J. Mol. Biol. (1988) **202**, 865–884

Predicting the Secondary Structure of Globular Proteins Using Neural Network Models

Ning Qian and Terrence J. Sejnowski

Department of Biophysics
The Johns Hopkins University
Baltimore, MD 21218, U.S.A.

(Received 25 September 1987, and in revised form 14 March 1988)

We present a new method for predicting the secondary structure of globular proteins based on non-linear neural network models. Network models learn from existing protein structures how to predict the secondary structure of local sequences of amino acids. The average success rate of our method on a testing set of proteins non-homologous with the corresponding training set was 64·3% on three types of secondary structure (α-helix, β-sheet, and coil), with correlation coefficients of $C_\alpha = 0{\cdot}41$, $C_\beta = 0{\cdot}31$ and $C_{coil} = 0{\cdot}41$. These quality indices are all higher than those of previous methods. The prediction accuracy for the first 25 residues of the N-terminal sequence was significantly better. We conclude from computational experiments on real and artificial structures that no method based solely on local information in the protein sequence is likely to produce significantly better results for non-homologous proteins. The performance of our method of homologous proteins is much better than for non-homologous proteins, but is not as good as simply assuming that homologous sequences have identical structures.

1. Introduction

Most of our knowledge of protein structure comes from the X-ray diffraction patterns of crystallized proteins. This method can be very accurate, but many steps are uncertain and the procedure is time-consuming. Recent developments in genetic engineering have vastly increased the number of known protein sequences. In addition, it is now possible to selectively alter protein sequences by site-directed mutagenesis. But to take full advantage of these techniques it would be helpful if one could predict the structure of a protein from its primary sequence of amino acids. The general problem of predicting the tertiary structure of folded proteins is unsolved.

Information about the secondary structure of a protein can be helpful in determining its structural properties. The best way to predict the structure of a new protein is to find a homologous protein whose structure has been determined. Even if only limited regions of conserved sequences can be found, then template matching methods are applicable (Taylor, 1986). If no homologous protein with a known structure is found, existing methods for predicting secondary structures can be used but are not always reliable. Three of the most commonly used methods are those of Robson (Robson & Pain, 1971; Garnier *et al.*, 1978), of Chou & Fasman (1978), and Lim (1974). These methods primarily exploit, in different ways, the correlations between amino acids and the local secondary structure. By local, we mean an influence on the secondary structure of an amino acid by others that are no more than about ten residues away. These methods were based on the protein structures available in the 1970s. The average success rate of these methods on more recently determined structures is 50 to 53% on three types of secondary structure (α-helix, β-sheet, and coil: Nishikawa, 1983; Kabsch & Sander, 1983*a*).

In this paper, we have applied a new method for discovering regular patterns in data that is based on neural network models. The brain has highly developed pattern matching abilities and neural network models are designed to mimic them. This study was inspired by a previous application of network learning to the problem of text-to-speech. In the NETtalk system (Sejnowski & Rosenberg, 1987), the input to the network is strings of letters representing words and the output is strings of phonemes representing the corresponding speech sounds. Predicting the secondary structure of a protein is a similar problem, in which the input symbols analogous to letters are amino acids and the output symbols analogous to phonemes are the secondary structures.

The goal of the method introduced here is to use the available information in the database of known

0022-2836/88/160865-20 $03.00/0

protein structures to help predict the secondary structure of proteins for which no homologous structures are available. The known structures implicitly contain information about the biophysical properties of amino acids and their interactions. This approach is not meant to be an alternative to other methods that have been developed to study protein folding that take biophysical properties explicitly into account, such as the methods of free energy minimization (Scheraga, 1985) and integration of the dynamical equations of motion (Karplus, 1985; Levitt, 1983). Rather, our method provides additional constraints to reduce the search space for these other methods. For example, a good prediction for the secondary structure could be used as the initial conditions for energy minimization, or as the first step in other predictive techniques (Webster *et al.*, 1987).

2. Methods

(a) *Database*

Proteins with known structures were obtained from the Brookhaven National Laboratory. Secondary structure assignments based on the atomic co-ordinates were assigned by the method of Kabsch & Sander (1983*b*). We selected a representative sample of proteins from the database that limited the number of almost identical sequences, such as the similar types of haemoglobin. Table 1 contains a listing of the 106 proteins that were used in our study. A subset of these proteins were taken out for testing and the remaining proteins used for the training set. Our results were highly sensitive to homologies between proteins in the testing and training sets, so homologies were exhaustively searched using diagon plots for all pairs of proteins (Staden, 1982). One of our 2 testing sets, listed in Table 2A, had practically no homologies in the training set. (α-Lytic protease in the testing set has very weak homologies with proteinase A in the training set but was included in the testing set to balance the proportion of residues with β-sheet structure. The inclusion of this protein reduced the overall testing accuracy, because β-sheet was the most difficult structure to predict.) A 2nd testing set with homologies is listed in Table 3A. The 6 proteins in the 2nd testing set had an average homology of 73% with 6 proteins in the corresponding training set, but little or no homology with the other training proteins, which were greatly in the majority. Special care was taken to balance the overall frequencies of α-helix, β-sheet and coil in the training and testing sets, as shown in Tables 2 and 3. The sequence of amino acids and secondary structures were concatenated to form 2 separate long strings for each of the training and testing sets, with spacers between the proteins to separate them during training.

Table 1

All proteins used to train and test networks

Code	Protein name	N	n_i	h	e	–
1abp	l-Arabinose-binding protein	1	All	106	18	182
1acx	Actinoxanthin	1	All	0	47	61
1apr	Acid protease	1	All	11	39	274
1aza	Azurin	2	1	13	43	73
1azu	Azurin	1	All	14	34	77
1bp2	Phospholipase A2	1	All	54	8	61
1cac	Carbonic anhydrase form c	1	All	18	68	170
1cc5	Cytochrome *c*5 (oxidized)	1	All	39	0	44
1ccr	Cytochrome *c* (rice)	1	All	44	0	67
1cpv	Calcium-binding parvalbumin b	1	All	52	6	50
1crn	Crambin	1	All	19	4	23
1ctx	α-Cobratoxin	1	All	4	16	51
1cy3	Cytochrome *c*3	1	All	16	0	102
1cyc	Ferrocytochrome *c*	1	All	35	0	68
1ecd	Haemoglobin (deoxy)	1	All	97	0	39
1est	Tosyl-elastase	1	All	13	82	145
1fc2	Immunoglobulin FC-Frag B complex	2	All	36	91	125
1fdh	Haemoglobin (deoxy, human fetal)	2	All	192	0	96
1fdx	Ferredoxin	1	All	5	4	45
1fx1	Flavodoxin	1	All	43	32	72
1gcn	Glucagon (pH 6-pH 7 form)	1	All	14	0	15
1gcr	γ-Crystallin	1	All	5	77	92
1gf1	Insulin-like growth factor	1	All	20	0	50
1gf2	Insulin-like growth factor	1	All	20	4	43
1gp1	Glutathione peroxidase	4	1,2	39	29	117
1hds	Haemoglobin (sickle cell)	4	1,2	152	0	135
1hip	High potential iron protein	1	All	10	9	66
1hmq	haemerythrin (met)	4	1	73	0	40
1ig2	Immunoglobunlin G1	2	All	15	186	255
1ige	Fc fragment (model)	2	1	16	121	185
1ins	Insulin	4	1,2	22	3	27
1ldx	Lactate dehydrogenase	1	All	114	45	170
1lz1	Lysozyme	1	All	39	10	81
1lzm	Lysozyme	1	All	83	14	67
1lzt	Lysozyme, triclinic crystal form	1	All	42	8	79
1mbd	Myoglobin (deoxy, pH 8·4)	1	All	113	0	40

