiple alignements. *Protein Eng 6*:849–854.

Lim VI. 1974. Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins. *J Mol Biol 88*:873–894.

Mathews BW. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysosyme. *Biochim Biophys Acta 405*:442–451.

Muggleton S, King RD, Sternberg MJE. 1992. Protein secondary structure prediction using logic. *Protein Eng 5*:647–657.

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, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. 1997. CATH—A hierarchic classification of protein domain structures. *Structure 5*:1093–1108.

Perrone MP, Cooper LN. 1993. When networks disagree: Ensemble methods for hybrid neural networks. In: Mammone RJ, ed. *Neural networks for speech and image processing*. Chapman and Hall.

Press WH, Flamery BP, Tenholsky SA, Vetterling WT. 1986. *Numerical recipes: The art of scientific computing*, Cambridge, MA: University Press.

Ptitsyn OB, Finkelstein AV. 1983. Theory of protein secondary structure and algorithm of its prediction. *Biopolymers 22*:15–25.

Qian N, Sejnowski TJ. 1988. Predicting the secondary structure of globular proteins using neural network models. *J Mol Biol 202*:865–884.

Riis SK, Krogh A. 1996. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignment. *J Comput Biol 1*:163–183.

Robson B, Pain RH. 1971. Analysis of the code relating sequence to conformation in proteins: Possible implications for the mechanism of formation of helical regions. *J Mol Biol 58*:237–259.

Robson B, Suzuki E. 1976. Conformational properties of amino acid residues in globular proteins. *J Mol Biol 107*:327–356.

Rosen B. 1996. Ensemble learning using decorrelated neural networks. *Connection Sci 8*:373–384.

Rost B. 1996. PHD: Predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol 266*:525–539.

Rost B, Sander C. 1993. Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol 232*:584–599.

Rost B, Sander C, Sneider R. 1994. Redefining the goals of protein secondary structure prediction. *J Mol Biol 235*:13–26.

Salamov AA, Solovyev VV. 1995. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J Mol Biol 247*:11–15.

Salamov AA, Solovyev VV. 1997. Protein secondary structure prediction using local alignments. *J Mol Biol 268*:31–36.

Shannon CE, Weaver W. 1949. *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.

Tatusov RL, Altschul SF, Koonin EV. 1994. Detection of conserved segments in

proteins: Iterative scanning of sequence databases alignments bloks. *Proc Natl Acad Sci USA 91*:12091–12095.

Thompson JD, Higgins DG, Gibson TJ. 1994. CLUTALW: Improving the sensitivity of progressive multiple sequence alignement through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res 22*:4673–4680.

Weiss SM, Kulikowski CA. 1991. Computer system that learn. San Mateo, CA: Morgan Kaufman.

Wolper D. 1992. Stacked generalization. *Neural Networks 5(2)*:241–260.

Yi T, Lander ES. 1993. Protein secondary structure prediction using nearest-neighbor methods. *J Mol Biol 232*:1117–1129.

Zell A, Mamier G, Vogt M, Mache N, Hübner R, Döring S, Herrmann K-U, Soyez T, Schmalzl M, Sommer T, et al. 1998. SNNS Stuttgart Network Simulator. User Manual, Version 4.2. University of Tübingen, Wilhem-Schickerd-Institute for Computer Science.

Zemla A, Venclovas C, Fidelis K, Rost B. 1999. A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins 34*:220–223.

Zhang CT, Chou KC. 1992. An optimization approach to predicting protein structural class from amino acid composition. *Protein Sci 3*:401–408.

Zimmerman K, Gibrat JF. 1998. In unison: Regularization of protein secondary structure predictions that makes use of multiple sequence alignments. *Protein Eng 10*:861–865.

Zvelebil MJJM, Barton GJ, Taylor WR, Sternberg MJE. 1987. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J Mol Biol 195*:957–961.