ISSN 0106 - 2646

NORDITA preprint

NORDITA - 95/34 S

IMPROVING PREDICTION OF PROTEIN SECONDARY STRUCTURE USING STRUCTURED NEURAL NETWORKS AND MULTIPLE SEQUENCE ALIGNMENTS

Søren Kamaric Riis.
Electronics Institute, Building 349, Technical University of Denmark, 2800 Lyngby, Denmark.

Anders Krogh.
Nordita, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark.

NORDITA · Nordisk Institut for Teoretisk Fysik

Blegdamsvej 17 DK-2100 København Ø Danmark

NORDITA preprint 95/34 S

Improving Prediction of Protein Secondary Structure using Structured Neural Networks and Multiple Sequence Alignments

Søren Kamaric Riis
Electronics Institute, Building 349
Technical University of Denmark
2800 Lyngby, Denmark
Email: riis@ei.dtu.dk

Anders Krogh
NORDITA, Blegdamsvej 17
DK-2100 Copenhagen, Denmark
Email: krogh@nordita.dk

March 31, 1995

keywords: protein secondary structure prediction, neural networks, multiple sequence alignment.

Abstract

The prediction of protein secondary structure by use of carefully structured neural networks and multiple sequence alignments have been investigated. Separate networks are used for predicting the three secondary structures α-helix, β-strand and coil. The networks are designed using a priori knowledge of amino acid properties with respect to the secondary structure and of the characteristic periodicity in α-helices. Since these single-structure networks all have less than 600 adjustable weights over-fitting is avoided. To obtain a three-state prediction of α-helix, β-strand or coil, ensembles of single-structure networks are combined with another neural network. This method gives an overall prediction accuracy of 66.3% when using seven-fold cross-validation on a database of 126 non-homologous globular proteins. Applying the method to multiple sequence alignments of homologous proteins increases the prediction accuracy significantly to 71.3% with corresponding Matthews' correlation coefficients $C_\alpha = 0.59$, $C_\beta = 0.52$ and $C_c = 0.50$. More than 72% of the residues in the database are predicted with an accuracy of 80%. It is shown that the network outputs can be interpreted as estimated probabilities of correct prediction, and therefore these numbers indicate which residues are predicted with high confidence.

1 Introduction

Prediction of protein structure from the primary sequence of amino acids is a very challenging task, and the problem has been approached from several angles. A step on the way to a prediction of the full 3D structure is to predict the *local* conformation of the polypeptide chain, which is called the secondary structure. A lot of interesting work has been done on this problem, and over the last 10 to 20 years the methods have gradually improved in accuracy. This improvement is partly due to the increased number of reliable structures from which rules can be extracted and partly due to improvement of methods.

Most often the various secondary structures are grouped into the three main categories α-helix, β-strand and "other". We use the term coil for the last category. Usually these categories

are defined on the basis of the secondary structure assignments found by the DSSP program (Kabsch & Sander, 1983). Some of the first work on secondary structure prediction was based on statistical methods in which the likelihood of each amino acid being in one of the three types of secondary structures was estimated from known protein structures. These probabilities were then averaged in some way over a small window to obtain the prediction (Chou & Fasmann, 1978; Garnier *et al.*, 1978). These methods were later extended in various ways to include correlations among amino acids in the window (Gibrat *et al.*, 1987; Biou *et al.*, 1988).

Around 1988 the first attempts were made to use neural networks to predict protein secondary structure (Qian & Sejnowski, 1988; Bohr *et al.*, 1988). The accuracy of the predictions made by Qian and Sejnowski seemed better than those obtained by previous methods, although tests based on different protein sets are hard to compare. This fact started a wave of applications of neural networks to the secondary structure prediction problem (Holley & Karplus, 1989; Kneller *et al.*, 1990; Stolorz *et al.*, 1992), sometimes in combination with other methods (Zhang *et al.*, 1992; Maclin & Shavlik, 1993). The type of neural network used in most of this work were essentially the same as the one used in the study of Qian and Sejnowski, namely a fully connected perceptron with at most one hidden layer. A very serious problem with these networks is the over-fitting caused by the huge number of free parameters (weights) to be estimated from the data. Over-fitting means that the performance of the network is poor on data that are not part of the training data, even though the performance is very good on the training data (Hertz *et al.*, 1991). In most previous work the over-fitting is dealt with by stopping the training of the network *before* the error on the training set is at a minimum, see *e.g.* (Qian & Sejnowski, 1988; Rost & Sander, 1993b) and section 2.3 of this paper. A significant exception is the work of Maclin and Shavlik (Maclin & Shavlik, 1993) in which the Chou-Fasman method (Chou & Fasmann, 1978) was built into a neural network before training. This procedure led to a network with much more structure than the fully connected ones.

The most successful application of neural networks to secondary structure prediction is probably the recent work by Rost and Sander (Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994), which has resulted in the prediction mail server called PHD (Rost *et al.*, 1994a). Rost and Sander use the same basic network architecture as Qian and Sejnowski trained on the three category secondary structure problem. Their networks have 40 hidden units and an input window of 13 amino acids, and the network is trained to predict the secondary structure of the central residue. They use two methods to overcome the problem of over-fitting. Firstly, they use early stopping, which means that training is stopped after the training error is below some threshold. Secondly, an arithmetic average is computed over predictions from several networks trained independently using different input information and training procedures. This technique of using an ensemble or committee of neural networks is known to help in suppressing noise and over-fitting (Hansen & Salamon, 1990; Krogh & Vedelsby, 1995). They also filter the predictions with a neural network which takes the predictions from the first network as input and gives a new prediction based on these. This technique was pioneered by Qian and Sejnowski, and helps in producing more realistic results by for instance suppressing α-helices or β-strands of length one. The most significant new feature in the work of Rost and Sander is the use of alignments. For each protein in the data set a set of aligned homologous proteins is found. Instead of just feeding the base sequence to the network they feed the multiple alignment in the form of a profile, *i.e.*, for each position an amino acid frequency vector is fed to the network. Using these and a few other "tricks", the performance of the network is reported to be above 71% correct secondary structure predictions using seven fold cross-validation on a database of non-homologous proteins.

One of the primary goals of the present work has been to carefully design neural network topologies particularly well suited for the task of secondary structure prediction. These networks contain much fewer free parameters than fully connected networks and thereby over-fitting is avoided. We use several methods well-known to the neural network community to further

improve performance. One of the most interesting is a learned encoding of the amino acids in a vector of three real numbers. We use the same set of protein structures as Rost and Sander (Rost & Sander, 1994) for training and evaluation of the method, which means that the results are directly comparable. Our initial goal has been to get as good predictions from single sequences as possible. This work had three stages. Firstly, individual networks were designed for prediction of the three structures. Secondly, instead of using only one network for each type of structure, an ensemble of 5 networks were used for each structure. Thirdly, these ensembles of single structure networks were combined by another neural network to obtain a three state prediction. This prediction from single sequences yields a result of 66-67% accuracy which is 3-4% better than a fully connected network on the same dataset. The method is then applied to multiple alignments as follows. For each protein in the alignment the secondary structure is predicted independently. The final prediction is then found by combining these predictions *via* the alignment as in (Zvelebil *et al.*, 1987; Russell & Barton, 1993; Levin *et al.*, 1993). By this method we obtain a result of 71.3%, which is practically identical to the result of Rost and Sander (Rost & Sander, 1994)

2 Materials and methods

2.1 Data set

When using neural networks for secondary structure prediction the choice of protein database is complicated by potential homology between proteins in the training and testing set. Homologous proteins in the database can give misleading results since neural networks in some cases can memorize the training set. Furthermore, the size of the training and testing sets can have a considerable influence on the results, because non-homologous proteins in general are very different. Using a small training set often results in bad generalization ability, while a small testing set gives a very poor estimate of the prediction accuracy. For evaluation of the method we therefore use ***seven-fold cross-validation*** on the set of 126 non-homologous globular proteins from (Rost & Sander, 1994), see Table 1. With seven-fold cross-validation approximately 1/7 of the database is left out while training, and the remaining part is used for testing. This is done cyclically seven times, and the resulting prediction is thus a mean over seven different testing sets. The division of the database into the seven subsets (set A-set G) shown in Table 1 is assumed not to have any influence on the results presented in the following sections. A more reliable estimate of the prediction accuracy could be achieved by using *Leave One Out* cross-validation where one protein is left out while training on the rest, but this would lead to very large computational demands. The proteins used all satisfy the homology-threshold defined by Sander and Schneider (Sander & Schneider, 1991), *i.e.*, no proteins in the database have more than 25% pairwise sequence identity for lengths > 80 residues. The proteins are taken from the HSSP-database version 1.0, release 25.0 (Sander & Schneider, 1991). The secondary structure assignment were done according to the DSSP algorithm (Kabsch & Sander, 1983), but the 8 types of structures were converted to three in the following way: H (α-helix), I (π-helix) and G (3_{10}-helix) were classified as helix (α), E (extended strand) as β-strand (β), and all others as coil (c).