Table 1 *(continued)*

Code	Protein name	N	n_i	h	e	–
1mbs	Myoglobin (met)	1	All	111	0	42
1mlt	Melittin	2	1	22	0	4
1nxb	Neutrotoxin b	1	All	0	26	36
1p2p	Phospholipase A2	1	All	45	6	73
1pfc	Fragment of IgG	1	All	4	34	73
1ppd	2-hydroxyethylthiopapain d	1	All	49	36	127
1ppt	Avian pancreatic polypeptide	1	All	18	0	18
1pyp	Inorganic pyrophosphatase	1	All	36	28	217
1rei	Immunoglobulin B-J fragment V	2	1	0	51	56
1rhd	Rhodanese	1	All	81	32	180
1rn3	Ribonuclease A	1	All	22	43	59
1sn3	Scorpion neurotoxin (variant 3)	1	All	8	12	45
1tim	Triose phosphate isomerase	2	1	106	42	99
1tgs	Trypsinogen complex	2	All	25	96	161
2act	Actinidin (sulphhydryl proteinase)	1	All	56	40	122
2adk	Adenylate kinase	1	All	108	22	64
2alp	α-Lytic protease	1	All	8	104	86
2ape	Acid proteinase, endothiapepsin	1	All	9	102	197
2app	Acid proteinase, penicillopepsin	1	All	30	147	146
2b5c	Cytochrome *b*5 (oxidized)	1	All	21	21	43
2cab	Carbonic anhydrase form b	1	All	17	77	162
2ccy	Cytochrome *c* (prime)	2	1	90	0	37
2cdv	Cytochrome *c*3	1	All	27	10	70
2cyp	Cytochrome *c* peroxidase	1	All	134	16	143
2dhb	Haemoglobin (horse, deoxy)	2	All	172	0	116
2fd1	Ferredoxin	1	All	0	0	106
2gch	γ-Chymotrypsin a	3	All	14	78	147
2gn5	Gene *5*/DNA binding protein	1	All	0	4	83
2grs	Glutathione reductase	1	All	125	86	250
2icb	Calcium-binding protein	1	All	47	0	28
2kai	Kallikrein a	3	All	17	86	188
2lh1	Leghaemoglobin (acetate, met)	1	All	107	0	46
2lhb	Haemoglobin V (cyano, met)	1	All	100	0	49
2mcp	Ig Fab mcpc603/phosphocholine	2	All	8	211	224
2mdh	Cytoplasmic malate dehydrogenase	2	All	213	110	327
2mt2	Cd, Zn metallothionein	1	All	0	0	61
2pab	Prealbumin (human plasma)	2	1	8	59	47
2rhe	Immunoglobulin B-J fragment V-MN	1	All	0	49	65
2sbt	Subtilisin novo	2	All	59	38	179
2sga	Proteinase A	1	All	12	98	71
2sns	Staphylococcal nuclease complex	1	All	26	28	87
2sod	Cu,Zn superoxide dismutase	4	1	0	58	93
2ssi	*Streptomyces* subtilisin inhibito	1	All	17	26	64
2stv	Satellite tobacco necrosis virus	1	All	18	82	84
2taa	Taka-amylase a	1	All	99	69	310
2tbv	Tomato bushy stunt virus	6	1,2,5	8	164	321
3c2c	Cytochrome *c*2 (reduced)	1	All	44	0	68
3cna	Concanavalin A	1	All	0	96	141
3fxc	Ferredoxin	1	All	7	15	76
3gpd	Glyceraldehyde-3-P-dehydrogenase	2	1	85	70	179
3hhb	Haemoglobin (deoxy)	2	All	196	0	92
3pcy	Plastocyanin (Hg^{2+} substituted)	1	All	4	35	60
3pgk	Phosphoglycerate kinase complex	1	All	143	46	226
3pgm	Phosphoglycerate mutase	1	All	69	15	146
3rp2	Rat mast cell protease	2	1	12	83	129
3sgb	Proteinase B	2	All	22	107	107
3tln	Thermolysin	1	All	118	52	146
451c	Cytochrome *c*551 (reduced)	1	All	38	0	44
4cts	Citrate synthase complex	2	1	223	18	196
4dfr	Dihydrofolate reductase	2	1	33	49	77
4fxn	Flavodoxin (semiquinone form)	1	All	47	29	62
4sbv	Southern bean mosaic virus coat protein	3	1,3	56	142	224
5atc	Aspartate carbamoyltransferase	4	1,2	134	62	268
5cpa	Carboxypeptidase	1	All	108	50	149
5ldh	Lactate dehydrogenase complex	1	All	124	31	178
5pti	Trypsin inhibitor	1	All	8	14	36
5rxn	Rubredoxin (oxidized)	1	All	0	8	46
6adh	Alcohol dehydrogenase complex	2	1	58	72	244
6api	Modified α-1-antitrypsin	2	All	109	124	142
8cat	Catalase	2	1	137	77	284

N, total number of subunit chains in the protein; n_i, subunit numbers used in this study; h, α-helix; e, β-sheet; –, coil.

Table 2
Proteins in testing and training set 1

A. *Testing set proteins with no homology with corresponding training set*

Code	Protein name
1abp	L-Arabinose-binding protein
1acx	Actinoxanthin
1hmq	Haemerythrin (met)
1ige	Fc fragment (model)
1nxb	Neurotoxin B
1ppd	2-Hydroxyethylthiopapain d
1pyp	Inorganic pyrophosphatase
2act	Actinidin (sulphhydryl proteinase)
2alp	α-Lytic protease
2cdv	Cytochrome c3
2grs	Glutathione reductase
2lhb	Haemoglobin V (cyano,met)
2sbt	Subtilisin novo
3gpd	Glyceraldehyde-3-P-dehydrogenase
6api	Modified α-1-antitrypsin

Total number of residues: 3520

Amino acid fractions

A:0·090	C:0·012	D:0·055	E:0·051	F:0·038
G:0·091	H:0·024	I:0·055	K:0·068	L:0·066
M:0·019	N:0·045	P:0·046	Q:0·035	R:0·032
S:0·072	T:0·070	V:0·079	W:0·014	Y:0·033

Secondary structure fractions

h, 0·241 *e*, 0·213 –, 0·547

B. *The training set*

Training set proteins: proteins in Table 1 minus Table 2A

Total number of residues: 18105

Amino acid fractions

A:0·087	C:0·015	D:0·056	E:0·048	F:0·038
G:0·086	H:0·024	I:0·045	K:0·067	L:0·083
M:0·015	N:0·048	P:0·045	Q:0·036	R:0·034
S:0·077	T:0·064	V:0·074	W:0·015	Y:0·035

Secondary structure fractions

h, 0·254 *e*, 0·201 –, 0·546

Table 3
Proteins in testing and training set 2

A. *Testing set proteins with homology with corresponding training set*

Code	Protein name
1p2p	Phospholipase A2
2ape	Acid proteinase, endothiapepsin
2rhe	Immunoglobulin B-J fragment V-MN
2sga	Proteinase A
3hhb	Haemoglobin (deoxy)
5ldh	Lactate dehydrogenase complex

Total number of residues: 1357

Amino acid fractions

A:0·101	C:0·012	D:0·051	E:0·032	F:0·034
G:0·103	H:0·026	I:0·041	K:0·047	L:0·091
M:0·012	N:0·054	P:0·036	Q:0·033	R:0·021
S:0·096	T:0·070	V:0·084	W:0·012	Y:0·035

Secondary structure fractions

h, 0·292 *e*, 0·211 –, 0·498

B. *The training set*

Training set proteins: proteins in Table 1 minus Table 3A

Total number of residues: 20268

Amino acid fractions

A:0·087	C:0·015	D:0·056	E:0·049	F:0·038
G:0·086	H:0·024	I:0·047	K:0·068	L:0·080
M:0·016	N:0·047	P:0·046	Q:0·036	R:0·034
S:0·075	T:0·064	V:0·074	W:0·015	Y:0·035

Secondary structure fractions

h, 0·249 *e*, 0·202 –, 0·549

(b) *Performance measures*

There are many ways to assess the performance of a method for predicting secondary structures. The most commonly used measure is a simple success rate, or Q_3, which is the percentage of correctly predicted residues on all 3 types of secondary structure:

$$Q_3 = \frac{P_\alpha + P_\beta + P_{\text{coil}}}{N}, \quad (1)$$

where N is the total number of predicted residues and P_α is the number of correctly predicted secondary structures of type a. The correlation coefficient (Mathews, 1975) is another useful measure, defined here for the α-helix:

$$C_\alpha = \frac{(p_\alpha\, n_\alpha) - (u_\alpha\, o_\alpha)}{\sqrt{(n_\alpha + u_\alpha)\,(n_\alpha + o_\alpha)\,(p_\alpha + u_\alpha)\,(p_\alpha + o_\alpha)}}, \quad (2)$$

where p_α is the number of positive cases that were correctly predicted, n_α is the number of negative cases that were correctly rejected, o_α is the number of over-predicted cases (false positives), and u_α is the number of underpredicted cases (misses). Similar expressions hold for C_β and C_{coil}. The Q_3 measure will be used to assay the overall success rate of network models during learning, although it is not as good an indicator as the individual correlation coefficients.

