Contents

1 Introduction

As the number of completely sequenced genomes grows, the amount of proteins whose structure and function are unknown increases rapidly. To experimentally determine a protein structure is time-consuming, and therefore computer methods for protein structure prediction are highly desirable. During the last decade methods for fold recognition have been vastly improved. In general the basis is a comparison of protein sequences with unknown folds to a library of determined protein structures. This is motivated by the notion that there is a limited number of naturally occurring folds, most of which are probably already known [1]. Three times an evaluation of the current techniques has been performed, at the Critical Assessment in Structure Prediction (CASP) meetings [2], [3].

In fold recognition a distinction is made between pairs of proteins that are homologous, i.e. share a common ancestry, and pairs that are analogous, i.e. have no common ancestry. Since proteins with a common ancestry often have a high sequence identity, predicting a fold for an unknown protein is much easier if a homologous, structurally determined protein can be identified. This work focuses on the obviously more difficult task of predicting folds when no homologous proteins can be found.

The method developed in this work is based on sequence alignments between a protein of interest and a library of known structures. All available information about the alignments is processed through a neural network, which forms a relationship between the different alignment parameters and produces a single output. This output reflects the fitness of every proposed fold for this sequence.

However, using only alignment data is a rather limited way to predict a fold. More clues to finding the correct fold of a protein can be found if for example an evaluation of secondary structures is included in the model. It is reasonable to assume that the secondary structures of two proteins sharing a fold will be fairly similar. Another angle is to thread the sequence onto a known fold and calculate the energy of the sequence in the proposed conformation - a good fit will produce a low energy. In a series of experiments these two additional sources of information are combined with the alignment data. The aim is to find an optimal combination for fold prediction, and to study the effect of including different types of information to the neural network.

2 Background

2.1 Proteins and protein folding

A protein can be described as a chain of amino acids joined by peptide bonds in a specific sequence. There are 20 different amino acids, yielding infinite sequence combinations. However, polypeptide chains are not simply linear but are folded into compact shapes. The amino acid sequence is referred to as the primary structure of the protein. The three-dimensional structure of a protein is described by three additional levels. Different parts of the sequence

are folded into different secondary structures, based on the chemical properties of the contained amino acids. Secondary structures consist of regularly repeated conformations of the polypeptide chain, such as α-helices and β-sheets. These structures in turn fold into tertiary structures, bringing together distant parts of the amino acid chain. The tertiary structure is often referred to as the fold, and is closely related to the function of the protein. Some proteins also possess quaternary structure, the association of two or more polypeptide chains.

2.2 Fold recognition methods

In protein fold recognition the aim is to find the structure a new sequence is most likely to adopt. This means detecting similarities between protein 3D structures that are not accompanied by any significant sequence similarity. Rather than prediciting how a sequence will fold the approach is to predict how well a certain fold will fit the sequence. The classical way to do this is to compare and align the unknown sequence to a library of known folds.

There is a wide variety of approaches to solve the fold recognition problem. Some are based solely on sequence information, others on multiple aligned sequences, some on structural information or on combinations of sequence and structural information. One of the best choices so far is hidden Markov models, using multiple sequence alignments to build a sequence profile [4]. Another successful model is threading, where sequences of unknown structure are fitted directly onto the backbone coordinates of a known protein structure [5]. Each sequence- structure model is then evaluated using some type of energy function.

Some of the reasons why threading shows such good results are believed to be the facts that the hydrophobic interactions of the protein core are taken into account, but also that secondary structures are recognized by threading [6]. Proteins having a similar fold are also very likely to have similar secondary structure. Since secondary structures can be predicted from the amino acid sequence with an accuracy of more than 70% today [7], many protein fold recognition methods now incorporate secondary structure predictions to improve the performance of the method.

2.3 GenTHREADER

Fold recognition methods in general are slow and require human intervention to interpret the results. This has been circumvented in a new protein fold recognition method called GenTHREADER, developed by David T. Jones at the University of Warwick [8]. First a traditional sequence alignment algorithm is used to generate alignments. These are then evaluated by a method derived from threading techniques. Energy terms are calculated, relating how the unknown protein fits in the proposed fold. As a final step each model is evaluated using a neural network, thus producing a single measure of the proposed prediction.

The neural network is given six inputs, namely an alignment score and a corresponding alignment length, the lengths of the two sequences being aligned, a solvation energy and a pair energy. The results seem very promising, although

there is room for improvement. For example, Jones proposes that including secondary structure information in the model might improve the performance. The genTHREADER study is the basis for this work.

3 Methods

In order to detect proteins of the same fold, a four-stage method is developed in this work. The first step is alignment of sequences, followed by an evaluation of how well secondary structure elements are fitted against each other in the alignment. Next energy terms are calculated, providing a measure of the similarities between the sequence and the structural models. Finally all the data is evaluated using a two-layer, feed-forward neural network.

3.1 Datasets

3.1.1 SCOP

The Structural Classification of Proteins database [9], Scop, is a hierachical database containing all proteins in the Brookhaven Protein Databank, PDB. It provides a detailed description of the structural and evolutionary relationships of proteins whose three-dimensional structures have been determined. The quality of the database is considered to be high, since the classification is done manually by its creator, Alexei Murzin. The hand tuning is a major advantage of the database, since it makes Scop independent of any specific sequence or structure comparison algorithm. Scop has been used to compare different fold recognition methods in several different studies, see eg. [14] and [4] .

The proteins are classified on four levels - *family*, *superfamily*, *fold* and *class*. Proteins sharing a *family* have a clear evolutionary relationship. They either have residue identites of 30% and greater, or lower sequence identities but very similar functions and structures. The *superfamily* level contains families with probable common evolutionary origin. They have low sequence identities, but similar structures and functional features. Proteins placed together at the *fold* level have a major structural similarity, with the same major secondary structures in the same arrangement with the same topological connections. *Class* is simply a grouping of folds, such as all alpha, all beta, alpha and beta etc., introduced to facilitate for the user.

3.1.2 Training and test sets

The benchmark database used in this work was created from the PDB40d set of Scop version 1.39. No two proteins in this benchmark will therefore have more than 40% sequence identity. Structures where residues were missing were removed from the database, as well as non-protein entries. This left a benchmark of 1508 sequences, representing 340 folds, 499 superfamilies and 716 families.

When working with neural networks two sets of data are needed. One of the sets is used during the training process, and one is used as a test set to evaluate