2.2 Measures of prediction accuracy

Several different measures of prediction accuracy have been suggested in the literature. The most common measure is the overall three-state prediction percentage Q_3 defined as the ratio of correctly predicted residues to the total number of residues in the database under consideration (Qian & Sejnowski, 1988; Rost & Sander, 1993b). Since our data set contains 32% α-helix, 21% β-strand and 47% coil, a random prediction yields $Q_3^{random} = 36.3\%$ if weighted by the percentage of occurrence. For comparison the best obtainable prediction by homology methods

Set A	256b_A	2aat	8abp	6acn	1acx	8adh	3ait
	1ak3_A	2alp	9api_A	9api_B	1azu	3b5c	1bbp_A
	1bds	1bmv_1	1bmv_2				
Set B	3blm	4bp2	2cab	7cat_A	1cbh	1cc5	2ccy_A
	1cd4	1cdt_A	3cla	3cln	4cms	4cpa_I	6cpa
	6cpp	4cpv	1crn	1cse	6cts	2cyp	5cyt
	1eca						
Set C	6dfr	3ebx	5er2_E	1etu	1fc2_C	1fdl_H	1fdx
	1fkf	2fnr	2fxb	1fxi_A	4fxn	3gap_A	2gbp
	2gcr	1gd1_O	2gls_A	2gn5	1gpl_A		
Set D	4gr1	1hip	6hir	3hmg_A	3hmg_B	2hmz_A	5hvp_A
	2i1b	3icb	7icd	1il8_A	9ins_B	1l58	1lap
	5ldh	2lh4	2lhb	1lrd_3			
Set E	2ltn_A	2ltn_B	5lyz	1mcp_L	2mev_4	2or1_L	1ovo_A
	1paz	9pap	2pcy	4pfk	3pgm	2phh	1pyp
	1r09_2	2pab_A					
Set F	2mhu	1mrt	1ppt	1rbp	1rhd	4rhv_1	4rhv_3
	4rhv_4	3rnt	7rsa	2rsp_A	1s01	1sdh_A	4rxn
	4sgb_I						
Set G	1sh1	2sns	2sod_B	2stv	2tgp_I	1tgs_I	3tim_A
	6tmn_E	2tmv_P	1tnf_A	4ts1_A	1ubq	2utg_A	9wga_A
	2wrp_R	1wsy_A	1wsy_B	4xia_A	2tsc_A		

Table 1: The database of non-homologous proteins used for seven-fold cross-validation. All proteins have less than 25% pairwise similarity for lengths > 80 residues and the crystal structures are determined at a resolution better than 2.5Å rms. The data set contains 24,395 residues with 32% α-helix, 21% β-strand and 47% *coil*.

is about $Q_3^{homology} = 88\%$ (Rost *et al.*, 1994b). Q_3 describes the performance of the method averaged over all residues in the database. For a single protein the expected prediction accuracy is better described by the per chain accuracy $< Q_3^{chain} >$ given by the average of the three-state prediction accuracy over all protein chains (Rost & Sander, 1993b).

A measure of the performance on secondary structure class $i = \alpha, \beta$ or coil is the percentage Q_i of correctly predicted residues observed in class i. These measures can be very helpful in detecting over- and under-prediction of one or more types of secondary structures. Note that Q_i differs from the two-state prediction accuracy $Q_{2,i}$ (Hayward & Collins, 1992) used when evaluating single-structure networks.

A complementary measure of prediction accuracy is the Matthews' correlation coefficients (Matthews, 1975) for each of the three secondary structures; C_α, C_β and C_c. The correlation coefficients are 1.0 if the predictions are all correct and -1.0 if all the predictions are false. The advantage of the correlation coefficients is seen in case of a random or trivial prediction. A trivial prediction of helices for all residues gives $Q_\alpha = 100\%$ and $Q_3 = 32\%$, but $C_i = 0.0$. Similarly C_i is close to zero for random predictions. The Mathews' correlation coefficients are widely used and the exact definitions can be found in (Qian & Sejnowski, 1988; Rost & Sander, 1993b; Matthews, 1975).

Even though Mathews' correlation coefficients give more reliable estimates of the prediction accuracy they do not express how realistic the prediction is. Consider the two predictions in Table 2 obtained from different methods (Rost & Sander, 1993b). Even though prediction 1 gives a higher Q_3 as well as higher correlation coefficients than prediction 2, the latter is more realistic seen from a biological point of view. The first method predicts unrealistic short helices in contrast to the long helix predicted by the second method. This illustrates the need

Observed	HHHHHHHHHHCCC
Prediction 1	CHHHCHHHCHCCC
Prediction 2	CCHHHHHHHHHHC

Table 2: Predictions from two different methods

of comparing predicted and observed mean lengths L_i of secondary structure segments. In addition to the mean lengths an interesting measure is the percentage of overlapping segments of observed and predicted secondary structure used by Maclin and Shavlik (Maclin & Shavlik, 1993). The percentage of segment overlap P_i^{Ovl} tells how good the method is at locating segments of secondary structure. This is of particular interest since the 3D structure of a given protein family to some extent is determined by the approximate location of regular secondary structure segments (Rost *et al.*, 1994b). Note that a trivial prediction of helices at all positions gives P_α^{ovl} = 100% if at least one observed helix segment exists. The overlap percentages should in other words only be used in combination with some of the performance measures mentioned above.

2.3 Neural networks for secondary structure prediction

The networks used in this work are all feed-forward layered networks, trained using the back-propagation algorithm in on-line mode, see *e.g.* (Hertz *et al.*, 1991). The main difference to previous works (Qian & Sejnowski, 1988; Bohr *et al.*, 1988; Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994; Maclin & Shavlik, 1993; Zhang *et al.*, 1992; Hayward & Collins, 1992) using these types of networks will be described in this section.

In most applications of neural networks to secondary structure prediction, fully connected networks with a wast number of adjustable weights have been used. For instance, the best network found in the work of Qian and Sejnowski (Qian & Sejnowski, 1988) had more than 10,000 weights. When training a network with that many weights from the limited number of proteins available one gets into the problem of *over-fitting*. At some point during training the network begins to learn special features in the training set, *i.e.* the network begins to memorize the training set. These special features can be considered as noise or atypical examples of mappings between amino acid sequence and secondary structure. Since the noise in the training and testing sets is uncorrelated the generalization ability on the testing set deteriorates at some point during training, see Figure 1. The point at which the generalization ability deteriorates is highly dependent on the initial weights and on the dynamics of the learning rule. Hence, it is almost impossible to determine at which point the training should be stopped in order to get an optimal solution. Usually *early stopping* is used, where the training is stopped after some fixed number of iterations (Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994) or by using a *validation* set to monitor the generalization ability of the network during training (Maclin & Shavlik, 1993). When the performance on the validation set begins to deteriorate the training is stopped. However, sacrificing data for validation sets can be crucial for the performance of the model, since the available amount of data is limited. Another method is to choose the network achieving the best performance on the test set by always saving the best network during training, as was done by Qian and Sejnowski. In that case the performance on the test set can not be expected to reflect the performance on independent data. The best approach of course is to deal with the root of the problem, namely finding the proper complexity of the network.

One of the main goals of this work has been to design networks that avoid over-fitting all together. By avoiding over-fitting, the learning and generalization errors stay almost identical, and therefore training can be continued until it reaches minimum training error.