(c) *Neural networks and their properties*

The neural network models used in this study are based on a class of supervised learning algorithms first developed by Rosenblatt (1959) and Widrow & Hoff (1960). These are networks of non-linear processing units that have adjustable connection strengths, or weights between them, and learning consists in altering the values of the weights in response to a "teaching" signal that provides information about the correct classification in input patterns. In the present study, the teacher was the secondary structure assignments of Kabsch & Sander (1983*b*) based on the Brookhaven databank of protein structures. In this section, we give a brief introduction to feedforward neural networks and the back-propagation learning algorithm used in this study. Further details can be found in Rumelhart *et al.* (1986) and Sejnowski & Rosenberg (1987).

A feedforward network is composed of 2 or more layers of processing units. The first is the input layer, the last is the output layer, and all the other layers between are termed hidden layers. There are feedforward connections from all the units in one layer to those on the next layer, as shown in Fig. 1. The strength of the connection from unit j to unit i, called a weight, is represented by a real number, w_{ij}. The state of each unit, s_i, has a real value in the range between 0 and 1. The states of all the input

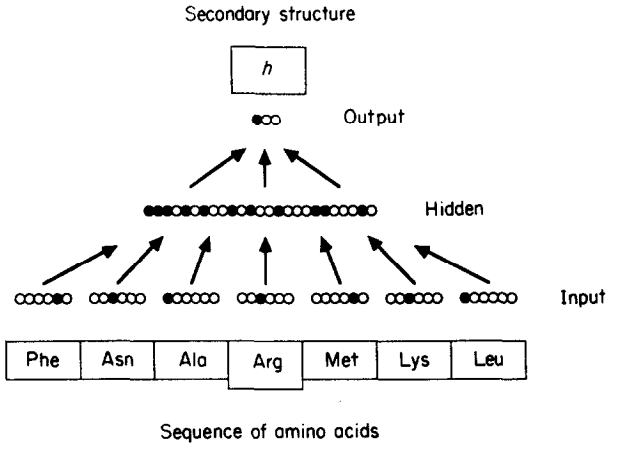

Figure 1. A diagram of network architecture. The standard network had 13 input groups, with 21 units/group, representing a stretch of 13 contiguous amino acids (only 7 input groups and 7 units/group are illustrated). Information from the input layer is transformed by an intermediate layer of "hidden" units to produce a pattern of activity in 3 output units, which represent the secondary structure prediction for the central amino acid.

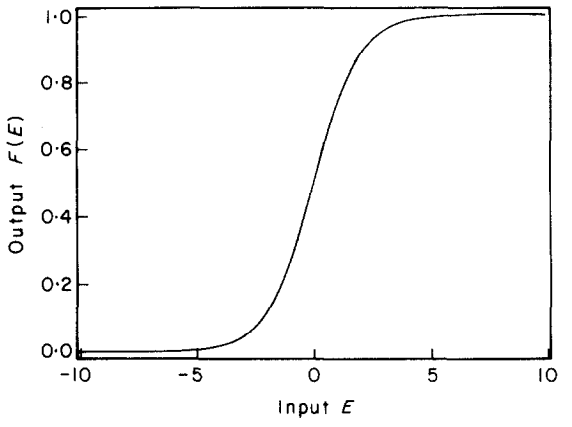

Figure 2. The output $F(E)$ of a processing unit as a function of the sum E of its inputs.

units that form an input vector are determined by an input window of amino acid residues (typically 13) through an input coding scheme (see the next section). Starting from the 1st hidden layer and moving toward the output layer, the state of each unit i in the network is determined by:

$$s_i = F(E_i) = \frac{1}{1+e^{-E_i}}, \tag{3}$$

where the total input E_i to unit i is:

$$E_i = \sum_j w_{ij} s_j + b_i \tag{4}$$

and b_i is the bias of the unit, as shown in Fig. 2.

The goal of this network is to carry out a desired input–output mapping. For our problem, the mapping is from amino acid sequences to secondary structures (as explained in detail in the next section). The back-propagation learning algorithm can be used in networks with hidden layers to find a set of weights that performs the correct mapping between sequences and structures. Starting with an initial set of randomly assigned numbers, the weights are altered by gradient descent to minimize the error between the desired and the actual output vectors.

A network with a single layer of modifiable weights (i.e. no hidden layers), called a "perceptron" (Rosenblatt, 1959), has been analysed extensively by Minsky & Papert (1969). An important concept introduced by them is the order of a mapping, defined as the smallest number n such that the mapping can be achieved by a perceptron whose input have supports equal to or smaller than n. The support of an input unit is the number of elements in the input array that are encoded by the input unit. For example, most of our networks use a local coding scheme in which the input units have a support of 1, since each of them codes only a single amino acid. We have also used 2nd order conjunctive encodings in which an input unit encodes combinations of 2 amino acids, and thus has a support of 2. By definition, if a mapping can be performed by a perceptron with the support of all of its input units equal to 1, then the order of this mapping is 1. Minsky & Papert (1969) showed very elegantly that many interesting mappings are of very high order and cannot be performed by a perceptron that does not have any input units with support larger than 1.

For the convenience of description, we define nth order perceptrons as those whose input units have size of support up to and including n. According to the above discussion, a 1st order perceptron can perform only a limited part of a higher order mapping correctly. In this paper, we define the 1st order features of a mapping as the part of the mapping that can be predicted by any 1st order perceptron, and the 2nd order features as the additional part of the mapping that can be performed by any 2nd order perceptron, and so on. With regard to the problem of predicting secondary structure of proteins, the 1st order features are the part of the mapping that can be predicted by each individual amino acid in the input window, and the 2nd order features are the part determined by all pairs of amino acids.

In principle, networks with hidden layers can extract higher-order features even when all of their input units have a support of 1. Learning algorithms for networks with more than one layer of modifiable weights have been introduced only recently (Ackley *et al.*, 1985; Rumelhart *et al.*, 1986). Not all of the information available may be extractable with a particular learning algorithm. An example is given in Results, section (a), where the back-propagation learning algorithm fails to recover a small amount of the 1st order features available to a 1st order perceptron.

A 1st order feature as defined above is stronger than the 1st order statistics used in standard statistical treatments. (We thank Dr Richard Durbin for pointing this out to us.) We illustrate the difference in the following example. Consider 2 sets of input–output mappings in Table 4. Define $P(I_i,O)$ as the joint probability that the ith ($=1, 2, 3, 4$) input unit is equal to I_i ($=0, 1$) and the output unit is equal to O ($=0, 1$). The joint probabilities are identical for both sets of mappings as shown in Table 5. Therefore, these 2 sets have the same 1st order statistics. However, these 2 sets can be learned by 2 different 1st order perceptrons with the weights given in Table 6. These 1st order perceptrons, therefore, have extracted more information than 1st order statistics. This observation will be used to explain why the neural network method yields better results than

Table 4
Two sets of mappings with identical first order statistics

Set 1					Set 2				
Input				Output	Input				Output
0	0	1	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0	1	0

the information theory method of Robson & Suzuki (1976).

(d) *Network design*

The network design used in this study is similar to the NETtalk system (Sejnowski & Rosenberg, 1987). The network maps sequences of input symbols onto sequences of output symbols. Here, the input symbols are the 20 amino acids and a special spacer symbol for regions between proteins; the output symbols correspond to the 3 types of secondary structures: α-helix, β-sheet and coil.

A diagram of the basic network is shown in Fig. 1. The processing units are arranged in layers, with the input units shown on the bottom and output units shown at the top. The units on the input layer have connections to the units on the intermediate layer of "hidden" units, which in turn have connections to the units on the output layer. In networks with a single layer of modifiable weights (perceptrons), there are no hidden units, in which case the input units are connected directly to the output layer.

The network is given a contiguous sequence of, typically, 13 amino acids. The goal of the network is to correctly predict the secondary structure for the middle amino acid. The network can be considered a "window" with 13 positions that moves through the protein, 1 amino acid at a time.

The input layer is arranged in 13 groups. Each group has 21 units, each unit representing 1 of the amino acids (or spacer). For a local encoding of the input sequence, 1 and only 1 input unit in each group, corresponding to the appropriate amino acid at each position, is given a value 1, and the rest are set to 0. This is called a local coding scheme, because each unit encodes a single item, in contrast with a distributed coding scheme in which each unit participates in representing several items. In some experiments, we used distributed codings in which units represented biophysical properties of residues, such as their hydrophobicity. Another coding scheme that we used was the 2nd order conjunctive encoding, in which each unit represented a pair of residues, 1 residue from the middle position and a 2nd residue at another position. Many more units are needed to represent a string of amino acids with conjunctive encoding, but this form of encording makes explicit information about the 2nd order features.