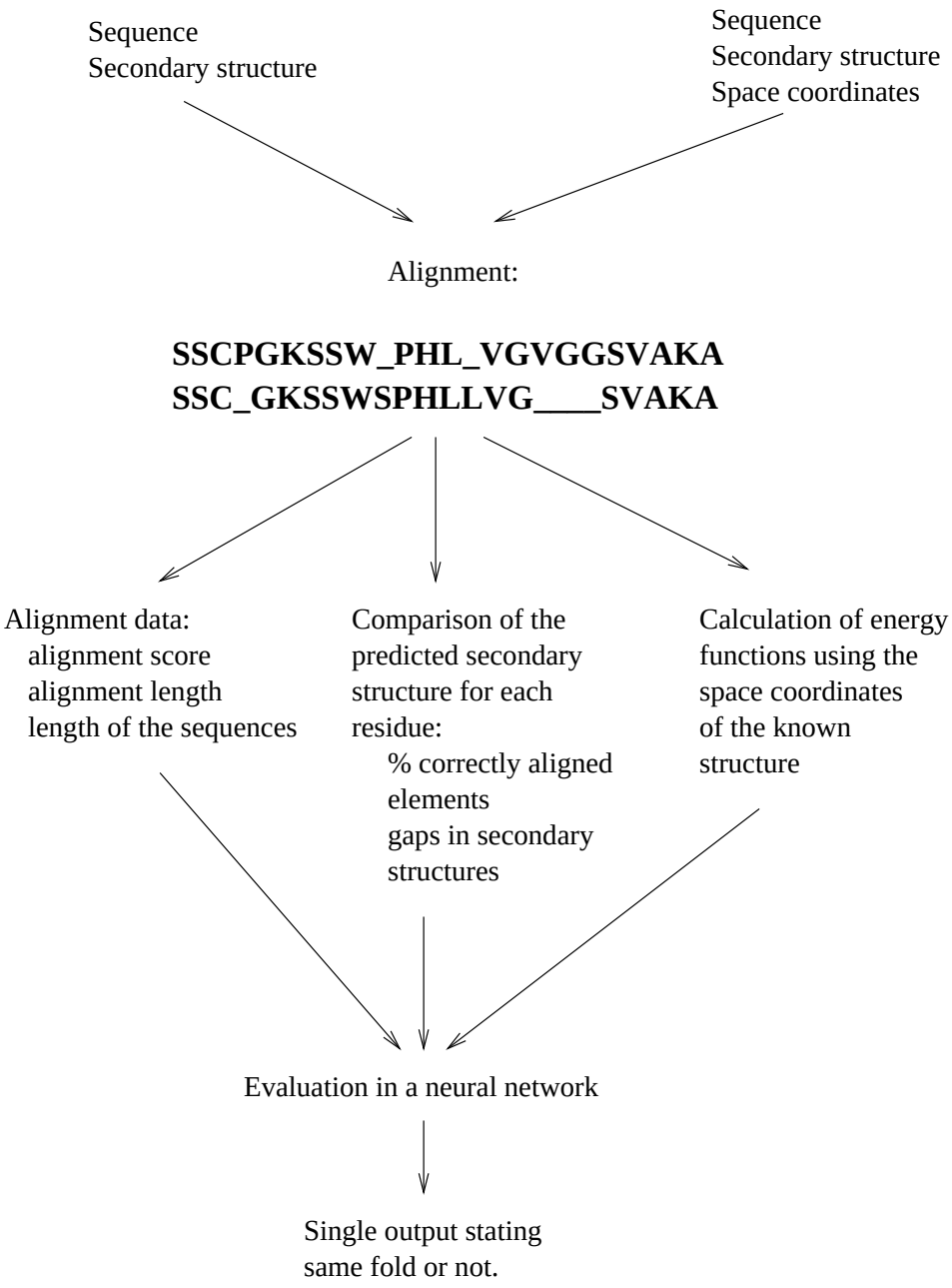

Figure 1: Overview of the method developed in this work.

the performance of the network. The 1508 sequences of the benchmark was therefore divided into three parts of 500, 500 and 508 sequences respectively. Two of these were used as training sets and one as a test set. The division into sets was done so that any specific fold, superfamily or family could only be found in one set, in order to avoid a biased test set.

Constructing a network is a very memory-consuming process, especially if many inputs and hidden neurons are used. Since all sequences were aligned against all other sequences in the same set very large training and testsets were obtained. These could not be used without restrictions if the training was to be computationally possible. Therefore a screening of all alignment scores was performed, removing all alignments with a score lower than 30. This approximately halved the sets, leaving about 250,000 alignments in the training sets and 125,000 in the test set.

3.2 Input data

3.2.1 Alignment algorithm

Aligning two or more sequences means loooking for character patterns that are the same in the sequences. This can be done in two ways, globally or locally, by altering the penalties for leaving gaps at the beginning and the end of an alignment. In this way a global alignment will find the best fit over the full length of the sequences, whereas a local alignment will optimize the alignment over smaller stretches that match very well. The problem to find the optimal alignment can be solved using dynamic programming, which is used in this work in the form of the Smith-Waterman algorithm[10] for local optimal alignment.

The Smith-Waterman algorithm solves the alignment problem by constructing a path matrix. The two sequences are represented by the dimensions in a two-dimensional array. The optimal alignment can then be drawn as a path originating at either border from one sequence and ending at the opposing border for the other sequence. All cells traversed by this path are aligned sequence pairs.

Different scoring matrices can be used in the calculation of an alignment, based on the physical and chemical properties of the amino acids. In this work the blosum50 and pam250 matrices are used. The pam matrices are based on the evolutionary distances between amino acids and relate to the number of point mutations that has occurred - pam stands for percent accepted mutation. Pam250 corresponds to 250 mutations in 100 amino acids [11].

Blosum stands for BLOcks SUbstitution Matrix. All sequences of the BLOCKS database were aligned, and based on the most conserved regions frequency tables were constructed, see [12]. Different levels were then created by weighting the degree of similarity between sequences. Blosum50 is derived from sequence blocks clustered at the 50% identity level.

Yet another set of parameters that can be varied in an alignment are the penalties for opening a gap in the alignment and for extending this gap. Decreasing the gap creation penalty favors the introduction of gaps into the alignment

Matrix	o p	e p
Blosum50	-10	-4
Blosum50	-10	-0.5
Blosum50	-5	-1
PAM250	-12	-2
PAM250	-10	-4

Table 1: List of all combinations of matrices and penalties used. o p = gap opening penalty, e p = gap extension penalty

whereas decreasing the gap extension penalty favors the formation of long gaps. Five different combinations of matrices and penalties were used in this work, and are listed in table 1.

Five parameters describing the alignment between two sequences are included in the model. These are the alignment score, the length of the aligned stretch, the length of the two sequences and the number of gaps in the alignment.

3.2.2 Secondary structure prediction

If the tertiary structures of two proteins are similar it is reasonable to assume that their secondary structures also should resemble each other to a large extent. The link between amino acid sequence and 3D structure however is not that strong. There are many examples of proteins exhibiting high structural similarity yet little or no sequence similarity. This depends on the fact that the environment of a particular residue tends to be more highly conserved than the particular identity of the residue. Since alignment scores are often low when studying proteins with no common ancestry, including a comparison between secondary structures might improve the prediction of fold and compensate for the lack of information at sequence level, as has been shown in several studies, see [7], [13] and [14]. An evaluation of how well secondary structures are fitted against each other in the alignment has therefore been included in the model.

The assignment of secondary structures for unknown protein sequences is done using the automatic algorithm STRIDE (secondary STRuctural IDEntification) [15]. STRIDE aims at reproducing the criteria used by crystallographers to practically assign secondary structures in newly determined protein structures. Two main structural properties are considered, namely hydrogen bond patterns and backbone geometry, expressed as mainchain dihedral angles. Each residue is classified as belonging to one of five states, H, E, T, G or C, where "H" stands for α-helix, "E" for β-sheet (i.e. extended conformation), "T" for turn, "G" for 3_{10}-helix and "C" for coil. In this work, "H" and "C" are both categorized as α-helix and "T" and "C" as turns.

Two parameters derived from evaluations of secondary structure similarities are calculated and included in the model. First, the percentage of correctly

aligned secondary structure elements is calculated using the following formula:

$$Sec.\ str. = \frac{\#correctly\ aligned\ secondary\ structure\ elements}{length\ of\ alignment} * 100 \qquad (1)$$

Second, the number of gaps that are located within important secondary structures are counted, i.e. gaps where the the amino acid in the "other" sequence is not classified as a "T" or a "C".

3.2.3 Energy functions

When an unknown sequence is aligned against a protein of known structure, the alignment provides a rough estimate of the structure of the unknown protein. If this implied structural model is evaluated, the fitness of the proposed fold can be examined. A common way of performing such an evaluation is to calculate knowledgebased potentials [2], [16], [17]. The coordinates of the known protein structure are combined with the alignment and transferred to the unknown sequence. In this way an energy function can be calculated and the implied model can be compared to other models.

Many different energy functions have been used for this kind of calculation, and it is not obvious what the ideal potential would look like. To be successful the function should be able to distinguish between correct and incorrect folds. However, this is not necessarily a feature of a function that mimics the real energy of a structure, that computationally would be very complex. Instead a very simple potential that roughly estimates the real world may be very well suited for the task. In this work one such simple function, developed by Park and Levitt [18], is used.

The energy function in question is a distance-dependent contact potential, similar to a van der Waals energy function. The protein structure is represented in a very simplified manner where each amino acid residue consists of two interaction centers, namely the α-carbon atom and the side-chain centroid. All α-carbons are considered energetically equivalent, whereas side-chains are distinct. The energy of a conformation is calculated according to:

$$\begin{aligned} E \quad &= \quad \sum_{(1\le i\le N)} \sum_{(i+4\le j\le N)} \quad \left(\frac{A_{ij}}{r_{ij}^8} - \frac{B_{ij}}{r_{ij}^4}\right) \\ &+ \quad \sum_{(1\le i\le N)} \sum_{(i+4\le j\le N)} \quad \left(\frac{A_{\alpha\alpha}}{r_{\alpha\alpha}^8} - \frac{B_{\alpha\alpha}}{r_{\alpha\alpha}^4}\right) \\ &+ \sum_{(1\le i\le N)} \sum_{(1\le j\le i-3)\bigcup(i+3\le j\le N)} \left(\frac{A_{i\alpha}}{r_{i\alpha_j}^8} - \frac{B_{i\alpha}}{r_{i\alpha_j}^4}\right) \end{aligned} \qquad (2)$$

where

$$\begin{aligned} A_{ij} &= -\epsilon_{ij}\,(R_{ij}^a)^8 \\ B_{ij} &= -2\epsilon_{ij}\,(R_{ij}^a)^4 \end{aligned} \qquad (3)$$

The contact energies are represented by ϵ_{ij}-values and the contact distances by R_{ij}-values, see eq. (3). These are tabulated in [18] and derived in [20]. The r_{ij}-values of eq. (2) correspond to the geometrical distance between two residues, and are calculated using x-, y-, and z-coordinates.