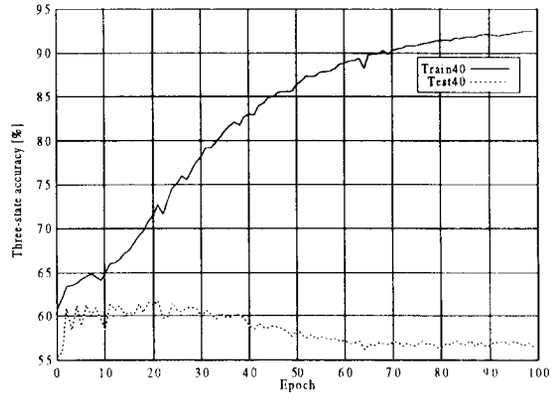

Figure 1: Three-state percentages (Q_3) for the training and testing set during training of the network with 40 hidden units used in (Qian & Sejnowski, 1988) (spacer unit omitted). The training set consists of sets B–G and for testing set A is used, see Table 1. The percentages are plotted against the number of training epochs, *i.e.* full sweeps through the training set. Because of the extreme number of weights the network develops a very poor generalization on the testing set. In less than 100 training epochs the training set is learnt almost to perfection while the performance on the testing set has dropped from the maximum value of approximately 62% to 57%. Qian and Sejnowski reported the best percentage obtained for the testing set as an estimate of the prediction accuracy.

Adaptive encoding of amino acids

As in most of the existing methods, the secondary structure of the j'th residue R_j is predicted from a window of amino acids, $R_{j-n}, ..., R_j, ..., R_{j+n}$ where $W = 2n + 1$ is the window size. These neural networks are often referred to as *sequence-structure* networks. Usually the amino acids are encoded by 21 binary numbers, such that each number corresponds to one amino acid. The last number corresponds to a space, and is used to indicate the ends of a protein. This encoding, which we will call the *orthogonal encoding*, has the advantage of not introducing any artificial correlations between the amino acids, but it is highly redundant, since 21 symbols can be encoded in 5 bits. This redundancy is one of the reasons why networks for secondary structure prediction tend to have a very large number of weights. However, according to Taylor (Taylor, 1986) the properties of the 20 amino acids with respect to the secondary structure can be expressed remarkably well by only two physical parameters: the hydrophobicity and the molecular volume. This suggests using another encoding scheme than the orthogonal one.

By a method called *weight sharing* (Le Cun *et al.*, 1989) it is possible to let the network itself choose the best encoding of the amino acids. The starting point is the above mentioned orthogonal encoding, but we omit the spacer input unit used by Qian and Sejnowski, and instead all inputs are set to zero for that part of the window where no residues are present. For each window position the 20 inputs are connected to M hidden units by $20 \times M$ weights. This set of weights (and the M thresholds) corresponding to one window position is identical to those used for all the other window positions, see Figure 2. More precisely, if the weight from input j to hidden unit i is called w_{ij}^k for the k'th window position, then $w_{ij}^k = w_{ij}^l$ for all k and l. These sets of weights are forced to stay identical during training; they always share the same values. In this way the encoding of the amino acids is the same for all positions in the window. The weights are learned by a straight-forward generalization of back-propagation in which weight

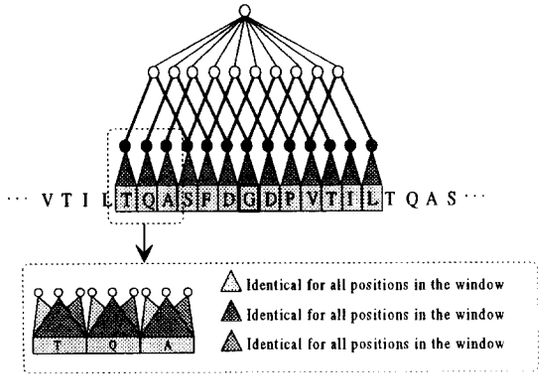

Figure 2: Network for predicting helices. The network uses the local encoding scheme and has a built-in period of 3 residues. Grey circles symbolize three hidden units and emphasized lines three weights. In the lower part of the figure shaded triangles symbolize 20 shared weights and shaded rectangles 20 input units. The single-structure network shown has a window size of 13 residues and only one output.

updates are summed for weights sharing the same value (Le Cun *et al.*, 1989). The use of weight sharing implies that the first layer only contains $21 \times M$ adjustable parameters including thresholds no matter the size of the window. In this work $M = 3$ is used and each of the 20 amino acids are thus represented by only three real numbers in the interval $[0; 1]$. This leads to a dramatic reduction of the almost 11,000 weights used in the first layer of Qian and Sejnowski's fully connected network, even if an extra hidden layer is added to the network.

The adaptive encoding scheme of the amino acids is called *local encoding*. Since the encoding is learned along with the other weights in the network it will be the 'optimal' encoding, in the sense that it yields the minimum error on the training set for that specific network and that specific task. The adaptive nature of the encoding also means that it depends on the initial weights (like the other weights in the network) and may differ between different runs of the learning algorithm.

Structured networks

It is a common assumption that a network (or any other adaptive method) with some built-in knowledge about the problem performs better than more general networks, see *e.g.* (Maclin & Shavlik, 1993). Many existing prediction methods use the same model for predicting the three types of secondary structure (helix, strand, and coil). Since the three secondary structures are very different it is possible that performance could be enhanced if separate networks are specifically designed for each of the three structures. We will now explain how prior knowledge about secondary structures can be used to design such single-structure networks.

The majority of the helices in the database used are α-helices. A residue in an α-helix is hydrogen bonded to the fourth residue above and the fourth residue below in the primary sequence, and it takes 3.6 amino acids to make a turn in an α-helix. It is likely that this periodic structure is essential for the characterization of an α-helix. These characteristics are all of local nature and can therefore easily be built into a network that predicts helices from windows of the amino acid sequence. In Figure 2 a network with local encoding (in the first hidden layer), a built-in period of 3 residues and a window size of 13 residues is shown. The second hidden layer in the network contains 10 units that are fully connected to the output layer giving a total

of 144 adjustable parameters. For comparison a standard network with no hidden units at all, orthogonal encoding, and a window length of 13 residues has 261 adjustable parameters.

In contrast to helices, β-strands and coil do not have such a locally described periodic structure. Therefore, the strand and coil networks only use the local encoding scheme, and a second hidden layer with 5-10 units fully connected to the first hidden layer as well as to the output layer. Early studies (results not shown) indicated that a window size of 15 residues was optimal for all three types of single-structure networks. Thus, a typical structured helix network contains 160 weights, while typical strand and coil networks contain about 300-530 weights.

As shown in Figure 2 the single-structure networks only have one output. If the output is larger than some decision threshold the prediction is α-helix, β-strand or coil depending on the type of structure under consideration. For an input/output interval of $[0;1]$ a decision threshold of 0.5 was found to be optimal.

The performance of the constrained single-structure networks are compared with the predictions obtained from perceptrons with no hidden units having window lengths of 13 amino acids. The single-structure networks are all trained balanced *i.e.*, for each positive example (helix) a negative example (non-helix) is chosen at random from the training set. In this way the same number of positive and negative examples are used in the training. According to Hayward and Collins (Hayward & Collins, 1992) balanced training gives only minor changes in the percentage of correctly classified residues ($Q_{2,i}$), but slightly better correlation coefficients. This is in good agreement with our own experiments (data not shown).

Filtering the predictions

As described earlier, some predictions may be very unrealistic from a biological point of view. For instance, prediction 1 in Table 2 has an α-helix of length one in the end. To obtain more realistic predictions a *structure-structure* network can be applied to the prediction from the previously described *sequence-structure* network. In the work of Qian and Sejnowski a window of 13 secondary structure predictions is used as input to a fully connected structure-structure network with 40 hidden units. Thus, this network has 3×13 inputs and 3 outputs, and the predicted secondary structure for the central amino acid is chosen as the largest of the three outputs. In this way the prediction becomes dependent on the surrounding structures. The structure-structure network is often called a *filter* network, because it is used to filter out bad predictions, although it can in principle do more than that. According to (Qian & Sejnowski, 1988; Rost & Sander, 1994) the filter network improves the three-state accuracy significantly and makes the prediction more realistic in terms of predicted mean lengths of secondary structure segments. A filter network similar to the one used by Qian and Sejnowski can also be applied when combining the predictions from the three single-structure networks. Notice that this network actually increases the size of the window used for the prediction of an amino acid. Since the first network uses a window size of 13 (or 15) amino acids, the second network receives information based on a total window of 25 (or 29) amino acids.

For the single-structure predictions a filter network can be applied in the same manner as for the three-state predictions. In this work each of the single-structure predictions are filtered with fully connected networks having 10 hidden units and a window size of 15 single-structure predictions. As will be shown, the filtration of the single-structure predictions before combining to a three-state prediction can be omitted without loss of accuracy.