In the basic network, the output group has 3 units, each representing one of the possible secondary structures for the centre amino acid. In other versions, more output units were used to represent a larger number of possible secondary structures, or several groups of output units were used to represent a sequence of secondary structures. See Results for more details. For a given input and set of weights, the output of the network will be a set of numbers between 0 and 1. The secondary structure chosen was the output unit that had the highest activity level; this was equivalent to choosing the output unit that had the least mean-square error with the target outputs.

Based on the discussions in section (c), above, a network with a single layer of modifiable weights and using a local coding scheme for the amino acid sequence is a 1st order perceptron and so can detect only 1st order features, i.e. the independent contributions of each amino acid to the secondary structure. However, a network can extract higher order features, such as correlations between pairs of amino acids and the secondary structure, if conjunctive input coding schemes are used to construct higher-order perceptrons or "hidden" processing units are introduced between the input and output layers.

Table 5
First order statistics for the two mappings in Table 4

$P(I_i, O=0)$

I_i	Input position i			
	1	2	3	4
0	0·5	0·25	0·25	0·25
1	0	0·25	0·25	0·25

$P(I_i, O=1)$

I_i	In position i			
	1	2	3	4
0	0·25	0·5	0·25	0·25
1	0·25	0	0·25	0·25

Table 6
Weights for the two mappings in Table 4

Set	Input position i			
	1	2	3	4
1	2	−2	1	−1
2	2	2	−1	1

Two sets of weights for the 2 single-layer networks that perform the mappings in Table 4. The biases of all the units are 0.

(e) *Network training procedure*

Initially, the weights in the network were assigned randomly with values uniformly distributed in the range [−0·3, 0·3]. The initial success rate was at chance level, around 33%. The performance was gradually improved by changing the weights using the back-propagation learning algorithm (Rumelhart *et al.*, 1986). During the training, the output values are compared with the desired values, and the weights in the network are altered by gradient descent to minimize the error. Details about our implementation of the learning procedure can be found in Sejnowski & Rosenberg (1987). A diffent random position in the concatenated training sequence of amino acids (see section (a), above) was chosen as the centre position of the input window at each training step. The surrounding amino acids were then used to clamp the input units in

the window. All the amino acids in the training set were sampled once before starting again. This random sampling procedure was adopted to prevent erratic oscillations in the performance that occurred when the amino acids were sequentially sampled. The performance of the network on the testing set was monitored frequently during training and the set of weights was kept that achieved the best average success rate on the testing set. The training time of a network with 13 input groups and 40 hidden units was approx. 1 h of Ridge 32 time (equivalent to a VAX 780 FPA) per 10,000 residues.

The performance of the network on the training and testing sets depends on many variables, including the number of training examples, the number of hidden units, and the amount of homology between the training and testing sets. If there are too few training examples, the network can "memorize" all of the correct outputs because of the large capacity of the weights. The resulting network is accurate on the training set but makes poor predictions on the testing set. When the number of training examples is large, the learning procedure finds common features amongst the examples in the training set that enable the network to correctly predict the secondary structure of testing proteins that were not included in the training set. The ability of a network to extract higher order features from the training set depends on the layer of hidden units and the types of input encoding. A significant amount of information about homologies between proteins is contained in their higher-order features (see below).

3. Results

(a) *Artificial structures*

Before training networks on the database of known protein structures, we first tested the method on artificial structures with known input–output relations. The results of these experiments helped in interpreting the results of training on real proteins whose statistics are not known *a priori*.

(i) *Generation of first order artificial structures*

Amino acid sequences were chosen either from real proteins in the training set or generated from the statistics of those proteins. The first step in the latter method was to measure the frequency of occurrence of each amino acid in the database, as given in Table 2. Amino acids were chosen randomly with the same frequencies and assembled into a long sequence. Once the primary amino acid sequences were determined, the secondary structures were then assigned to each amino acid according to the information given in Tables 1, 2 and 4 of Garnier *et al.* (1978), which were based on the statistics of real proteins. These Tables were used by Robson and co-workers to predict secondary structures, but here we used them to generate artificial secondary structures.

Each amino acid in a window of 17 amino acids, eight on either side of the central residue, independently contributed toward an information measure for the type of secondary structures S:

$$I(s_i=S|R_{i-8}, \cdots R_i, \cdots R_{i+8}) = \sum_{j=-8}^{8} I(s_i=S|R_{i+j}), \quad (5)$$

where $I(s_i=S|R_k)$ represents the contribution from the kth position to the secondary structure s_i of R_i, which ranges over $S=\alpha$, β, coil. The secondary structure with the largest information measure was assigned to the central amino acid, and the process was repeated for each position in the sequence. This is a first order mapping.

(ii) *Prediction of secondary structure*

A network with 17 input groups having 21 units per group, 40 hidden units and three output units was trained on the training set of artificial structures. The learning curves shown in Figure 3 rose quickly for both the training set and a testing set of artificial structures. The learning algorithm is capable of discovering the "rules" that were used to generate the artificial structures and to predict with high accuracy the secondary structure of "novel" artificial structures that were not in the training set. Similar results were obtained when a network with one layer of weights and a local coding scheme was used. This was expected since, by construction, there were only first order features in the data.

The central amino acid has the largest influence on the secondary structure in the artificial structures, based on Robson's information tables. This should be reflected in the sizes of the weights from the input groups. The average magnitude of the weights from each input group is plotted in Figure 4 for a network at different stages of training. The average magnitude of the weights generally increased with time, but those at the centre more quickly than those near the ends of the window.

(iii) *Effects of noise in the data*

The long-range effects on the secondary structure would effectively add noise to the short-range effects that could be captured in a short window. In an effort to mimic these effects, we generated a new set of artificial structures that included a 30% random component to the rules used above. The networks with 40 hidden units and a local coding

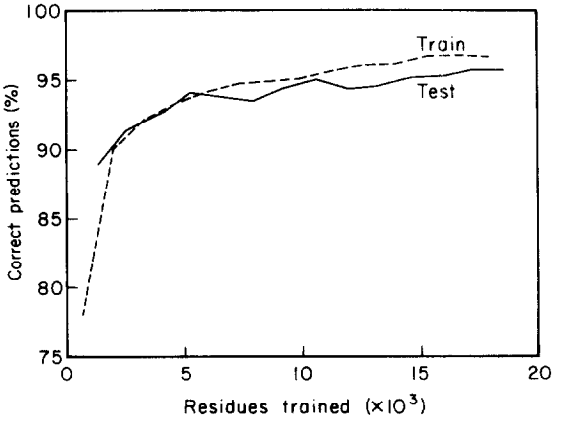

Figure 3. Learning curves for artificial structures. The percentage of correct predicted secondary structure is plotted as a function of the number of amino acids presented during training for both the training and testing sets.

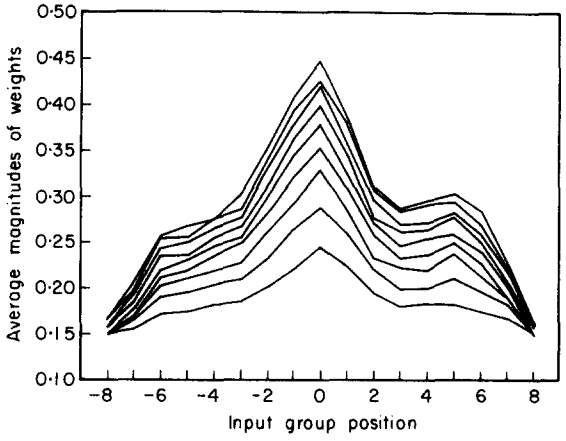

Figure 4. Values of the average magnitude of the weights from each weight group shown at several times during training on artificial structures. The lowest curve represents the averaged weights early in training.