The three different terms of eq. (2) correspond to interactions between the α-carbons, the sidechains or between an α-carbon and a side chain. They will further on be referred to as $E_{\alpha\alpha}$, E_{ij} and $E_{i\alpha}$. The $E_{\alpha\alpha}$ term describes the shape of the backbone in the model. The specificity of the protein, that is how well each sidechain actually fits into the model, is described by the E_{ij} term. One of the most important factors in fold recognition is the hydrophobicity in the core of the protein, in other words the solubility of the protein. This is referred to by Jones as the solvation energy, see [8], and is represented in eq. (2) by the $E_{i\alpha}$ term. For a more detailed description of the energy function, see [18].

3.3 Neural networks

3.3.1 An introduction to neural networks

Artificial neural networks (ANN) are a class of computational algorithms that are based on the model of biological neural networks. A neuron, in f. eg. the human brain, is a very subtle device. It is capable of detecting and summing signals arriving on multiple input connections and transmitting these signals to other neurons. In a similar way the artificial neural network is able to learn complex relationships between multiple variables, reduce the complexity and produce a single output. The structure is built from several *nodes*, each corresponding to a neuron, see figure 2. The nodes are connected through synapses, where the synapse connecting the nodes i and j is characterized by a *weight*, w_{ij}. All input values will be scaled according these weights and summed to produce an output.

In this work a fully-connected, multilayer, feed-forward network is used, see figure 3, with error back-propagation. Typically this kind of network consists of a set of source nodes, that constitute the *input layer*, one or more *hidden layers* of computation nodes, and an *output layer* of computation nodes. The hidden neurons enable the network to learn complex relationships by extracting progressively more meaningful features from the input patterns. The learning process is also enhanced by the high degree of connectivity [21][22].

The model of each neuron in the network includes a sigmoidal nonlinearity at the output end in order to produce a continuous response to the inputs. This nonlinearity is defined by the *logistic function*:

$$y_j = \frac{1}{1 + e^{-v_j}} \tag{4}$$

$$v_j = \sum_j w_{ji} y_i \tag{5}$$

where v_j is the net activity level of neuron j, and y_j is the output of the neuron. Nonlinearity is one of the most important features of a neural network.

During the training process the weights of all synapses are iteratively adjusted to match a desired output. The objective is to adjust the free parameters

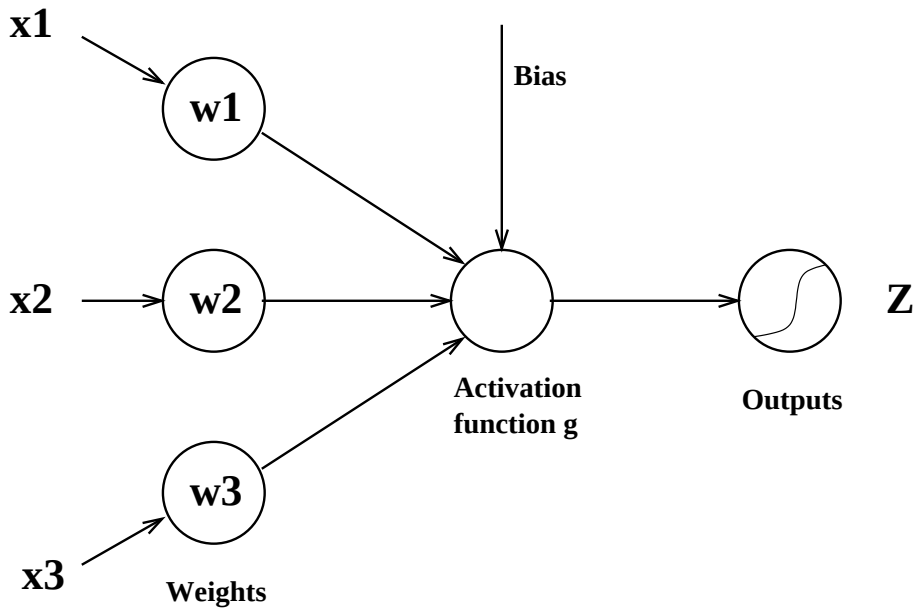

Figure 2: Simple illustration of a neural network. Neurons are represented by *nodes*. Each node is characterized by a weight, which is iteratively adjusted during the training process to match the desired output.

of the network so as to minimize the total error. The produced output is compared to a known, desired output. The error is then back-propagated through the net, adjusting the synaptic weights accordingly. In short the error back-propagation algorithm works as follows.

The error signal, $e_j(n)$, of output node j at the n:th iteration is defined by

$$e_j(n) = d_j(n) - y_j(n) \tag{6}$$

where $d_j(n)$ is the desired response and $y_j(n)$ is the actual response for neuron j. The total error function, summing the error over all neurons, is written as

$$E(n) = \sum \frac{1}{2} e_j^2(n) \tag{7}$$

The key factor of the algorithm is the calculation of the weight adjustment, $\Delta w_{ji}(n)$. In order to minimise the error the change needed should be proportional to, and in the opposite direction of, the gradient of the error function.

$$\Delta w_{ji}(n) = -\eta \frac{\delta E(n)}{\delta w_{ji}(n)} \tag{8}$$

η is a constant called the *learning rate,* determining the rate of the synaptic adjustment. Another way to express $\Delta w_{ji}(n)$ is

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{9}$$

where $\delta_j(n)$ is the local gradient of neuron j, pointing in the direction of the required changes of the synaptic weight. This local gradient is defined as

$$\delta_j(n) = e_j(n) f'(v_j(n)) \tag{10}$$

A full derivation of the error backpropagation algorithm can be found in [21]. All neural network models in this work were built using the NETLAB toolbox by Cristopher Bishop and Ian Nabney at Aston University, UK, a toolbox that runs under MATLAB^{TM}. At the time when this is written, a free copy can be downloaded from http://www.ncrg.aston.ac.uk/netlab/index.html. For all networks a logistic activation function was used. The scaled conjugate gradient algorithm was chosen as optimization algorithm during training.

3.3.2 Specificity, sensitivity and Mathews' correlation coefficient

Three measures are used to evaluate the performance of a network. The specificity indicates how many protein pairs are predicted to belong to the same classification (fold, superfamily or family) [23]. In other words, specificity is the probability that a protein with a score above a certain threshold, i.e. classified as having the same fold as the compared protein, really is a true hit.

$$Specificity = \frac{P^t(score)}{P^t(score) + P^f(score)} \tag{11}$$

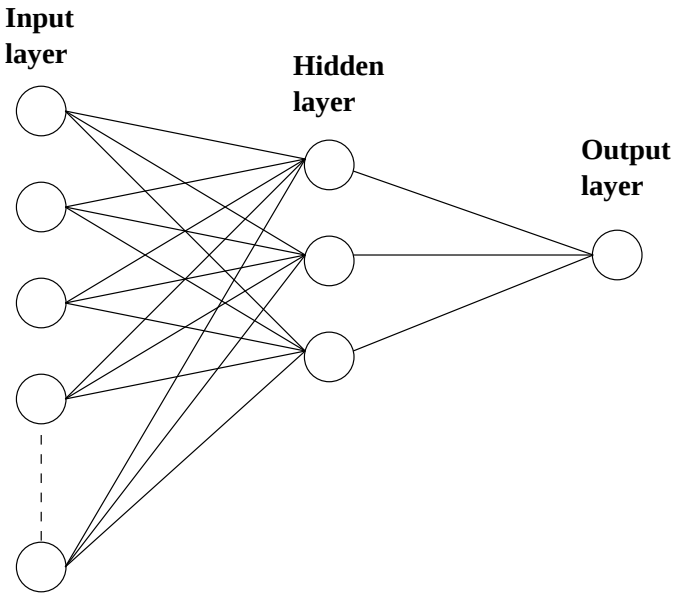

Figure 3: Illustration of a fully-connected, feed-forward, two-layer network. A set of source nodes constitute the input layer. The nodes of the input layer are connected to the computational nodes in a hidden layer. The final output layer consists of one or more computational nodes.

Sensitivity on the other hand shows the models ability to locate sequences with the same classification, i.e. the fraction of possible correct hits found.

$$Sensitivity = \frac{P^t(score)}{N^f(score) + P^t(score)} \tag{12}$$

Mathews' correlation coefficient, CM, is a more direct measurement of the network performance, indicating the accuracy of the classification [24]. A CM-value of 1 is obtained if the prediction by the network is totally correct. If the value is 0 however, the prediction is no better than a random guess.

$$CM = \frac{(P^t N^t) - (P^f N^f)}{\sqrt{(N^t + N^f)(N^t + P^f)(P^t + N^f)(P^t + P^f)}} \tag{13}$$

Like the specificity and sensitivity, the Mathews correlation coefficient uses a threshold score to separate correctly from incorrectly classified scores. In all three formulas P^t, P^f, $N^t$and $N^f$are the number of true positives, false positives, true negatives and false negatives respectively.