Using softmax for combining single-structure predictions

Usually the neural network outputs three values, one for each of the three structures. This type of network does not necessarily choose one of the three structures. For instance it can (and sometimes do) classify one input pattern as all three types of structure, *i.e.*, it gives large outputs on all three output units. In practice of course, the input is classified as the structure giving

the largest output, but conceptually this type of classification is more suited for independent classes. It may be beneficial to build in the constraint that a given input belongs to only one of the three structures. This can be done by a method called *Softmax* (Bridle, 1990), which ensures that the three outputs always sum to one (for secondary structure prediction the same idea was used in (Stolorz *et al.*, 1992)). Hence, the outputs can be interpreted as the conditional probabilities that a given input belongs to each of the three classes. Simulation studies done by Richard and Lippmann (Richard & Lippmann, 1991) show that neural network classifiers provide good estimates of Bayesian a posterior probabilities (conditional probabilities). In the Softmax method the usual sigmoidal activation function

$$O_i = g(h_i) = \frac{1}{1 + e^{-h_i}} \tag{1}$$

in the output layer is replaced by the normalizing function

$$O_i = \frac{e^{h_i}}{\sum_j e^{h_j}}, \tag{2}$$

where O_i is the i'th output and the sum in the denominator extends over all outputs. In these formulas $h_i = \sum_j w_{ij} x_j$ is the net input to output unit i, w_{ij} is the weight connecting output unit i to hidden unit j, and x_j is the output of hidden unit j.

Here a log-likelihood cost function is used instead of the usual squared error cost function. If ζ_i is the target output of the i'th output unit, then the contribution to the cost function from one training example can be written as

$$E(\mathbf{w}) = \sum_i \zeta_i \log(\frac{\zeta_i}{O_i}) \tag{3}$$

whereas the usual cost function is $\sum_i (\zeta_i - O_i)^2$. The weight update formulas are easily calculated and turn out to be identical to the ones used in backpropagation if the entropic cost function defined in (Hertz *et al.*, 1991) is applied.

To combine and filter the single-structure predictions a single neural network is used. This network takes the outputs from the three single structure networks as input and uses the softmax function (2) on the three output classes. The combining network takes a window of 15 consecutive predictions of helix, strand and coil as input, and the input layer is fully connected to the output layer via 10 hidden units. When using Softmax the predictions can be interpreted as estimated probabilities of correct prediction. Results on how well the outputs match probabilities will be shown.

Ensembles of single-structure networks

The solution found by a neural network after training depends on the initial weights and the sequence of training examples. Thus, training two identical networks often results in two different solutions, *i.e.*, two different local minima in the objective function are found. Since the solutions are not completely correlated the combination of two or more different networks often improves the overall accuracy (Rost & Sander, 1994; Hansen & Salamon, 1990). For complex classification tasks the use of ensembles can be thought of as a way of averaging out statistical fluctuations. Furthermore, the combination of two or more different solutions can in some cases contribute valuable information. This is especially true if the ensemble members disagree as discussed in (Krogh & Vedelsby, 1995). One obvious way to make the ensemble members disagree is to use different networks and/or training methods. In this work ensembles of 5 different single-structure networks (for each type of secondary structure) are used. The networks all use the local encoding scheme and the differences are introduced by using various periods in the α-network and by using different numbers of hidden units.

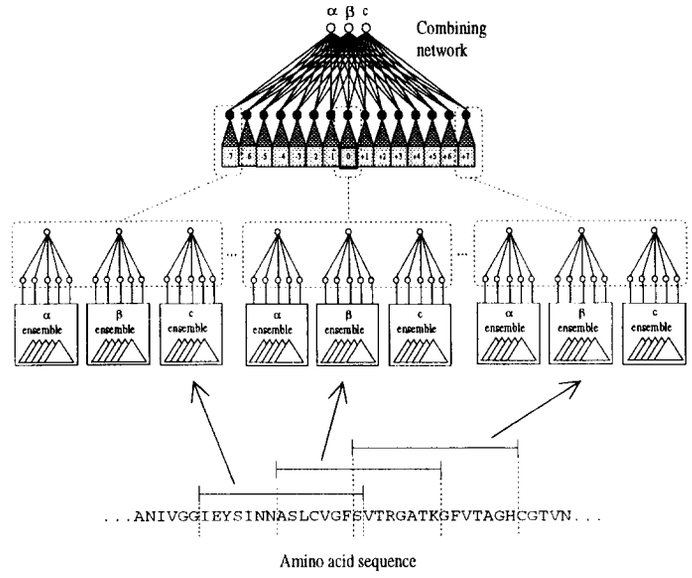

Figure 3: The ensemble method for combining and filtering ensembles of single-structure networks. The combining network (top of figure) takes a window of $3 \times 5 \times 15$ predictions from the ensembles of single-structure networks (3 structures, 5 networks for each structure, and a window length of 15). In the combining network the ensembles for each of the three structures are weighted separately by position specific weights for each window position.

The usual way to combine the ensemble predictions is to sum the predictions using uniform (equal) weighting of the ensemble members. Instead, we have chosen to use a neural network for the combination. Rather than first training a filter network for each of the individual networks in the ensemble, our approach is to combine and filter the whole ensemble with only one network. This network takes a window of predictions from all the single-structure networks in the ensemble and then decides one output for the central residue. However, using a fully connected network results in considerable over-fitting since a window length of 15 residues equals $15 \times 3 \times N_E$ inputs for ensembles of N_E networks. Here $N_E = 5$ is used leading to a total of 225 inputs. One way to reduce the number of weights is by weighting each of the three single-structure ensembles separately for all positions in the window. In this way segments of 5 inputs corresponding to, *e.g.*, 5 helix network outputs are connected to one hidden unit in the combining network, see Figure 3. Thus, for a given position in the input window each of the three ensembles are averaged using position specific weights. This constraint gives a total of $3 \times 15 = 45$ hidden units that are fully connected to the output layer consisting of three units. The prediction for the central residue is chosen as the largest of the three outputs that are normalized with softmax. The combining network is trained unbalanced.

2.4 Using multiple alignments of homologous proteins

Multiple alignments of homologous proteins contain more information about secondary structures than single sequences alone, because the secondary structure is considerably better con

served than the amino acid sequence. The use of multiple alignments can give significant improvements in the secondary structure prediction (Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994; Zvelebil *et al.*, 1987) especially if weakly related proteins are included in the alignments. The latter only holds if the alignment of the weakly related proteins is good, *i.e.*, resembles the structural alignment obtained by superposition of protein backbones (Levin *et al.*, 1993).

Recently Rost and Sander have had significant success by using *sequence profiles*[1] from such alignments as input to the neural network instead of a single sequence (Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994). When using profiles instead of single sequences, correlations between amino acids in the window will not be available to the network. Although this may not degrade performance in practice, we have chosen another approach, which conserves these correlations. It is the approach also taken in (Zvelebil *et al.*, 1987; Russell & Barton, 1993; Levin *et al.*, 1993), where the predictions are made from the single sequences and then combined afterwards using the alignment. This method also has the advantage of being able to use any secondary structure prediction method (based on single sequences) and any alignment method.

To the protein for which a secondary structure prediction is wanted (called the *base protein*), a set of homologous proteins are found. This set of proteins including the base protein is used for the secondary structure prediction in the following way.

1. The secondary structure for each of the homologous proteins in the set is predicted independently from the amino acid sequences. Any prediction method based on single sequences can be used at this stage, but we use the ensemble method described above.

2. The protein sequences in the set are aligned by some multiple sequence alignment method and each protein is assigned a weight (see below).

3. For each column in the alignment a consensus prediction is found from the predictions corresponding to each of the amino acids in the column (see below).

For each column the consensus is obtained either by *weighted average*, or by *weighted majority*. The weighted average is calculated by first multiplying the α-helix predictions by the weights of the proteins, and then summing the weighted helix predictions column-wise. Similarly the weighted sums of β-strand and coil predictions are calculated. Note that insertions in the alignment do not contribute to the column sums. The largest of the three column sums then determine the predicted secondary structure for this column. In the weighted majority, the prediction for each amino acid is chosen by the largest of the three outputs. Then the total sum of α-helix predictions is calculated by column-wise summing of the weights for those proteins where an α-helix is predicted. Similarly the total sums of β-strand and coil predictions are found. The secondary structure obtaining the largest column sum is chosen as the predicted one for this column. In this way, weighted majority becomes dependent on the estimated weights for each of the proteins in the alignment.