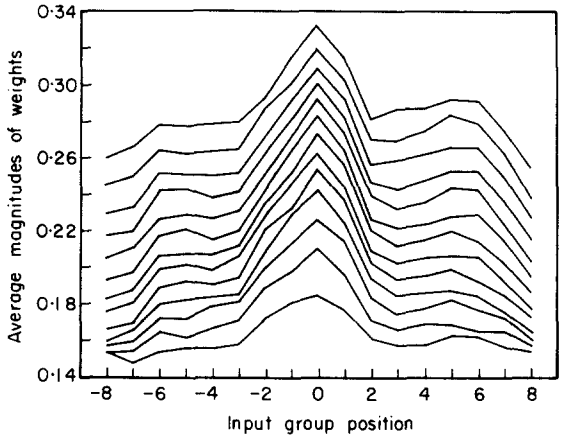

Figure 6. Values of the average magnitude of the weights from each weight group shown at several times during training on artificial structures with 30% noise added to the secondary structures. The lowest curve represents the averaged weights early in training.

scheme trained on proteins with 30% noise were able to learn the training, though not as well as the training set without noise, as shown in Figure 5. The performance on the testing set reached 63%, close to the theoretical limit of 70%. When a network with one layer of weights and a local coding scheme was used, both learning and training performances were about 63%. This indicates that the learning algorithm can extract 90% (63/70) of all the first order features. The noise had an interesting effect on the weights, as shown in Figure 6. The central weights were larger in magnitude, as before, but now even the weights from the end groups continue to increase with time. The ratio of the average magnitude of weights from the central group to the average magnitude of weights from the end groups was much smaller when noise was added to the training set.

(iv) *Effects of irrelevant weights*

Networks were trained having either 17 input groups or 21 input groups, but the secondary structures were generated from a group of size 17. The larger network was trained to determine the effect of the extra weights to inputs that could not contain any information about the secondary structure. The success rate of the network with 21 input groups was consistently lower than the network with 17 groups by about 1 to 2%. Thus, irrelevant weights can interfere with the performance of the network. The time evolution of the weights was different in the network with 21 input groups for the weights outside the middle window of 17 input groups. These weights fluctuated around 0·15, close to their initial, randomly generated values, compared with weights in the central groups that tended to increase with time.

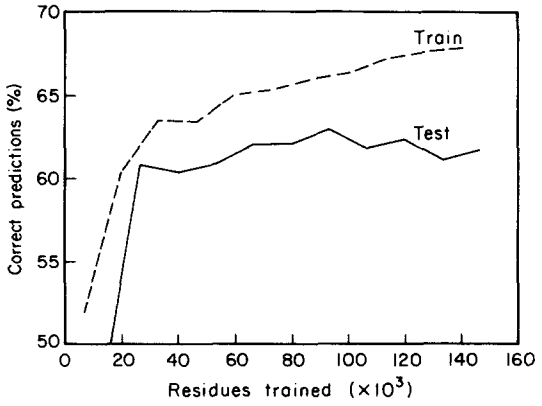

Figure 5. Learning curve for artificial structures with 30% noise added to the secondary structures. The percentage of correctly predicted secondary structures is plotted as a function of the number of amino acids presented during training for both the training and testing sets.

(v) *Second order artificial structures*

The first order common features of artificial structures were learned very quickly by a network with one layer of weights. We generated a new set of artificial structures with both first order and second order features to determine how well a network with hidden units could learn the higher order common features.

The second order contribution of the residue at R_{i+j} to the secondary structure of the residue at R_i depends jointly on the identity of both residues. We generalized the first order information measure given by Garnier *et al.* (1978) given in equation (5) to include this second order contribution:

$$I(s_i = S|R_{i-8}, \ldots R_i, \ldots R_{i+8}) = \sum_{j=-8}^{8} I(s_i = S|R_i, R_{i+j}) + B_S, \quad (6)$$

where B_S are constant biases used to match the relative proportion of secondary structures ($S = \alpha$, β, coil), to those of real proteins ($B_\alpha = 0$, $B_\beta = -55$, $B_{\text{coil}} = 150$), and:

$$I(s_i = S|R_i, R_{i+j}) = I(s_i = S|R_{i+j}) + A(S, R_i, R_{i+j}, j), \quad (7)$$

where $A(S,R_i,R_{i+j}, j)$ for each possible combination of (S,R_i,R_{i+j}, j) is a random (but fixed) number taken from a uniform distribution in the range $[-a, a]$. The magnitude of a determines the amount of second order features added into the original first order features. We chose $a = 100$ to match the fraction of first order features observed in real proteins.

When the local coding scheme for the inputs was used to train a network with one layer of modifiable weights (first order perceptron), the maximum testing success rate was 63%. This represents the amount of structure that can be predicted solely from the first order features. When a network with 80 hidden units and the same input coding scheme was used, the learning was much slower and the success rate was 65% and climbing very slowly after 70,000 training examples.

Improved performance was obtained using a second order conjunctive coding scheme for the inputs as described in Methods. This coding scheme makes it possible for a network with only one layer of weights to have access to second order features for the inputs. When such a network was trained on the artificial second order structures, the learning was much faster and the testing success rate was 85%. The dependence of the asymptotic success rate on the size of the training set is shown in Figure 7.

(b) *Real proteins*

(i) *Testing with non-homologous proteins*

We trained standard networks (13 input groups, local coding scheme, and 3 output units) with either 0 or 40 hidden units. The learning curves for the training and testing sets are shown in Figure 8. In all cases, the percentage of correctly predicted structures for both the training and testing sets rose quickly from the chance level of 33% to around 60%. Further training improved the performance of the networks with hidden units on the training set, but performance on the testing set did not improve but tended to decrease. This behaviour is an indication that memorization of the details of

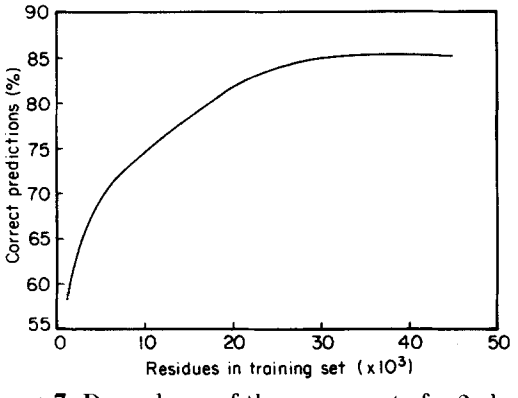

Figure 7. Dependence of the success rate for 2nd order artificial structures as a function of the training set size. The input encoding was a 2nd order conjunctive scheme.

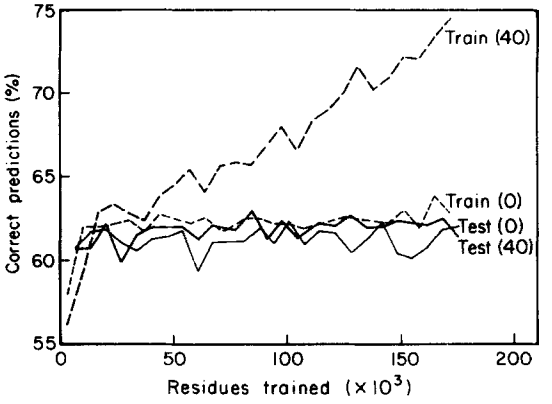

Figure 8. Learning curves for real proteins with testing on non-homologous proteins. Results for 2 networks are shown, one with no hidden units (direct connections between input and output units) and another with 40 hidden units. The percentage of correctly predicted secondary structure is plotted as a function of the number of amino acids presented during training.

the training set is interfering with the ability of the network to generalize. The peak performance for a network with 40 hidden units was $Q_3 = 62{\cdot}7\%$, with the corresponding $C_\alpha = 0{\cdot}35$, $C_\beta = 0{\cdot}29$ and $C_{coil} = 0{\cdot}38$. The performance with no hidden units is similar, as shown in Figure 8 and indicated in section (b) (iii), below.

The values of the weights for the network with no hidden units are given in Tables 13, 14 and 15 in the Appendix, and a graphical representation of these weights, called a Hinton diagram, is shown in Figure 9. The relative contribution to the secondary structure made by each amino acid at each position is apparent in this diagram. Physical properties of the amino acids can be correlated with their contributions to each form of secondary structure; in this way, hypotheses can be generated concerning the physical basis for secondary structure formation (see Fig. 9).

(ii) *Change of the weights in the network during training*

The average magnitude of the weights from each input group is plotted as a function of the input group and of time in Figure 10. The network had 17 input groups and 40 hidden units. The weights are largest in the centre and are approximately symmetric around the centre group. Over time, both the central peak and the flanks increase in size. This behaviour is similar to the previous experiments on artificial structures to which noise had been added to the data (Fig. 6) but unlike the behaviour of the weights when noise was not present (Fig. 4). This suggests that the weights from groups more than about eight residues from the central amino acid may not contribute information to the prediction of the secondary structure, even though the weights to these distant groups are large and increase with time during training. This conjecture was tested by varying the number of input groups and the results are reported below.