The network outputs are sorted in a descending order, whereafter specificities and sensitivities are calculated for each output at three classification levels - fold, superfamily and family. A complete separation of the different levels in the classification is done, discarding family hits when studying superfamilies and ignoring both family and superfamily hits when judging fold level performance. The specificity and sensitivity values are then plotted against each other, yielding a curve that in a detailed way shows the performance of a network. CM-values are calculated and plotted for a range of different cutoffs.

4 Implementation and testing

4.1 Analysis of raw data

First of all an analysis of the raw data was performed in order to establish what conclusions could be drawn directly from the raw data. Do the alignment score, the secondary structure information or the energy calculations by themselves provide enough information to discriminate between sequences sharing a fold and sequences with different folds?

4.1.1 Raw data distributions

The first study was a simple division of the testset into same fold/different fold categories. Each score was then plotted against the number of sequences that either belong to the same fold or not. In this way separation could easily be visualized. The analysis was performed for alignment scores, energy functions and secondary structure information respectively. Since the different penalty combinations as well as the choice of matrix has an impact on the alignment score, and thus on all other parameters since they are based on the alignment, the study was made on all different matrix/penalties combinations of table 1.

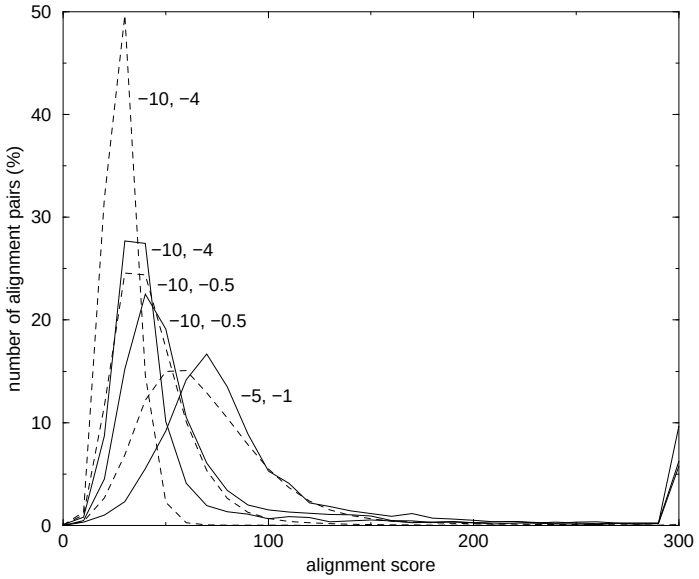

Figure 4: Alignment scores for the three different blosum50 penalty sets, see table 1 plotted against the number of alignment pairs in %. The data has been separated into sequences sharing a fold (solid line) versus sequences of different folds (dashed line), in order to see if a separation can be detected.

For all parameters the best separation was obtained using the blosum50 matrix with either penalties -10 and -4 or -5 and -1. The pam250 matrix did not show good separation for any parameters using either combination. This is probably because the penalty scores used were not really optimized for this matrix.

As can be seen in figure 4, the alignment score provides a slight separation between the two categories. The difference is however so small that it would not be enough to make even a qualified guess as to whether two proteins have the same fold or not. The same is true about the energy functions - figure 5 shows the separation for the total energy function (ref eq. (2)). The distinction between the two categories is even smaller here than for alignment score. As for secondary structure information, the percentage of correctly aligned secondary structure elements showed no separation at all for any matrix or penalty combination. It is clear that by themselves, neither of the parameters would be enough to make a clear prediction of fold for all proteins.

4.1.2 Specsens plots of rawdata

Another means of telling if a parameter can separate sequences sharing either a fold, a superfamily or a family from those that do not is to calculate and plot the specificity and sensitivity for each sequence pair. The ideal situation is to have a high sensitivity at all specificities, and to be able to obtain a specificity

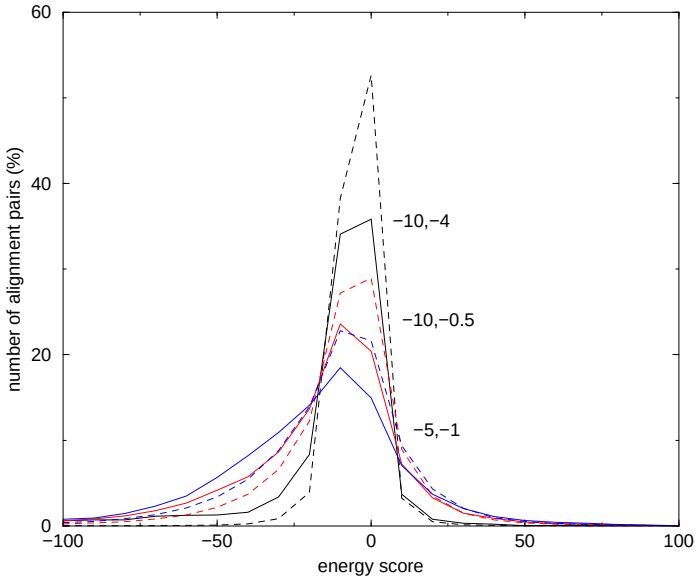

Figure 5: Total energy scores for the three blosum50 penalty sets of table 1, plotted against the number of alignment pairs in %. The data has been separated into sequences sharing a fold (solid line) versus sequences of different folds (dashed line), in order to see if a separation can be detected.

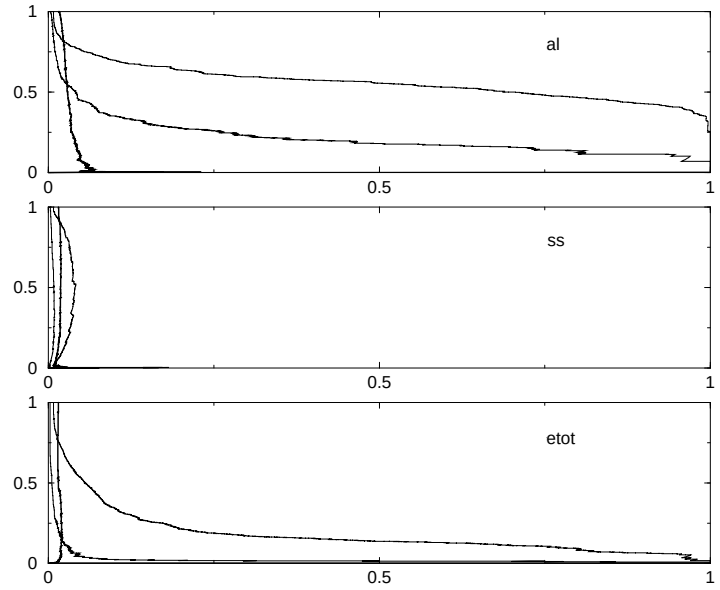

Figure 6: Specificity versus sensitivity plotted for alignment data (top), percentage of correctly aligned secondary structure elements (bottom) and total energy (bottom). Each plot contains three graphs - family level (thin line), superfamily level (medium line) and fold level (thick line).

of 1. That would mean that all pairs classified as belonging to the same fold really are true hits.

Specsens plots were constructed for each important parameter, see figure 6, in order to further establish the results from the above raw data analysis. Each plot contains three graphs, representing specificity and sensitivity calculated at family, superfamily or fold level. These plots confirm the above conclusion that alignment scores give the best discrimination. At family level the separation is very good and at superfamily level it is acceptable. Fold predicition however can not be performed using solely the alignment score - the results are far to bad. The plots further confirm that no separation at all can be done using only secondary structure information. When it comes to the total energy function, there is some separation at family level only.

In an attempt to look closer at the composition of the energy function, specsens plots were also constructed for the three energy terms $E_{\alpha\alpha}$, E_{ij} and $E_{i\alpha}$, see figure 7. The results are very clear - the small ability to discriminate between folds using the total energy function is due to only one term, namely E_{ij}. This term corresponds to the interactions between the sidechains in the protein, and correlates with the specificity of the proposed protein model.