Weighting the aligned proteins

If an alignment contains many very similar proteins and a few that differ significantly from the majority, then the minority will have almost no influence on the prediction. Therefore it is often a good idea to try to weight the proteins differently. Several weighting schemes have been suggested in recent years, see *e.g.* (Altschul *et al.*, 1989; Vingron & Argos, 1989; Sibbald & Argos, 1990; Gerstein *et al.*, 1994; Henikoff & Henikoff, 1994). Here we will use a newly developed one based on the maximum entropy principle (Krogh & Mitchison, 1995).

[1]Sequence profiles are the frequencies of the 20 amino acids in each column of the alignment

For an alignment of N proteins, the entropic weights are found by maximizing the entropy of the alignment defined by (Krogh & Mitchison, 1995):

$$S(w_1, \ldots, w_N) = \sum_{j=1}^{M} e_j = -\sum_{j=1}^{M} \sum_{x} p_j(x) \log p_j(x) \tag{4}$$

where the sum is extended over all alignment columns $j = 1, \ldots, M$ and over the 20 different amino acids x. $p_j(x)$ is the *weighted* amino acid frequencies for column j, *i.e.*, $p_j(x)$ is a function of the weights assigned to the aligned proteins, see (Krogh & Mitchison, 1995). The entropy is a concave function in the weights, and it is therefore easy to maximize. We have used simple gradient ascent in this work, although more efficient techniques are available. The problem with entropic weighting and any other alignment based weighting scheme is that erroneously aligned proteins can be assigned very high weights, which obviously is wrong. Since aligning weakly related proteins often results in erroneous alignments (Vingron & Argos, 1989; Levin *et al.*, 1993) the weighting schemes should be used with precaution. For this reason we have also tested a combination of the uniform and the entropic weights. Thus, for protein i in the alignment the weight is given by:

$$w_i = \frac{\epsilon}{N} + (1 - \epsilon) w_i^{entropic} \tag{5}$$

where $\epsilon = 0.5$ is used in this work.

To improve the alignment prediction a one-hidden-layer filter network is applied to the consensus prediction. This network takes a window of 15 consecutive alignment predictions as input. In addition the column entropy e_j and the weighted number of insertions and deletions (InDels) for each column are used as input. Thus, the filter network has a total of $15 \times (3 + 3) = 90$ inputs. The entropy of each alignment column indicates how well the current position is conserved. That is, if the column entropy is close to zero, then the variation of the amino acids in this column is small, *i.e.* this position is well conserved in the protein family. On the other hand, if the column entropy is large, then the variation of amino acids is large, *i.e.* this position is very poor conserved. Since regular secondary structure segments are more conserved than coil segments, a large variation of amino acids is often observed in coil regions (Rost *et al.*, 1994b). Thus, a large column entropy often corresponds to a coil region, and a small entropy to an α-helix or a β-strand region. The weighted number of InDels is the number of insertions and deletions on the considered alignment position weighted by equation (5). InDels most often occur in coil regions. To avoid over-fitting the number of hidden units is 5.

The alignments used to test this method are taken from the HSSP-database version 1.0, release 25.0 (Sander & Schneider, 1991). For each of the 126 non-homologous proteins the corresponding HSSP file is found. These files consists of homologous proteins that have at least 30% sequence identity for alignment lengths > 80 residues, and larger for shorter proteins (Sander & Schneider, 1991). There are two minor problems using the HSSP files for secondary structure predictions. For creating the alignments in the HSSP files, knowledge about the secondary structure of the base protein is used, since no insertions or deletions in regular secondary structure segments are allowed. Furthermore, there might be homologies between proteins in different HSSP files, although the base proteins do not have significant homologies, and this might give homology between the test and training sets. In our experience these points have insignificant influence on the results, and using the HSSP files gives us the advantage of being able to directly compare our results with those of (Rost & Sander, 1994).

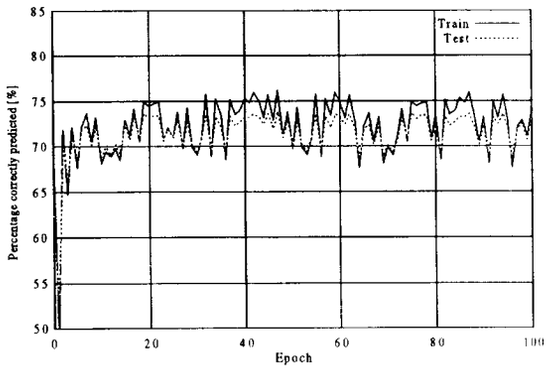

Figure 4: Percentage ($Q_{2,\alpha}$) of residues predicted correctly by the α-network as a function of the number of training epochs (full sweeps through the training set: set B-G). The solid curve shows the percentage of learned residues in the training set and the dotted curve the prediction accuracy on the testing set (set A).

3 Results

3.1 Two-state predictions by single-structure networks

The result of training the structured α-network on set B–G and using set A as testing set is shown in Figure 4. This figure shows two interesting features: 1) Over-fitting is gone, *i.e.* the accuracy on the training and testing sets are approximately equal; 2) the training and testing percentages oscillate in phase. The first observation means that this network gives reliable estimates of prediction accuracy on new proteins not in the database used for developing the method. The observed fluctuations is mostly due to the use of balanced training where a different set of negative examples (non-helix) are used in each training epoch. Since the in-phase oscillations are observed for all of our networks, the final network weights are chosen as follows. The network is trained for 100 training epochs, and in each epoch the training error is measured. If the training error is lower than in all previous epochs the corresponding weights are saved. In this way, the set of weights corresponding to the smallest training error seen during all 100 epochs is found.

For the single structure predictions a fully connected network with hidden units only performs as well as a one-layer network if the training is stopped at the right time, see (Hayward & Collins, 1992). Therefore we use a one-layer network as a reference model. In Table 3 the results obtained with the single-structure networks are summarized. From the table it is seen that the structured networks predict the three secondary structures better than the reference models. Furthermore, the structured helix network learns the training set better than the reference helix network despite the fact that the latter contains more weights (reference α-network: 261 weights, structured α-network: 160 weights). This shows that the learned representation of the amino acids is considerably better than the orthogonal representation.

When comparing two-state predictions for different testing sets a considerable variation in performance is seen. For testing set B the helix network classifies $Q_{2,\alpha} = 72.5\%$ of the residues correctly while this number is $Q_{2,\alpha} = 77.7\%$ for testing set F. The variation between the seven different testing sets is observed for all of the single-structure networks and is partly due to the different distributions of the three secondary structures, and partly due to the fact that non-homologous proteins in general are very different. This emphasizes the importance of using

	$Q_{2,i}^{Train}$ (%)	$Q_{2,i}^{Test}$ (%)	C_i^{Train}	C_i^{Test}
α-network:				
Reference	74.10	72.54	0.42	0.37
Structured	75.59	74.98	0.42	0.39
Filter	76.31	76.46	0.42	0.40
β-network:				
Reference	75.08	73.84	0.39	0.36
Structured	78.10	76.48	0.41	0.37
Filter	81.52	81.34	0.41	0.41
coil-network:				
Reference	71.91	70.78	0.43	0.41
Structured	72.09	71.33	0.44	0.42

Table 3: Two-state predictions of α-helix, β-strand and coil found by seven-fold cross-validation. The reference networks are perceptrons with window lengths $W = 13$. The structured networks all use the local encoding scheme and the α-network has a built in period of 3 residues. The fully connected filter network takes a window of 15 predictions from the structured network as input and has 10 hidden units. The filter is trained unbalanced.

cross-validation when estimating prediction accuracies.

To improve the two-state prediction a filter network is applied. This network takes a window of 15 consecutive predictions from the sequence-structure network as input. The filter network is fully connected and contains 10 hidden units. As shown in Table 3 the filter improves prediction approximately 1.5% for helices and almost 5% for strands. Filtering the coil predictions gives only about 0.5% improvement, probably because coil is an irregular structure, *i.e.*, it is not so much dependent on the surrounding structures.

3.2 Combining single-structure networks

To obtain a three-state prediction the single-structure networks are combined with a filter network. The filter network takes a window of 15 consecutive secondary structure predictions as input and has 10 hidden units. In Table 4 is shown the results achieved when using the non-filtered and the filtered single-structure predictions as input. From this table it is seen that filtering the single-structure predictions before the combining network does not improve performance. This is because the combining network in itself acts like a filter network. For comparison, the performance of a network identical to Qian and Sejnowski's with 40 hidden units is also shown in Table 4. The performance of this network is evaluated on the same set of non-homologous proteins by seven-fold cross-validation, and it is seen that the fully connected network only obtains $Q_3 = 63.2\%$ compared to $Q_3 = 65.4\%$ obtained by combining the unfiltered single-structure predictions. Note that the results obtained with the Qian and Sejnowski model is found by using the best performance on each of the seven testing sets, which over-estimates the performance. For the combining network the above defined stop criterion is used.