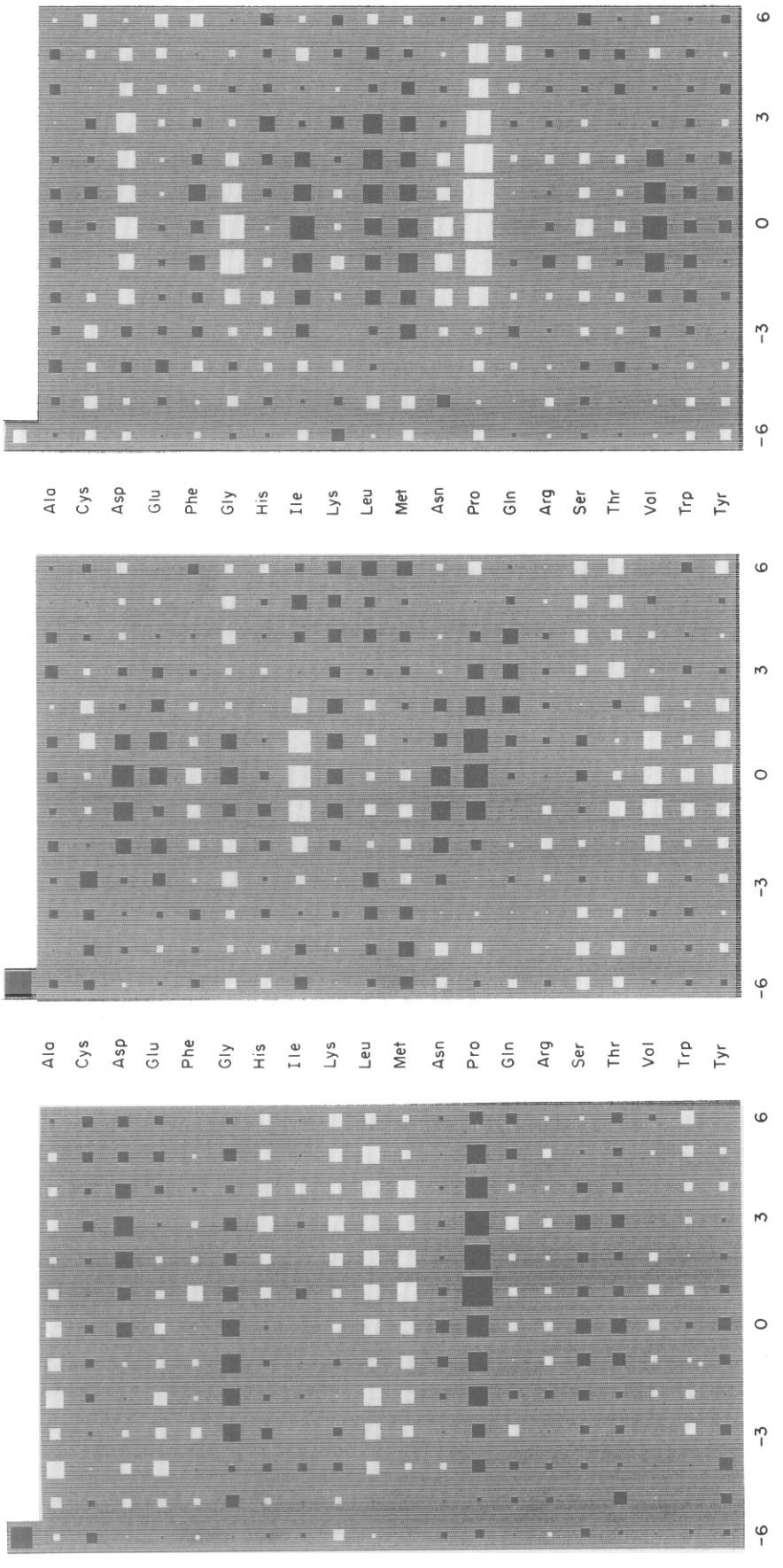

Figure 9. Hinton diagram showing the weights from the input units to 3 output units for a network with 1 layer of weights and a local coding scheme trained on real proteins. Three grey rectangular blocks show the weights to the 3 output units, each representing 1 of the 3 possible secondary structures associated with the centre amino acid. A weight is represented by a white square if the weight is positive (excitatory) and a black square if it is negative (inhibitory), with the area of the square proportional to the value of the weight. The 20 amino acid types are arranged vertically and the position of each of them in the 13-residue input window is represented horizontally and is numbered relative to the centre position. The weight in the upper left-hand corner of each large rectangle represents the bias of the output unit in eqn (4). The contribution to each type of secondary structure by amino acids at each position is apparent in this diagram. For example, proline is a strong α-helix breaker, while alanine, leucine and methionine are strong helix formers, especially when they are on the C-terminal side. Two basic amino acids, lysine and arginine, are helix formers when they are on the C-terminal side, while glutamate, an acidic amino acid, supports helical structure when it is on the N-terminal side. Isoleucine, valine, tryptophan and tyrosine are strong β-sheet formers and show no preferences toward the C-terminal or N-terminal side.

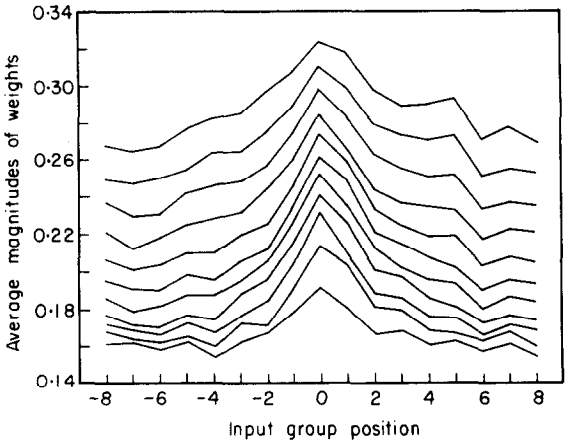

Figure 10. Values of the average magnitude of the weights from each weight group shown at several times during training on real proteins. The lowest curve represents the average magnitudes of the weights early in training.

Based on the observations with artificial structures that small randomly fluctuating weights were useless and could even interfere with the performance of the network, we systematically pruned small weights in one experiment. In a network with 17 input groups, 40 hidden units and 1 output group, we set all of the weights smaller than 0·15 to zero after every 12,000 amino acids were presented during training. We found that at the end of training, 60% of the weights were zero and the performance was slightly improved.

(iii) *Dependence on the number of hidden units*

Table 7 shows the surprising result that the peak performance on the testing set was almost independent of the number of hidden units although the learning rates of the training set (not shown) became slower as the number of hidden units decreased. Even more surprising, the testing success rate of a network with no hidden units was about the same as one with 40 hidden units, as shown in Figure 8. Furthermore, the training and testing performances of the network with no hidden units were indistinguishable.

Table 7

Dependence of testing success rate on hidden units

Hidden units	Q_3(%)
0	62·5
3	62·5
5	61·6
7	62·2
10	61·5
15	62·6
20	62·3
30	62·5
40	62·7
60	61·4

Dependence of the performance of the non-homologous testing set on the number of hidden units.

Table 8

Dependence of testing success rate on window size

Window size	Q_3(%)	MC_α	MC_β	MC_{coil}
21	61·6	0·33	0·27	0·32
17	61·5	0·33	0·27	0·37
15	62·2	0·35	0·31	0·38
13	62·7	0·35	0·29	0·38
11	62·1	0·36	0·29	0·38
9	62·3	0·33	0·28	0·38
7	61·9	0·32	0·28	0·39
5	60·5	0·28	0·26	0·37
3	57·7	0·22	0·20	0·30
1	53·9	0·11	0·14	0·17

Dependence of the performance of the non-homologous testing set on number of input groups. MC_α, MC_β and MC_{coil} are the maximum correlation coefficients during training, which may occur at different stages.

These results suggest that the common features in the training and testing proteins are all first order features and that all of the first order features learned from the training set that we used were common features. The higher order features (the information due to interactions between 2 or more residues) learned by the network were specific to each individual protein, at least for the proteins that were used. In a later section, we show that if the training set is too small then not all the first order features learned during training are common features.