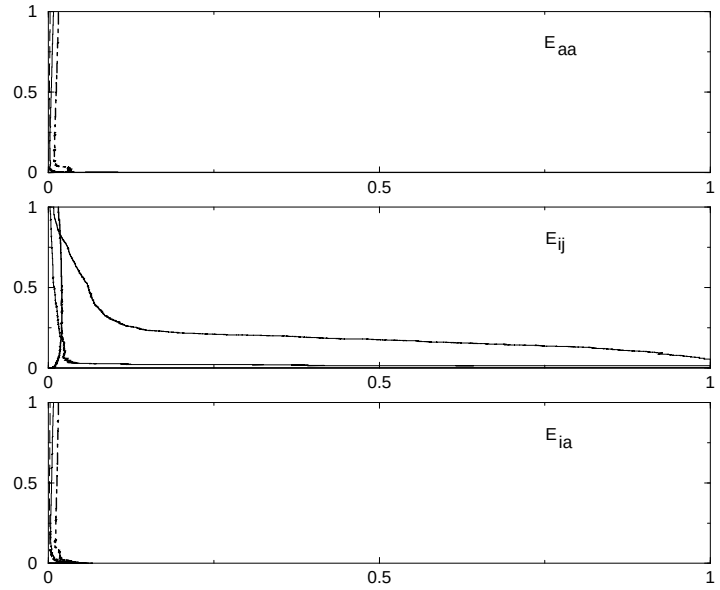

Figure 7: Specificity versus sensitivity plotted for the three energy terms. From top to bottom: $E_{\alpha\alpha}$, E_{ij} and $E_{i\alpha}$. Each plot contains three graphs - family level (thin line), superfamily level (medium line) and fold level (thick line).

4.2 Partial Least Squares Regression (PLS)

As a first step to finding a relationship between the different parameters, a linear model called Partial Least Squares Regression, PLS, is used here. PLS is based on linear multivariate regression and tries to find the best linear model based on a set of training examples, much as a neural network. During the training process, a weight is given to each parameter, and these weights can then be visualized. In this way it is possible to get a rough estimation of what parameters are important in the relationship. This work will not cover the details of PLS, since it is only used to visualize the importance of the different parameters. A detailed description can however be found at [25]. The PLS analysis was performed using the MATLAB^{TM} environment from The MathWorks, Inc. (USA) and the PLS_Toolbox software from Eigenvector Research, Inc. (USA).

Once a model has been built using PLS, the weights given to each parameter in the input vector can be visualized in a plot, see figure 8. In this way, a simple estimation of how the parameters should influence the final model can be obtained. This plot strengthens the tendencies of the raw data study, since the greatest weight is given to the alignment score. The secondary structure and the potentials are given approximately equal, but opposite, weights. E_{ij} is clearly the most informative energy term in this very simple linear model.

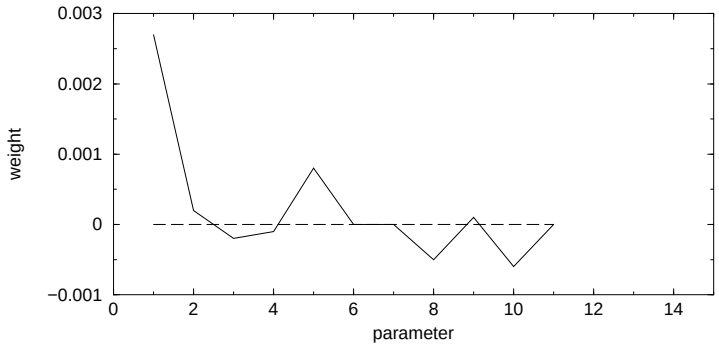

Figure 8: Illustration of the weights given to the eleven parameters in the PLS model. 1= alignment score, 2= alignment length, 3,4= sequence lengths, 5= percentage of correctly aligned secondary structures, 6= gaps in alignment, 7= gaps within secondary structures, 8= E_{tot}, 9= $E_{\alpha\alpha}$, 10= E_{ij} and 11= $Ei\alpha$.

4.3 Network series

4.3.1 Alignment data only

In an initial stage, a network was simply fed with alignment information. Four different parameters were used, namely the alignment score, the length of the aligned stretch and the lengths of the two sequences. These are the four parameters that are further on referred to as alignment data. The performance was optimal when the network used five hidden neurons and was trained at the superfamily level. At fold level, the number of possible correct hits is too small compared to the total number of sequence pairs for the network to detect a pattern. When trained at superfamily level however, the relationship found is so general that performance is good also when fold recognition is studied.

The top panel of figure 9 shows the specificity and sensitivity of the described network. A comparison with the best raw data specsens plot, i.e. the alignment score plot of figure 6, reveals a slight decrease in performance on family and superfamily level. On fold level however the network has the desired effect - the improvement is remarkable. The CM value curve looks promising, rising steadily with the cutoff until it reaches 0.38 at a cutoff of 0.9, see table 2.

Matrices and penalties that are optimal for obtaining the best alignment are not necessarily optimal for fold recognition. The different matrices are also optimal for different folds. Therefore, in order to obtain a maximum amount of information, the usage of several different matrices and gap penalties may be preferable [26]. Two different matrices are used in this work, blosum50 and pam250, together with different sets of penalties. The blosum50 matrix with an opening penalty of -10 and an extension penalty of -4 was used as default in all tests. All combinations used can be seen in table 1.

A study was made where alignment data from two matrix/penalties combinations at a time were used as input to a network. In all cases, the default

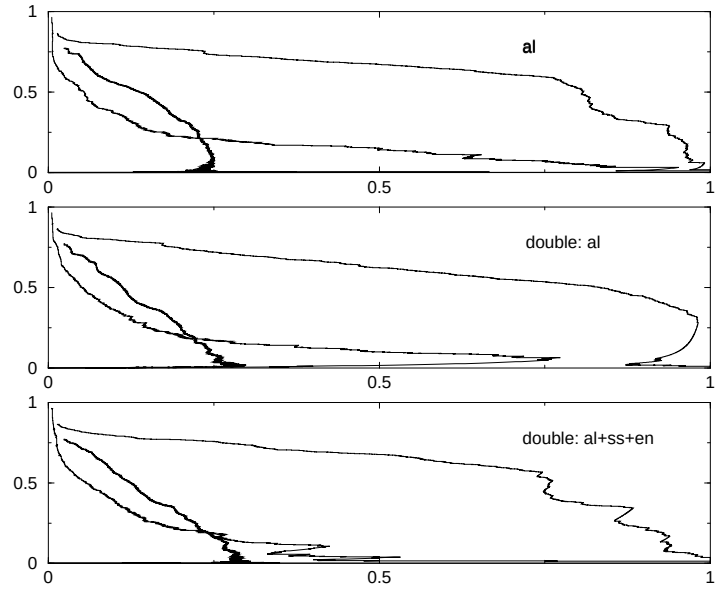

Figure 9: Specsens plots. The top panel shows the results for a network where only alignment data has been included. In the other two data from two different matrix/penalties combinations are used. The middle graph shows the results from a combination of alignment data only, whereas in the bottom one all eleven parameters have been used. In both graphs the matrices were blosum50 with penalties -10 and -4 and pam250 with penalties -12 and -2. The networks are trained at the superfamily level.

setup was combined with one of the other setups of table 1. The ten network inputs were used along with eleven hidden units, and the networks trained at superfamily level. However, the study did not give the expected results. The networks were obviously not able to find a good relationship between the different parametersets. The best combination was between blosum50 and pam250, using penalties -10/-4 and -12/-2 respectively. The specificity and sensitivity for this network is plotted in the middle panel of figure 9. The results at family level are approximately the same as when only one set of parameters is used, though at family level a decrease in performance has occurred. What is interesting to note is that performance has not changed at fold level. The CM vaue at a 0.9 cutoff is 0.37 and thus has not really changed either.

Studies have shown that alignment scores are often biased by the length of the aligned sequences [27]. Long sequences have more possibilites for aligning the sequences and thus the probability of a high alignment score is increased. This can partly be avoided by scaling the alignment score with a factor depending on the length of the sequences [28]. A scaling using the factor

$$\frac{alignment\ score}{ln(length1 * length2)} \tag{14}$$

Network	as	ss	en	$CM_{0.9}$
Raw alignment data	x			0.31
Sec. str. info. included	x	x		0.35
En. funcs included	x		x	0.41
Sec. str. info and en. funcs included	x	x	x	0.36
Alignment data doubled	x			0.37
All data doubled	x	x	x	0.36
Scaling between -1 and 1	x	x	x	0.48
Minimax scaling	x	x	x	0.47
Mean/SD scaling	x	x	x	0.07
Limited training set	x	x	x	0.51

Table 2: The Mathews' correlation coefficient, CM, is listed for all presented networks at a cutoff of 0.9. The al, ss and en columns refer to what types of data were included in the networks - as = alignment data, ss = secondary structure information and en = energy functions.

was tested in this work [29]. Also, feeding the network with a scaled alignment length parameter was attempted, according to

$$\frac{alignment\,score}{length1 + length2} \tag{15}$$

Networks were designed where the two parameters were combined with alignment data, one at a time or both. However, these parameters only seemed to confuse the network. Especially including the logarithmically scaled alignment score gave bad results, at all levels (specsens graphs not shown). When both parameters were included performance was approximately the same as without them. Therefore these parameters were abandoned in the following work.