The effect of the local encoding scheme is illustrated by a three-state network, which uses the adaptive encoding of amino acids in the first layer and 5 hidden units in the second layer. This network has a window size of 15 residues leading to a total of only 311 adjustable weights compared to approximately 11,000 weights in Qian and Sejnowski's network. Despite this difference the local encoding network gives about the same Q_3 and better correlation coefficients, indicating that the amino acids are well described by only three real parameters, and that the fully connected networks are highly over-parametrized. Results after filtration are shown in Table 4. The filter network has an input window of 15 and a hidden layer consisting of 10 units.

When combining ensembles the approach described in section 2. (Figure 3) is used. Ensem

	Q_3 (%)	C_α	C_β	C_c
Combined single-structure nets:				
Filtered input	64.46	0.45	0.39	0.42
Unfiltered input	65.39	0.46	0.41	0.43
Ensembles	66.27	0.48	0.41	0.44
Alignments:				
Uniform (Not filt.)	68.81	0.55	0.46	0.48
Entropic (Not filt.)	68.68	0.54	0.46	0.47
Entropic+Uniform (Not filt.)	69.20	0.55	0.46	0.48
Entropic+Uniform (Filtered)	71.32	0.59	0.52	0.50
Reference models:				
Qian and Sejnowski network	63.16	0.40	0.35	0.41
Local encoding (Not filt.)	63.10	0.42	0.36	0.41
Local encoding (Filtered)	64.20	0.44	0.37	0.41

Table 4: Cross-validated three-state predictions obtained by various methods. *Ensembles* refers to the combination of ensembles of 5 single-structure networks with a constrained network and *Alignments* refers to using multiple alignments of homologous sequences in combination with ensembles. The alignment prediction is obtained by weighted average and different weighting schemes are shown. The effect of filtering the alignment prediction is also shown. For comparison is shown the performance of a fully connected network similar to the one used by Qian and Sejnowski with 40 hidden units (input spacer units are omitted). Note that the performance for the fully connected network is given by the best performance on the testing set during training and that the previously defined stop criterion is used for all other networks. Also shown is a three-state prediction network with local encoding in the first layer and 5 hidden units in the second layer fully connected to the output layer. Results with and without filtration are shown.

bles give an improvement of approximately 0.9% in the overall three-state prediction accuracy mostly due to a better helix prediction (higher C_α), see Table 4. This is less than the improvement of more than 2% reported by Rost and Sander (Rost & Sander, 1994) when using ensembles of neural networks for secondary structure prediction. This is probably because the single-structure networks used in this work are very well adjusted and that no over fitting is observed. The networks used by Rost and Sander have a considerable tendency to over-fit, and we believe that an important role of the ensemble in their work is to "average out" the over-fitting. This is possible if the members in the ensemble over-fit *differently*, i.e., they make different errors, and therefore their average output is generally better than the output of any single network in the ensemble (Hansen & Salamon, 1990; Krogh & Vedelsby, 1995).

3.3 Using multiple alignments of related proteins

To improve the performance of the ensemble method multiple alignments are applied as described previously. Since only minor differences where observed between the weighted majority scheme and the weighted average scheme only results from the latter (which tend to be the best) will be presented in the following. As can be seen in Table 4 the difference between uniform weighting and entropic weighting is surprisingly small. Furthermore, the entropic weighting seems to be slightly inferior to the uniform weighting. As already discussed, any weighting scheme suffers from assigning large weights to erroneous alignments, and that might be one of the reasons for this. However, using the combined weighting scheme a gain of approximately 0.5% is seen compared to both uniform and entropic weighting.

Using a network to filter the alignment prediction as described in section 2.4 gives an amazing gain of more than 2% in the three-state accuracy. The filter takes a window of 15 "raw" alignment

predictions, the column entropy and the weighted number of InDels as input. The network is fully connected and contains 5 hidden units. Thus, the filtered alignment prediction yields $Q_3 = 71.3\%$, and the corresponding Matthews correlation coefficients of $C_\alpha = 0.59, C_\beta = 0.52$, and $C_c = 0.50$ indicates a very good prediction. Comparing the filtered alignment prediction to the one obtained using single sequences a gain of 5% is observed.

In order to further improve the performance, the following additional inputs were tried.

1. Normalized distance from the central residue to the ends of the protein
2. Normalized length of the protein
3. Frequency of the 20 amino acids in the base protein

The last two inputs contain global information about the protein under consideration. However, none of these attempts lead to significant improvements (they all resulted in a gain of less than 0.1%).

Compared to the single-sequence method the alignment method obtains 5% higher classification rate, and a considerable increase is seen in the Matthews correlation coefficients for all three structures. This confirms that evolutionary information is extremely important in the description of secondary structure from amino acid sequences. In the work of Levin *et al.* (Levin *et al.*, 1993) a gain in accuracy of almost 7% is reported when applying multiple alignments to a combination of the GOR (Garnier *et al.*, 1978) and SIMPA (Levin & Garnier, 1988) methods. A similar gain is reported by Rost and Sander (Rost & Sander, 1994) when using profiles to train a neural network resembling the one used by Qian and Sejnowski. The smaller gain of 5% found in this work is probably due to a better single-sequence method than the ones used by the above mentioned authors.

The increase in prediction accuracy when using multiple alignments is mostly due to a better prediction of α-helices and β-strands as shown in Table 5. Thus, the increase in Q_c is only about 2% compared to more than 11% for Q_β. This indicates that multiple alignments mostly contributes information about regular secondary structures in agreement with the fact that the three-dimensional structure of a protein family mainly is determined by the approximate location of helices and strands (Rost *et al.*, 1994b). Hence, the ends of regular secondary structure segments are less well defined than the core of regular secondary structure segments. This is verified in Table 5 where it is seen that the core of helix and strand segments are predicted considerably better than the mean for all residues. The corresponding percentages for coil shows that the core and ends of coil segments are approximately equally well defined.

As discussed earlier, the performance of the prediction method should not be based only on the percentages of correctly predicted residues. In order to see how realistic the prediction is the predicted and observed mean lengths of secondary structure segments are shown in Table 5. It is seen that the alignment method gives a much better prediction of segment lengths for helices and strands than the single-sequence method. The predicted helix segments have nearly the same lengths as the observed helix segments and the underprediction of β-strands tends to be slightly worse for the ensemble method. However, the overprediction of coil seems to remain unchanged when using alignments.

Since β-sheets often contain non-local interactions the strands are poorly defined from local sequences of amino acids. This is reflected in the β-strand prediction shown in Table 5; only $Q_\beta = 57.0\%$ of the observed strands are being correctly predicted. This should be compared to $Q_\alpha = 68.9\%$ and $Q_c = 79.2\%$. In some sense it is more interesting if the algorithm finds segments of helix or strands at approximately correct locations. Even though the strand prediction is clearly inferior to the helix prediction (in terms of Q_α and Q_β) segments of these two structures are located equally well. As shown in Table 5 an impressive 83% of all predicted helix segments overlaps with at least one observed helix segment. The corresponding percentage for strand

Ensembles	α-helix	β-strand	coil
Q_i (All)	64.2%	45.7%	76.9%
Q_i (Core)	71.5%	53.5%	79.5%
L_i^{Obs}	9.1	5.1	6.2
L_i^{Pred}	7.5	3.8	7.8
P_i^{Ovl}	68%	69%	91%
Alignments	α-helix	β-strand	coil
Q_i (All)	68.9%	57.0%	79.2%
Q_i (Core)	76.6%	67.1%	81.9%
L_i^{Obs}	9.1	5.1	6.2
L_i^{Pred}	9.3	4.4	7.8
P_i^{Ovl}	83%	80%	95%

Table 5: The performance of the ensemble and alignment method on each of the three secondary structures found by seven-fold cross-validation. "All" refers to all residues, while "Core" refers to all residues except the first and last residue in segments of secondary structure.