(iv) *Dependence on the number of input groups*

We studied the dependence of testing success rate on the size of the input window using a standard network with 40 hidden units. The results shown in Table 8 indicate that when the size of the window was small the performance on the testing set was reduced, probably because information outside the window is not available for the prediction of the secondary structure. When the size of the window was increased, the performance reached a maximum at around 13 groups (6 on either size of the centre residue). For larger window sizes, the performance deteriorated, probably for the reason given in section (a) (iv), above. Similar results were obtained for networks without hidden units.

(v) *Dependence on size of the training set*

A standard network with 13 input groups and no hidden units was trained on training sets with different numbers of amino acids in them. The maximum performance of the network as a function of the training set size is presented in Figure 11. The maximum occurred after different training times in the different networks.

The maximum performance on the training set decreases with the number of amino acids in the training set because more information is being

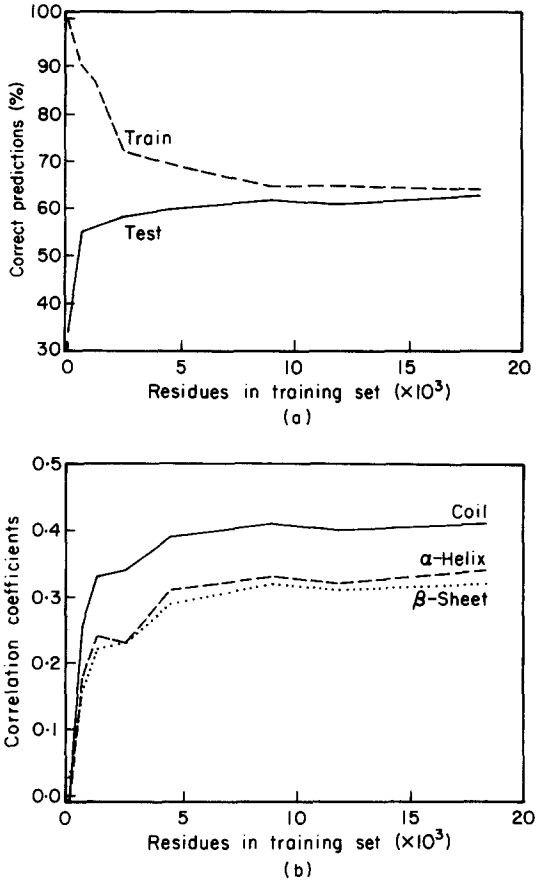

Figure 11. Dependence of the prediction accuracy on the size of the training set of non-homologous proteins. (a) Percentage correct for the training and testing sets. (b) Correlation coefficients for the testing set.

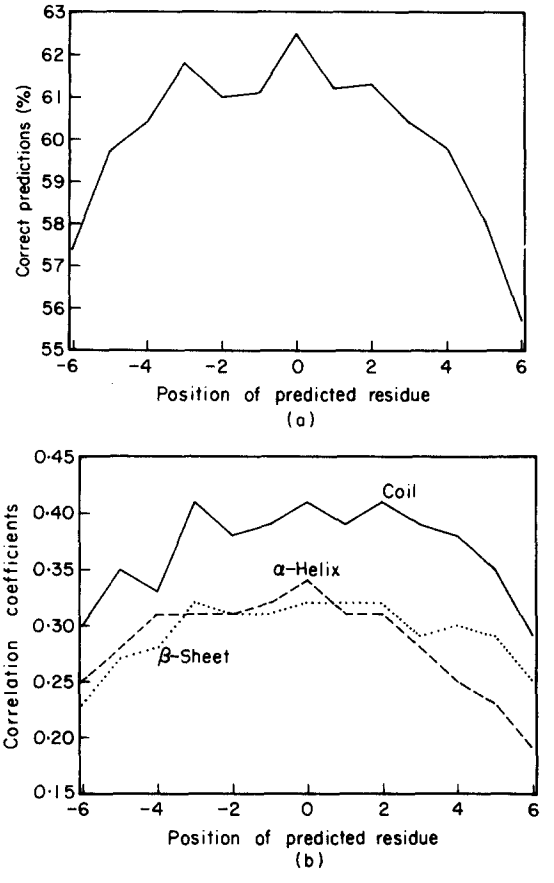

Figure 12. Dependence of the prediction accuracy on the position within a window of 13 amino acids. The position is indicated relative to the centre of the window, so that −2 refers to a network that is attempting to predict the secondary structure of the amino acid 2 positions toward the N-terminal from the central residue. (a) Success rate as a function of position. (b) Correlation coefficients as a function of position.

encoded in a fixed set of weights. The testing success rate, on the other hand, increases with size because the larger the training set, the better the network is able to generalize. When the training set is small, the network is able to "memorize" the details, but this strategy is not possible when the training set is large. Another conclusion from Figure 11 is that a further increase of the data set is unlikely to improve the performance of the network on the testing set.

(vi) *Relative importance of information on the N and C-terminal sides*

We trained a network with no hidden units and a window size of 13 to predict the secondary structure of the amino acid m positions away from the centre. There are 13 values of m ranging from −6 to 6, where a negative value indicates a position to the N-terminal side of centre. The maximum testing success rate and maximum correlation coefficients are shown in Figure 12. All curves are approximately symmetric around the centre and have broad maxima between −3 and +3. This result is consistent with about equal contributions from the information in the N-terminal and C-terminal sequences.

(vii) *Prediction near the N terminus*

Other methods for predicting the secondary structure are more accurate near the N terminus of most proteins (Argos *et al.*, 1976). In Table 9 the success rate for our method on the 25 amino acid N-terminal sequence is compared with the average success rate. The performance of our method on this segment is significantly higher, consistent with previous findings. Our method considers only local interactions, which suggests that local interactions are more important in determining the secondary structure at the N terminus of globular proteins, as proposed by other authors.

(viii) *Cascaded networks improve performance*

For a given input sequence, the output of the network is a three-dimensional vector whose components have values between 0 and 1. The secondary structure for the above networks was predicted by choosing the output unit with the largest value, as mentioned in Methods. However, information about the certainty of the prediction is

Table 9
Prediction of a short segment at the N-terminal end

Segment	Q_3(%)	C_α	C_β	C_{coil}
1st 20 residues	73·8	0·45 (62)	0·45 (69)	0·54 (209)
1st 25 residues	72·2	0·46 (91)	0·45 (84)	0·52 (250)
1st 30 residues	68·0	0·41 (117)	0·39 (111)	0·48 (282)
1st 40 residues	63·4	0·33 (167)	0·35 (156)	0·43 (352)
All but 1st 25	61·3	0·34 (758)	0·27 (664)	0·36 (1675)

The numbers in parentheses are the numbers of residues in the testing sets.

not exploited by this procedure. Neither is the information available in the correlations between neighbouring secondary structure assignments, since predictions are made one residue at a time. However, we can take advantage of this additional information by designing a second network.

The inputs to the second network were sequences of outputs from the first network, trained as described above. Hence, the input layer of the second network contained 13 groups with three units per group, each group representing the complete information about the secondary structure assignment derived from the first network. The first network was fixed while the second network was trained on the same set of training proteins as the first network. The average performance for two cascaded networks was $Q_3 = 64{\cdot}3\%$, $C_\alpha = 0{\cdot}41$, $C_\beta = 0{\cdot}31$ and $C_{coil} = 0{\cdot}41$ with 40 hidden units in both nets. This was our best result on the testing set of non-homologous proteins. Performance on each of the non-homologous proteins in the training set is given in Table 10. The weights for a second network without hidden units (whose input is from the first network in Tables 13 to 15) is given in Table 16.

The improvement provided by the second network is apparent in Figure 13, which compares the predictions made by the first and second networks. The second network "cleans up" the predictions of the first by joining short fragments of secondary structure and eliminating isolated assignments. The improvement was mainly in the regions of α-helix and coil, but not in regions of β-sheet.

Table 10
Results on non-homologous testing proteins

Protein	C_α	C_β	C_{coil}	Q_3(%)
1abp	0·33	0·31	0·23	61
1acx	—	0·28	0·28	65
1hmq	0·46	—	0·49	72
1ige	0·18	0·42	0·50	68
1nxb	—	0·49	0·43	71
1ppd	0·39	0·24	0·49	66
1pyp	0·32	0·34	0·48	73
2act	0·40	0·36	0·35	64
2alp	0·30	0·32	0·29	57
2cdv	0·47	0·25	0·38	71
2grs	0·41	0·30	0·44	64
2lhb	0·50	—	0·58	74
2sbt	0·26	0·36	0·34	66
3gpd	0·40	0·25	0·45	64
6api	0·34	0·27	0·32	52
Weighted average	0·41	0·31	0·41	64·3

Results of a 2-network cascade with 40 hidden units each for non-homologous testing set of proteins (Table 2).