4.3.2 Including secondary structure information and energy functions

In the next stage secondary structure information was added to the alignment data, cf section 3.2.2. Three parameters were added, namely the percentage of correctly aligned secondary structure elements, the number of alignment gaps in the secondary structures and, as a reference, the total number of gaps in the alignment. This raised the number of inputs to the networks to seven. Eight hidden units and 200 training cycles gave satisfying results. Default alignment settings were used.

Combining the alignment data with energy functions followed, cf section 3.2.3. Since more information can be retrieved if the eq. (2) is presented as separate terms, four energy parameters were used: E_{tot}, $E_{\alpha\alpha}$, E_{ij} and $E_{i\alpha}$. The eigth inputs were optimally followed by nine hidden units to provide good results.

Quite a big difference in performance can be seen when secondary structure information or energy functions are added to the networks, see figure 10. Especially at family level the results are improved. At superfamily level performance is also better, especially at high specificities. Including secondary structure information seems to give a slightly better result than including energy functions for high specificities at family and superfamily level. At fold level however, the situation is reversed although changes are minor. The energy function network manages to rise to somewhat higher specificities, but no real conclusion can be drawn from this. Since there is a natural variance in performance when the same network architecture is implemented several times, no change can really be noted at fold level.

Strangely enough the CM values show an inverted result, see table 2. At a cutoff of 0.9, the CM value decreases slightly to 0.35 when secondary structure is included, but rises to 0.41 when energy functions are included. The variations are small however, and could partly be due to the above mentioned natural variance in network performance. With all information taken together, the conclusion must be that including either energy functions of secondary structure information will increase the performance of a network. The increase in performance as a whole seems to be equal for both setups.

Judging from the specsens curves of the energy parameters, see figure 7, one might expect that it would suffice to use only the E_{ij} energy term or the total energy. However, although not detectable in the raw data, all energy terms contain valuable information of the fitness of the proposed fold. $E_{i\alpha}$ for example relates the hydrophobicity of the proposed model and can be compared to the solvation energy used by Jones [2], [8]. According to Jones, the solvation potential is the largest component of any useful fold recognition potential. The importance of each parameterer was stressed in an experiment where one parameter at a time was added as input to a network. It was clear that the best results were obtained when all energy terms plus the total energy were added together. Using only the total energy function, the family specsens curve would not rise to a specificity of one. At 50% specificity the sensitivity was 59 %, compared to 68% when all energy parameters were added separately.

Finally the alignment data, secondary structure information and energy functions were all fed to the neural network. As a total eleven parameters were used: alignment score, alignment length, lengths of the sequences, correctly aligned secondary structures, gaps within secondary structures, total number of gaps and four energy terms. The optimal network setup included twelve hidden neurons and 200 training cycles. The result can be seen in figure 11. When compared to networks with only alignment data as input a big improvement has taken place; for high specificities the sensitivity is better at both family and superfamily level. At fold level no real change can be detected using only the specsens plots.

While an improvement can be detected in a comparison with only alignment data, the results do not differ much when compared to the other to networks depicted in figure 10. By mere eye inspection the network with a combination of alignment data and secondary structure information seems to be the better one.

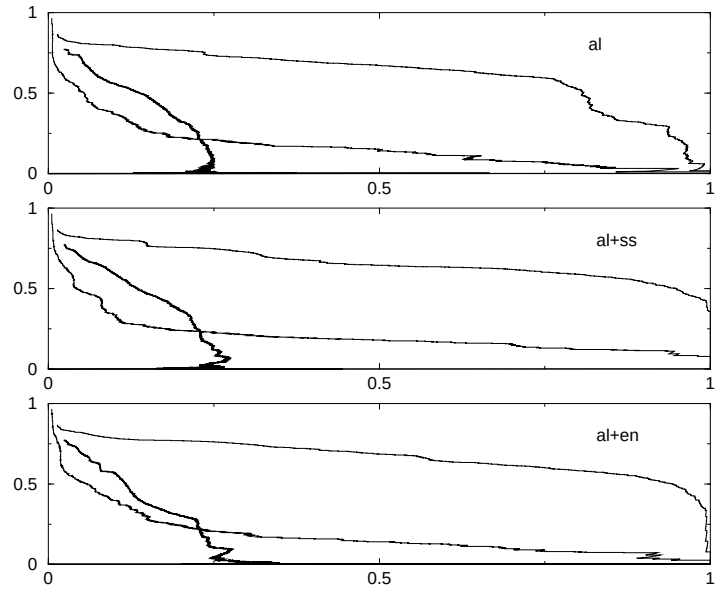

Figure 10: Specificity and sensitivity plots showing the gain in performance when secondary structure information and energy functions are added to the networks. In the top graph, only alignment data is used. In the middle secondary structure information has been added, whereas in the bottomgraph energy functions have been included. All networks are trained on the superfamily level.

Level	Spec.	raw as	as	as+ss	as+en	as+ss+en
Fam	100%	25%	0%	37%	7%	28%
Sufam	100%	8%	0%	7.5%	2.5%	5%
Fam	50%	55%	67%	64%	68%	64%
Sufam	50%	19%	14%	18%	14%	11%

Table 3: The table states the sensitivity at 100% and 50% specificity for three networks. As+ss symbolises the network with alignment data and secondary structure information, as+en the one with alignment data and energy function and as+ss+en contains all three. The family and superfamily levels are studied.

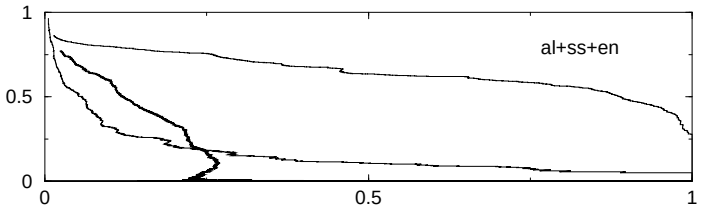

Figure 11: Specificity and sensitivity plots when alignment data has been combined with both secondary structure information and energyfunctions. The network is trained on the superfamily level.

Studying the sensitivity at 100% specificity in the three networks strengthens this conclusion, see table 3. Especially at family level the difference in sensitivity is large. These results also hold at 50% specificity, although the differences are smaller. However, as stated above, there is a natural variance in the performance of a network, which makes it difficult to draw any certain conclusions for such small differences in performance.

A closer look at the CM value also shows that no real improvement has taken place, see table 2. At a cutoff of 0.9 the CM value is 0.36, which is between the results for networks with either secondary structure information or energy functions. The variations are however too small for a proper analysis.

As in section 4.3.1 a series of networks were created combining information based on different matrices. The blosum50 matrix with penalties -10 and -4 was combined with the other setups of table 1. These very large networks were designed to use 22 inputs and 20 hidden units. The results can be seen in the bottom panel of figure 9. A large degeneration of the performance at family and superfamily level can be noted. Nevertheless, as in the middle panel where only alignment data is doubled, the fold prediction capability has not deteriorated - if anything it has improved a bit. It is also interesting to note that the CM value at a cutoff of 0.9, which is 0.36, is approximately the same as when a single parameter setup is used, i.e. 0.38, see table 2.

These results are ambiguous to interpret. Clearly the network is unable to find a good relationship between the many parameters, but still the results are maintained at fold level. This is indeed the case in all the networks presented

so far - whatever changes might be induced at family or superfamily level, the results are unaltered at fold level.

4.3.3 Scaling of the parameters

The parameters fed to the network have very broad ranges. The energies may vary between -500 and 500, whereas the gap parameters are usually between 0 and 10. It is possible that this will cause the network to favour certain parameters above others. Therefore a study was made where all parameters were scaled to be in the same range.

Three different types of scaling were used. In a first attempt all parameters were simply divided with the absolute maximum value. In this way the parameters were distributed between -1 and 1. Next minimax scaling was used, scaling all parameters between 0 and 1 using their minimum and maximum values. Finally mean/SD scaling was performed, using the scaling factor

$$\frac{x_i - \overline{x}}{\sigma} \tag{16}$$

where

$$\sigma = \sqrt{\frac{\sum (x_i - \overline{x})^2}{N}}. \tag{17}$$

The results of the scaling attempts can be viewed in figure 12. Scaling between -1 and 1 gives almost the same result as minimax scaling, and shows a slight improvement of the predictions at family level. On the superfamily and fold levels however these networks performed the same or slightly worse as networks without scaling. The CM values however are very interesting to study, see table 2. Calculated at a cutoff of 0.9 they have risen to 0.47 and 0.48 respectively - this is a 25% improvement from the unscaled network using the same parameters.