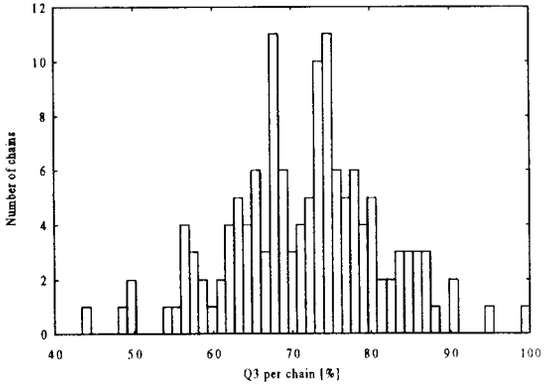

Figure 5: Distribution of per chain three-state accuracies obtained by the alignment method. The average three-state accuracy is $< Q_3^{Chain} >= 70.8\%$ with a standard deviation of $\sigma = 9.3\%$.

is 80%. These high overlap percentages illustrate that the alignment method is very good at locating and distinguishing segments of regular secondary structure.

For a new protein with unknown structure the performance is better described by the per chain accuracy Q_3^{Chain}. The alignment method yields $Q_3^{Chain} = 70.8\% \pm 9.3\%$. This is slightly smaller than the performance measured per residue, which means that long chains are predicted slightly better than short chains. Albeit the expected per chain accuracy lies between $Q_3^{Chain} = 61.5\%$ and $Q_3^{Chain} = 80.1\%$ the prediction can be significantly worse as illustrated in Figure 5. For four of the chains in the data set the three-state accuracy is less than 50%. Most prediction methods are good at capturing general features contained in the database used for training. Hence, the more atypical a given protein is compared to proteins in the training set the more likely is a poor prediction.

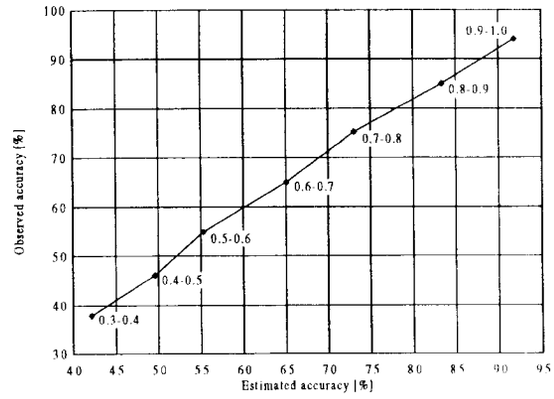

Figure 6: Observed accuracy versus estimated accuracy for the alignment method. Each dot on the curve corresponds to a prediction interval (0.3-0.4, 0.4-0.5 etc.). The estimated accuracy is given by the arithmetic average of network predictions in the given interval, and the observed accuracy by the three-state prediction percentage for these residues.

3.4 The neural network output as estimated probabilities of correct prediction

The prediction for a certain residue is given by the output unit with the largest output. The actual output value for this unit can be interpreted as the probability that the prediction is correct. To see if this interpretation is correct, one can find the actual accuracy of predictions for residues giving an output in a certain interval. In Figure 6 is shown the observed three-state accuracy versus the estimated accuracy for those residues producing an output in a certain interval. The estimated prediction accuracy is given by the arithmetic average of network predictions in the interval. The figure shows that a linear relationship exists between the estimated and the observed accuracy verifying that the network outputs can indeed be interpreted as estimated probabilities of correct prediction. Note that the lowest estimated probability is 0.33 since the three outputs must sum to one and since the prediction is chosen as the largest of the three outputs.

In Figure 7 is shown the observed accuracy plotted against the percentage of residues predicted with outputs above a certain value. This is another way to see that the higher output of the filter network the more reliable is the prediction. In this figure one can see that 72% of the database yields $Q_3 = 80\%$ and 36% scores about $Q_3 = 90\%$. Thus, for more than 36% of the database an accuracy comparable to that of homology methods is achieved. This position specific reliability measure can be used to locate those regions of a new protein with unknown structure that are predicted with particular high confidence thereby making an experimental determination of the structure considerably easier. These results are very similar to the results of (Rost & Sander, 1994).

Since β-strands are predicted less accurate than both α-helices and coil the estimated probability for this structure is generally smaller than the probabilities for α-helices and coil. In Figure 8 the percentages of observed helices, strands and coil predicted with outputs in the given intervals are shown. Most β-strands are predicted with an output below 0.6 corresponding to a relatively uncertain prediction. In contrast an impressive 27% of all observed helices are predicted with outputs in the interval 0.9–1.0 corresponding to a very high reliability. Furthermore, helices are generally predicted with considerably higher confidence than both coil and

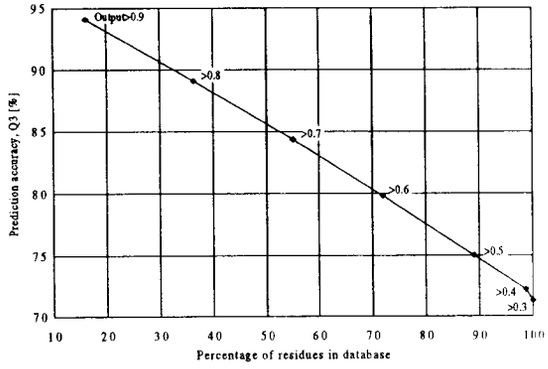

Figure 7: Observed accuracy versus percentage of residues yielding predictions above the value shown on the curve.

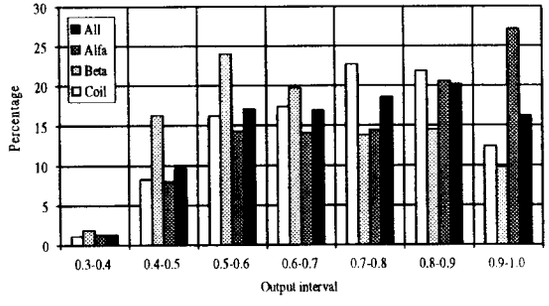

Figure 8: Percentage of observed helices, strands and coil predicted with a given output. Note that more than 27% of the observed helices are predicted in the interval 0.9-1.0 corresponding to very high confidence.

β-strands. This indicates that helices are better defined by local sequences of amino acids than the other structures, which is in agreement with the local nature of the helix structure.

4 Conclusion

Secondary structure prediction by use of highly structured neural networks and multiple alignments of homologous proteins have been investigated. By using small neural networks for predicting each of the three secondary structures over-fitting was avoided and a consistent stop criterion based on in-phase fluctuation of the training and testing error was developed. One of the features of the single-structure networks were an adaptive encoding of the amino acids, in which each of the 20 amino acid were represented by three real numbers. This alone decreases the number of network weights tremendously as compared to fully connected networks. The effect of this method was illustrated by a network for three state prediction containing only 311 adjustable weights, which outperforms a standard fully connected network with more than 10,000 weights! The low number of weights used in our single-sequence networks indicates that the implemented mapping from a window of the amino acid sequence to the secondary structure

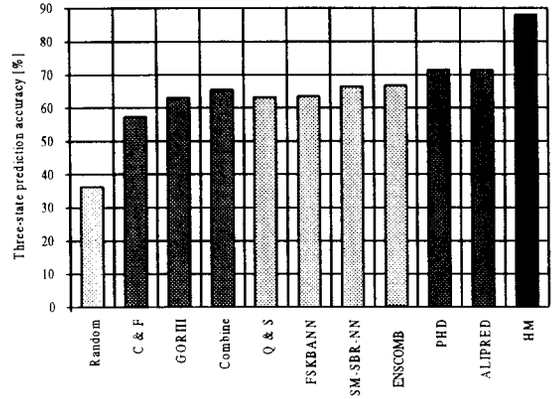

Figure 9: Three-state predictions obtained by methods using cross-validation. The methods are grouped into non-network methods, network methods, and network methods using the information contained in multiple alignments. Furthermore, the predictions that could be obtained from a random predictor (Random) and by homology modeling (HM) are shown. **Non-network methods:** C+F is the Chou-Fasmann algorithm (Chou & Fasmann, 1978), where the results are taken from (Maclin & Shavlik, 1993) where a cross-validation is performed on the data set used by Qian and Sejnowski; GORIII uses information theory (Gibrat *et al.*, 1987); Combine combines three different non-network methods (Biou *et al.*, 1988). **Network-methods:** Qian and Sejnowski is similar to the fully connected feed-forward network described in (Qian & Sejnowski, 1988) evaluated on the database given in Table 1, note that the performance is chosen as the best accuracy obtained on the testing sets, and spacer units are omitted; FSKBANN is a method that designs a multi-layer feed-back network from the Chou-Fasmann algorithm (Maclin & Shavlik, 1993); SM-MBR-NN is a method combining a statistical module, a memory based reasoning module and a neural network (Zhang *et al.*, 1992); ENSCOMB is our ensemble based single-sequence method. **Network-methods using alignments:** PHD is the profile network developed by Rost and Sander (Rost & Sander, 1994) using sequence profiles as input; ALIPRED is our alignment method. Note that PHD, ENSCOMB and ALIPRED all use the database shown in Table 1 and therefore can be compared directly. The data sets used in C+F, FSKBANN, and SM-MBR-NN have sequence similarities above 30% between proteins in the training and testing sets. All other methods are reported to have no significant sequence homology.

is relatively simple.