The mean/SD scaling gives poor results, with a CM of 0.07 at a cutoff of 0.9 and an extremely bad specsens plot. This is not surprising, since alignment scores belong to an extreme value distribution rather than a gaussian distribution. Since the alignment score is one of the major information sources for the network, a mean/SD scaling of this parameter should, and does, produce bad results.

4.3.4 Limiting the training set

The training set used in all the above experiments is unsorted, which means that some families or superfamilies are represented by many sequences whereas others are represented by only one. This approach was chosen since it agrees with the real world (as it is represented by the protein databases at least). There are many times more superfamilies and families than folds in for example the SCOP database.

In an attempt to determine whether this is positive or negative for the performance of a network a limitation of the training set was introduced. Only one

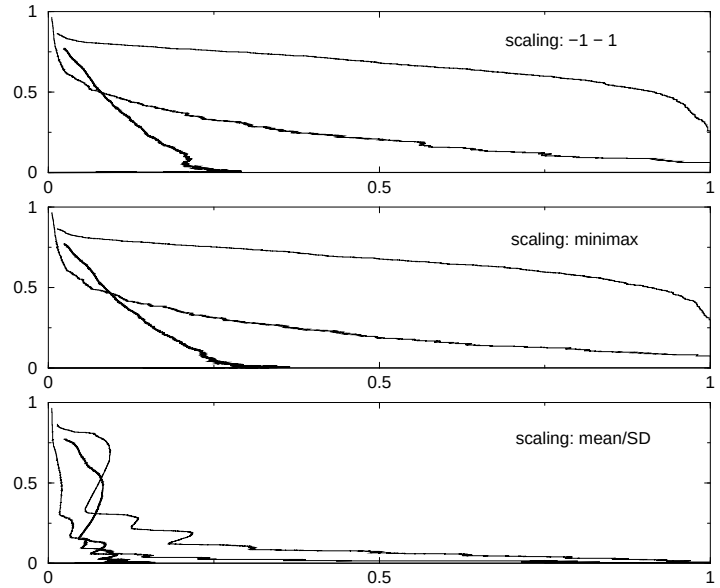

Figure 12: Specificity versus sensitivity for networks where the inputs have been scaled. From top to bottom: scaling between -1 and 1, minmax scaling and mean/SD scaling. Cf section 4.3.3. All networks were trained at the superfamily level.

sequence per family was included in one training set, and only one per superfamily in another. The selection of representatives for a family or superfamily was based on what sequence has the best resolution of its structure in the PDB database. In cases where the resolutions were the same the choice was random. A full parameter set was used in these networks, i.e. alignment score, alignment length, sequence lengths, percentage of correctly aligned secondary structures, two gap parameters and four energy terms.

Imposing a limitation on family level has a rather interesting result. While performance at the family and superfamily levels is quite bad, performance at fold level is the same, or even a bit better, than with a complete training set, see figure 13. The CM value at a 0.9 cutoff is very interesting, see table 2. It rises to 0.51, which is clearly the best so far. This means that the predictions of the network are much more accurate. Presumably the elimination of sequence redundancy has equaled the amount of sequences in the different folding groups. Above all the tendency of the network to learn relations at superfamily and family level instead of at fold level seems to have been reduced. A limitation at superfamily level had the same tendency, but with worse performance (figure not shown).

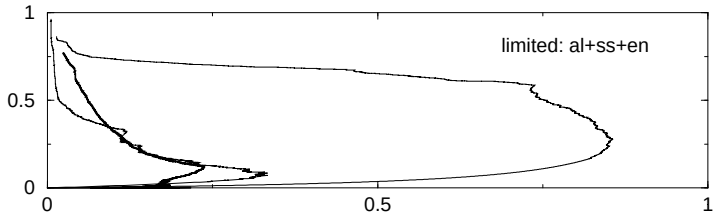

Figure 13: Specificity versus sensitivity for networks where the training set has been limited so as to include only one sequence per family. The network is trained on the family level with a complete test set.

5 Discussion

5.1 Network performances

Alignment scores relate to the degree of similarity between two sequences, and provide enough information by themselves to enable recognition of sequences within the same protein family. If the goal is to predict the fold of an unknown sequence however, it might be better not to study alignment scores only. Using information about the sequence lengths and the alignment length and processing this information through a neural network improves the possibility to predict the superfamily and, above all, the fold of a sequence. The network creates a complex, non-linear model that evaluates all inputs and produces a single output, stating the degree of similarity between the two sequences according to the model.

The aim of this study was to investigate what factors could further improve the ability of a neural network to predict fold. Energy functions evaluating the fitness of a proposed fold had been tested before by Jones, [8], whereas including secondary structure predictions was a new angle. The results presented above indicate that in both cases an improvement can be seen in the network performance, at all levels. However, a bigger difference at fold level was expected. The reason why the processing of the alignment data through a neural network gives such a drastic and immediate improvement at fold level is not quite clear. It might be suspected that the classifications in the SCOP database are not optimal for certain folds - some folds, especially within the small protein class, should probably be classified as superfamilies instead. Therefore the isolation of sequences with the same folds but not the same superfamilies might not be complete, causing the specsens curves at fold level to look better than they should.

Judging from the raw data study, energy functions provide more information than secondary structure information does. Therefore it is interesting to see that there is no real difference in performance between networks using secondary structure information and networks using energy functions. Combining them did not lead to significant improvements though - it even looks as if the performance decreased slightly. This may be a matter of optimizing the network architecture,

but it indicates that it is probably not worthwhile constructing bigger networks where the training period is rather time-consuming.

The following work consisted of finding ways to manipulate the inputs to the network in order to produce even better prediction capabilities. Several experiments were performed, such as scaling the data, limiting the training set, hinting the network by providing functions of some input parameters, and using alignment data from two matrices to make the network more general. Most of these alterations gave the same result - status quo or degeneration at superfamily and family level, but slight improvements at fold level. Most interesting are the scaling experiments, where the CM value rises 25% at minimax scaling, and the use of a limited training set, where the CM value rises 40%. Scaling removes the broad ranges of the input parameters, which might impare the performance of the network. By limiting the training set so as to include only one sequence per family the represented number of families, superfamilies and folds are more similar. Thereby the network is more likely to detect the vague relationships between sequences sharing a fold, which also seems to be the case.

5.2 Comparison of the network predictions

In almost all specsens plots of section 4.3 it is very hard to detect any real changes at fold level. Also, the complex model elaborated by a neural network is difficult to visualize. Therefore it is not clear whether a network has only learned to recognize a specific fold or if the predictions are really general. Studying the top 100 hits between sequences that share a fold but not superfamily reveals more of the specific predicting capability of a network.

Looking at the top 100 hits of all of the above presented networks reveals that they all have the same two types of fold among the top ranks, see tables 4 and 5. These folds both belong to the class of small proteins, which is usually dominated by metal ligand, heme, and/or disulfide bridges, as stated in the Scop database [9]. The first of the two folds is the knottin fold, whith members such as small inhibitors, toxins and lectins. Knottins are disulphide-bound and contain a β-hairpin. The second is the rubredoxin-like fold, which binds iron or zinc and contains two Cys-X-X-Cys motifs. The sequences of these are very short, 40-50 aminoacids only, which might be one reason why they are so easily recognized by the networks. However, the main reason why they get such high scores in all of the networks is probably a classification problem. Many of the different small protein folds should probably be brought together to superfamilies instead.

For the network where only alignment data has been used as input, the top 100 hits consist almost entirely of sequences belonging to these two folds. The scores are very evenly distributed among the top hits, ranging from 0.784 to 0.773. This means that there is no reliable fold identification, but all true hits seem to get fairly the same score.

When secondary structures are added the range among the scores is broadened. The top hits are as high as 0.955, and again the two small protein folds are overrepresented among the one hundred highest scores. However, at a score of 0.823 an $\alpha + \beta$ protein is found, namely an α/β-hammerhead. Further, at

Network	Top scoring fold	Score
al	Small proteins: knottin + rubredoxin-like	0.784
al+ss	Small proteins: knottin + rubredoxin-like	0.955
al+en	Small proteins: knottin + rubredoxin-like	0.949
al+ss+en	α/β proteins: β/α TIM-barrel	0.995

Table 4: The top score for four networks is listed. Network types: al = only alignment data is used, al+ss = secondary structure information is included, al+en = energy functions are included, al+ss+en = all data is combined.

0.700, there is an all β protein belonging to the carbonic anhydrase fold, which contains a single β-sheet with ten strands.

Even more folds turn up at high ranks when energy functions are added. The top 20 scores still consist of the small proteins, with scores between 0.949 and 0.944. The carbonic anhydrase fold turns up already at 0.939, and is followed by an α/β fold with the score 0.922. This is a β/α TIM-barrel. At 0.727 an all α protein is found - it is a vanadium-containing peroxidase. The structure is multihelical with a duplication. With such high scores, clearly the predictions are becoming more reliable. Also the generality of the network seems improved, since a wider range of folds are found at high scores.

The final combination of alignment data, secondary structure information and energy functions seems to lead to a slight decrease in performance when the specsens plots and CM values were studied. Studying the top 100 scores also reveals a slightly altered pattern. The top score is as high as 0.995, and corresponds to the β/α TIM-barrel. The following twenty scores are small proteins, and the scores decrease rapidly. At 0.631 the all α vanadium-containing peroxidase is found, as well as the all β carbonic anhydrase.

This study confirms the conclusion that adding secondary structure information and energy functions vastly improves the performance of the networks. A variety of different folds from different classes are highly ranked, especially in the energy function network. This indicates that the performance of the networks is rather general - they do not seem to recognize only one fold type. In combination with the study of CM values and specsens plots, it is still hard to determine if secondary structure information has an advantage over energy functions or the other way around. However, the data do confirm that combining them does not give better results.

An examination of the top ranks for each protein in the test set gives a more clear picture of the network performances. The number of proteins that were matched against a correct fold with a different superfamily at first rank and among the five and ten best ranks are listed in table 6, and the corresponding numbers for superfamilies with different families can be seen in table 7. First of all, these results show that the network with only alignment data does not have the same fold prediction capability as the augmented networks. They further indicate that at fold level the network where energy functions are included actually does perform slightly better than the network with secondary struc-

Fold	al	al+ss	al+en	al+ss+en
Small proteins: knottin	0.784	0.949	0.949	0.967
Small proteins: rubredoxin-like	0.784	0.949	0.949	0.963
$\alpha + \beta$ proteins: α/β-hammerhead	-	0.823	0.792	-
All β proteins: carbonic anhydrase	-	0.700	0.939	0.631
α/β proteins: β/α TIM-barrel	-	-	0.922	0.995
All α proteins: vanadium-containing chloroperoxidase	-	-	0.727	0.631

Table 5: The best score among the top one hundred is listed for six different folds and four network types. The codes for the networks are the same as in table 4.

Network	Rank 1	Rank 5	Rank 10
Raw	0	0	0
Al	0	0	0
Al+ss	0	1	11
Al+en	3	8	8
Al+ss+en	0	8	10
Minimax scaling	1	2	9
Limited tr. set	2	7	15
Double al	8	24	34

Table 6: Reflects how many proteins were matched against a correct fold with a different superfamily, at first rank and among the five and ten best ranks.

Network	Rank 1	Rank 5	Rank 10
Raw	0	0	0
Al	0	1	2
Al+ss	0	1	1
Al+en	0	1	3
Al+ss+en	5	8	9
Minimax scaling	0	2	3
Limited tr. set	4	7	11
Double al	1	4	7

Table 7: Reflects how many proteins were matched against a correct superfamily with a different family, at first rank and among the five and ten best ranks.

tures. At superfamily level the network with all data seems to have the best performance. It is also interesting to note what good rankings the finetuning experiments gave. Judging from the numbers at fold level doubling the alignment data certainly seems worthwhile, as well as limiting the training set and scaling the parameters.

6 Conclusions and future improvements

First of all, it has been shown in this work that fold recognition is vastly improved by a neural network processing of alignment data. At a minimum this data should consist of the alignment score, the alignment length and the length of the two sequences. Further, it is clear that the performance of the neural networks is improved when secondary structure information or energy functions are added as inputs. No real distinction can be made however between these two augmented networks - they seem to possess the same fold prediction capabilities. Judging from the top ranks however, it seems to be more rewarding to use potentials. It would be interesting to try out other energy functions - there is a wide variety currently in use in threading applications [16]. It is also possible that another means of presenting the secondary structure information could produce better results.

Combining all data in one network does not improve the network performances. Finetuning at fold level can however be accomplished using eg. minimax scaling or a limited training set. Doubling the alignment data by performing alignments with different scoring matrices seems very favorable, although the penalty sets need to be optimized.

The networks perform surprisingly well at fold level. The relationships found by the networks seem to be quite general - no fold is favoured above others. However it is probably a good idea to either exclude the class of small proteins from the training and test sets, or to make sure that the small proteins included really do belong to different folds. As a final comment, this approach to fold recognition seems promising and is worth studying further.

References

[1] A. G. Murzin and A. Bateman. Distant homology recognition using structural classification of proteins. *Proteins: Struct. Funct. Genet.* 1:105-112, 1997.

[2] D. T. Jones. Progress in protein structure prediction. *Curr. Op. Struct. Biol.* 7:377-387, 1997.

[3] M. J. E. Sternberg. Progress in protein structure prediction: assessment of CASP3. *Curr. Op. Struct. Biol.* 9:368-373, 1999.

[4] J. Hargbo and A. Elofsson. A study of hidden Markov models that use predicted secondary structures for protein fold recognition. *Proteins: Struct. Funct. Genet.* 36:68-87, 1999.

[5] D. T. Jones, W. R. Taylor and J. M. Thornton. A new approach to protein fold recognition. *Nature* 358:86-89, 1992.

[6] C. M.-R. Lemer, M. J. Rooman and S. J. Wodak. Protein structure prediction by threading methods: evaluation of current techniques. *Proteins: Struct. Funct. Genet.* 23:337-355, 1995.

[7] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* 267:1026-1038, 1997.

[8] D. T. Jones. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* 287:797-815, 1999.

[9] G. Murzin *et al.* SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247:536-540, 1995.

[10] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.* 147:195-197, 1981.

[11] M. Dayhoff, R. M. Schartz and B. C. Orcutt. A model of evolutionary change in proteins. In: M. Dayhoff, ed. *Atlas of protein sequence and structure, vol 5 (suppl 3).* Silver Spring, Maryland: National Biomedical Research Foundation. pp 345-352.

[12] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA* 89:10915-10919, 1992.

[13] D. Fischer and D. Eisenberg. Protein fold recognition using sequence-derived predictions. *Protein Science* 5:947-955, 1996.

[14] D. W. Rice and D. Eisenberg. A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *J. Mol. Biol.* 267:1026-1038, 1997.

[15] D. Frishman and P.Argos. Knowledge-based protein secondary structure assignment. *Proteins:struct., func. and gen.* 23:566-579, 1995.

[16] D. T. Jones and J. M. Thornton. Potential energy functions for threading. *Curr. Op. Struct. Biol.* 6:210-216, 1996.

[17] H. Flöckner, F. Domingues and M.J. Sippl. Proteins folds from pair interactions: a blind test in fold recognition. *Proteins: Struct. Funct. Genet.* 1:129-133, 1997.

[18] B. Park and M. Levitt. Energy functions that discriminate X-ray and near-native folds from well-constructed decoys. *J. Mol. Biol.* 258:367-392, 1996.

[19] B. Park *et al.* Factors affecting the ability of energy fuctions to discriminate correct from incorrect folds. *J. Mol. Biol.* 266:831-846, 1997.

[20] D. A. Hinds and M. Levitt. A lattice model for protein structure prediction at low resolution. *Proc. Natl Acad. Sci.* 89:2536-2540, 1992.

[21] S. Haykin. Neural networks. A comprehensive foundation. 1994, Prentice Hall Inc., New Jersey, USA. pp. 142-153.

[22] H. M. Cartwright. Applications of artificial intelligence in chemistry. 1993, Oxford University Press, Oxford, UK.

[23] E. Lindahl and A. Elofsson. Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.* 295:613-625, 2000.

[24] B. Mathews. *Biochim. Biophys. Acta* 405:442-451, 1975.

[25] A. Höskuldson. *Journal of Chemometrics* 2:211-228, 1980.

[26] R. A. Abagayan and S. Batalov. Do aligned sequences share the same fold? *J. Mol. Biol.* 273:355-368, 1997.

[27] S. F. Altschul *et al.* Gapped blast and psi-blast: a new generation of protein database search programs. *Nucl. Acids Res.* 25:3389-3402, 1997.

[28] W. R. Pearson. Comparison of methods for searching protein sequence databases. *Protein Science* 4:1145-1160, 1995.

[29] O. Emanuelsson, H. Nielsen, G. von Heijne. Manuscript under preparation.