Another neural network was used to combine ensembles of the single structure predictors, and this gave a cross validated accuracy of 66.3%. This is as good as or even better than results obtained by most other prediction methods based on single-sequences as input (non-alignment methods). See Figure 9.

The use of multiple alignments gave a considerable gain in prediction accuracy as shown in Figure 9, and the prediction accuracy of 71.3% obtained by the alignment method is comparable to the one obtained by Rost and Sander with their profile network (PHD) (Rost & Sander, 1994). In our work, the alignments were used in a very different way, because the predictions were done on the individual sequences first, and then combined, instead of using a profile as an input to the network. It is very interesting that the two methods perform the same, because the profile method averages out all high order correlations in the proteins. It indicates that these

correlations are of minor importance. We would also like to note the other major difference, namely that the combined system is an order of magnitude smaller than the PHD system in terms of the number of adjustable parameters.

In the predictions based on multiple alignments we weighted the sequences by the method described in (Krogh & Mitchison, 1995). This weighting did not improve performance, which we believe is a result of giving large weights to sequences that may be aligned slightly wrong. It turned out that combining the weighting scheme with uniform weighting gave the best results, although the gain was only about 0.5% compared to no weighting.

The interpretation of neural network outputs as conditional probabilities was illustrated by use of the softmax approach. A linear relationship between estimated and observed prediction accuracy was observed, and therefore residues predicted with particular high confidence can easily be identified. It is noteworthy that more than 72% of the residues in the data set were predicted with a network output larger than 0.6. This corresponds to an observed prediction accuracy of approximately 80% for these residues. The remaining 28% of the database was predicted with less accuracy indicating that the secondary structure of these residues are not well described by local windows of amino acids.

When training neural networks the final accuracy can depend on small fluctuations in initial conditions *etc.*, and thus the percentages can vary within an interval of about 0.5%. Therefore we have not reported on results from training with additional information that only gave of the order of 0.1% each. By including such information, fine-tuning the sequence weights, and training the whole system many times to pick the best, we would be able to come very close to 72% accuracy or maybe even higher. In that case, however, the cross-validation results would heavily influence the selection of the right combination, and thus the final estimate of the prediction accuracy would be biased. It is also important to notice that adding or removing a single sequence from one of the seven sets can change the performance by as much as 0.5%. For these reasons we do not believe it is reasonable to compare results at a very fine level.

Most of the ideas we had to improve the performance of neural networks have been tested in this project, which is actually a few more than reported in the present paper. Although we did not improve on the overall accuracy, when compared to the best methods, we believe that this type of work is important, because we learn about both protein secondary structure and about the prediction methods. It is well known that interactions between amino acids far apart in primary sequence but close in space are of immense importance to protein folding. To increase the accuracy of secondary structure predictions even further, we believe that these global interactions in some way must be taken into account.

Acknowledgments

We would like to thank Burchard Rost for supplying us with details of his own work, as well as helpful comments. We also acknowledge interesting discussions with Tim Hubbard, Søren Brunak, Chris Sander, Saira Mian, Benny Lautrup and several others. This work was supported by a grant from the Novo Nordisk Foundation.

References

Altschul, S., Carroll, R., & Lipman, D. (1989). Weights for data related by a tree. *Journal of Molecular Biology,* **207** (4), 647–653.

Biou, V., Gibrat, J.-F., Levin, J., Robson, B., & Garnier, J. (1988). Secondary structure prediction: combination of three different methods. *Protein Engineering.* **2**, 185–191.

Bohr, H., Bohr, J., Brunak, S., Cotterill, R., Lautrup, B., Nørskov, L., Olsen, O., & Petersen, S. (1988). Protein secondary structures and homology by neural networks: The α-helices in rhodopsin. *FEBS Letters,* **241**, 223–228.

Bridle, J. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: *Neural Information Processing Systems 2,* (Touretzky, D., ed) pp. 211–217, San Mateo, CA: Morgan Kaufmann.

Chou, P. & Fasmann, G. (1978). Prediction of the secondary structure of proteins from their amino acid sequence. *Advances in Enzymology,* **47**, 45–148.

Garnier, J., Osguthorpe, D., & Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology,* **120**, 97–120.

Gerstein, M., Sonnhammer, E., & Chothia, C. (1994). Volume changes in protein evolution. *Journal of Molecular Biology,* **236** (4), 1067–1078.

Gibrat, J.-F., Garnier, J., & Robson, B. (1987). Further developments of protein secondary structure prediction using information theory. *Journal of Molecular Biology,* **198**, 425–443.

Hansen, L. & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **12** (10), 993–1001.

Hayward, S. & Collins, J. (1992). Limits on alpha-helix prediction with neural network models. *Proteins,* **14** (3), 372–81.

Henikoff, S. & Henikoff, J. (1994). Position-based sequence weights. *Journal of Molecular Biology,* **243** (4), 574–578.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the Theory of Neural Computation.* Redwood City: Addison-Wesley.

Holley, H. & Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences of the United States of America,* **86**, 152–156.

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

Kneller, D., Cohen, F., & Langridge, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology,* **214** (1), 171–82.

Krogh, A. & Mitchison, G. (1995). Maximum entropy weighting of aligned sequences of proteins or DNA. Submitted.

Krogh, A. & Vedelsby, J. (1995). Neural network ensembles, cross validation and active learning. In: *Advances in Neural Information Processing Systems 7,* (G. Tesauro, D. S. T. & Leen, T. K., eds) , Cambridge MA: MIT Press. To appear.

Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation,* **1**, 541–551.

Levin, J. & Garnier, J. (1988). Improvements in a secondary structure prediction method based on a search for local sequence homologies and its use as a model building tool. *Biochim. Biophys. Acta,* **955**, 283–295.

Levin, J., Pascarella, S., Argos, P., & Garnier, J. (1993). Quantification of secondary structure prediction improvement using multiple alignments. *Protein Engineering*, **6**.

Maclin, R. & Shavlik, J. (1993). Using knowledge-based neural networks to improve algorithms: Refining the Chou-Fasman algorithm for protein folding. *Machine Learning*, **11**, 195–215.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Qian, N. & Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, **202** (4), 865–84.

Richard, M. & Lippmann, R. (1991). Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, **3**, 461–483.

Rost, B. & Sander, C. (1993a). Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, **90** (16), 7558–7562.

Rost, B. & Sander, C. (1993b). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, **232** (2), 584–599.

Rost, B. & Sander, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, **19**, 55–72.

Rost, B., Sander, C., & Schneider, R. (1994a). PHD – an automatic mail server for protein secondary structure prediction. *Computer Applications in the Biosciences*, **10** (1), 53–60.

Rost, B., Sander, C., & Schneider, R. (1994b). Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology*, **235** (1), 13–26.

Russell, R. & Barton, G. (1993). The limits of protein secondary structure prediction accuracy from multiple sequence alignment. *Journal of Molecular Biology*, **234** (4), 951–7.

Sander, C. & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9** (1), 56–68.

Sibbald, P. & Argos, P. (1990). Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology*, **216** (4), 813–8.

Stolorz, P., Lapedes, A., & Xia, Y. (1992). Predicting protein secondary structure using neural net and statistical methods. *Journal of Molecular Biology*, **225** (2), 363–77.

Taylor, W. (1986). The classification of amino acid conservation. *J. theor. Biol.* **119**, 205–218.

Vingron, M. & Argos, P. (1989). A fast and sensitive multiple sequence alignment algorithm. *Computer Applications in the Biosciences*, **5** (2), 115–21.

Zhang, X., Mesirov, J., & Waltz, D. (1992). Hybrid system for protein secondary structure prediction. *Journal of Molecular Biology*, **225** (4), 1049–63.

Zvelebil, M., Barton, G., Taylor, W., & Sternberg, M. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, **195**, 957–961.