(ix) *Methods that did not improve performance*

We experimented with many variations of the basic network, but none of them helped improve the performance on the testing set. The following methods were of little or no help (less than 1%):

(ix)(a) *Modification of the input representations*

The local input representation of the amino acids we used contains no information about their biophysical properties. We tried using distributed coding schemes representing charge, size, hydrophobicities, and other detailed information about the conformation of the side groups. In another attempt, we used the information measures of Robson (Garnier *et al.*, 1978) as part of the input representations. A second order conjunctive encoding was also used. We experimented with varying the input representations during the learning without success.

These physical properties are of known biophysical importance for determining the secondary structure. The failure to improve performance does not necessarily imply that the network is not capable of taking advantage of these properties; an alternative interpretation is that the network is already extracting all of the relevant information available from these properties. The failure of the second order conjunctive encoding proves that no second order common features about the secondary structure are present locally.

(ix)(b) *Modifications to the network architecture*

We examined a number of variations of the standard network architecture. We studied networks with up to seven output groups corresponding to a secondary structure prediction of up to seven contiguous amino acids. All sets of output for a given amino acid were averaged before making a prediction.

Many networks were studied that had altered connectivities: networks with two hidden layers; networks with direct connections between the input and output layers as well as through a layer of

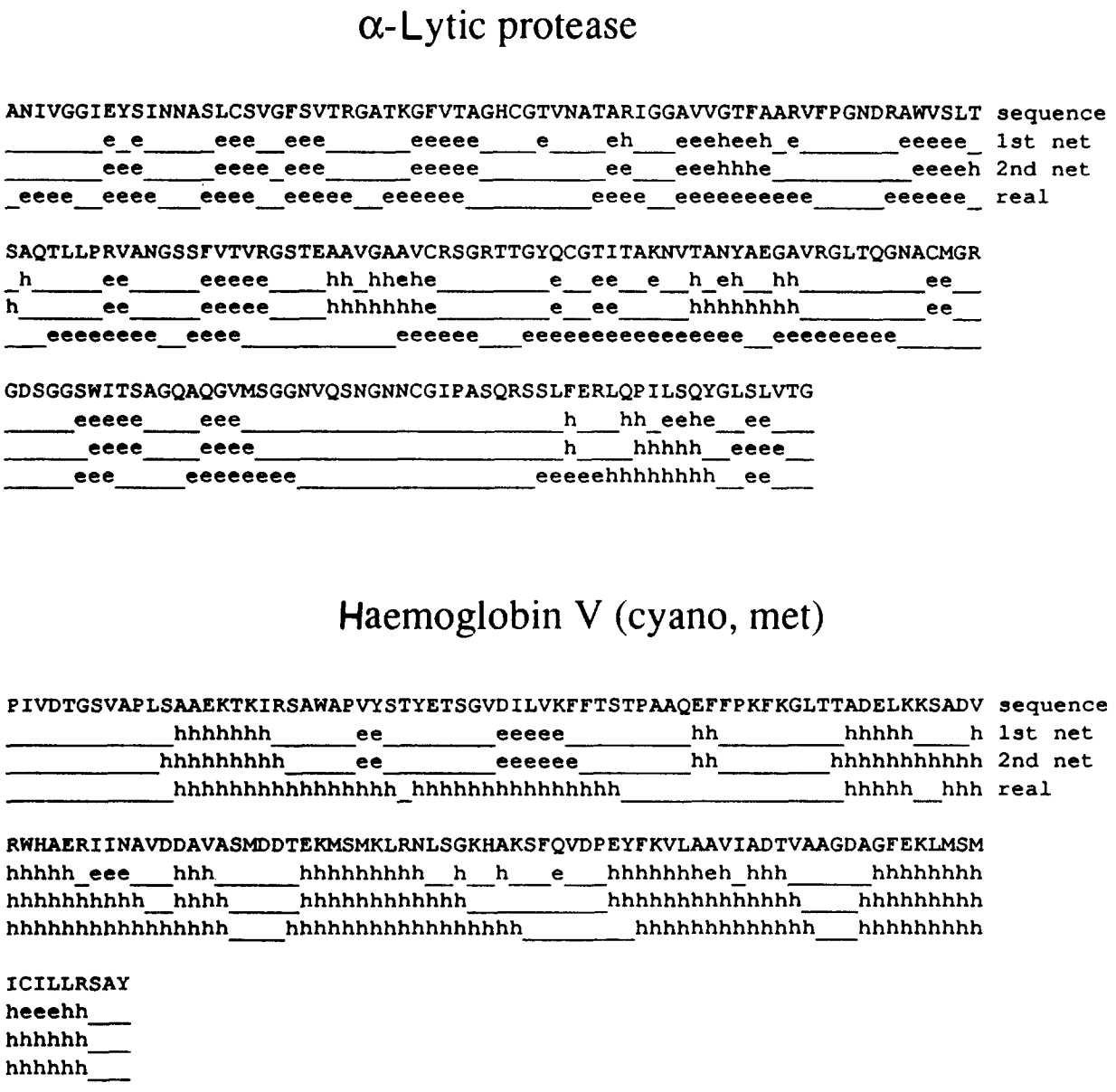

Figure 13. Comparison of the predictions for the secondary structure of haemoglobin V (21hb) and a α-lytic protease (2alp). The results are shown for 1 network and 2 cascaded networks. The 2nd network improved on the prediction of the 1st network for haemoglobin V, which was rich in α-helix, but slightly decreased the accuracy of the prediction for α-lytic protease, which contained a high proportion of β-sheet structure.

hidden units; recursive networks, in which the previous outputs were fed back to serve as extra input units.

Multiplicative synaptic weights were used that depended quadratically on pairs of input units. In particular, we used multiplicative units to represent global variables such as the position of the amino acid in the protein and the average hydrophobicity of the protein.

(ix)(c) *More detailed secondary structure predictions*

Kabsch & Sander (1983*b*) distinguish three types of helices, two types of β-structures and two types of turns, as well as the coil. We attempted to train a network on this finer classification scheme first, and then to collapse the assignments into the three final states.

(x) *Comparison with other methods*

The performance of our method for secondary structure prediction is compared with those of Robson and co-workers (Garnier *et al.*, 1978), Chou & Fasman (1978) and Lim (1974) in Table 11. The original measures of accuracy reported by these authors were based in part on the same proteins from which they derived their methods, and these proteins are equivalent to our training set. The performance of our networks with hidden units on the training set was as high as $Q_3 = 95\%$ after sufficiently long training. However, these methods should be compared on proteins with structures that were not used in or homologous with those in the training set. The results of testing these three methods on novel proteins is reported in Table V of

Table 11
Comparison of methods

Method		Q_3(%)	C_α	C_β	C_{coil}
Robson		53	0·31	0·24	0·24
Chou–Fasman		50	0·25	0·19	0·24
Lim		50	0·35	0·21	0·20
Neural	1 net	62·7	0·35	0·29	0·38
Network	2 nets	64·3	0·41	0·31	0·41

Comparison with other methods for predicting secondary structure on a non-homologous testing set of proteins (Table 2). Q_3 is the average success rate on 3 types of secondary structure and C_α, C_β and C_{coil} are the corresponding correlation coefficients for the α-helix, β-sheet and coil, respectively. Results are shown for a single network (1 net) or a 2-network cascade (2 nets).

Nishikawa (1983) and are listed in Table 11 with the performance of our networks on the non-homologous testing set of proteins.

The correlation coefficient introduced by Mathews (1975) is another measure of the quality of a prediction, one that takes into account over-prediction as well as underprediction. These parameters have been calculated by Nishikawa (1983) for previous methods and are listed in Table 11 with the correlation coefficients of our method. Our predictions are better than all previous methods for all secondary structure predictions. Our method has a success rate that is an absolute improvement of 11% and a relative improvement of 21% over the method of Robson *et al.* (Garnier *et al.*, 1978), which is the most reliable of other existing methods. The correlation coefficients of our method have a relative improvement of 32%, 29% and 41% for the α-helix, β-sheet and coil, respectively.

Our training and testing sets of proteins were different from those used to construct and test the previous methods. To determine how much of our improvement was due to this difference, we trained a new network using 22 of the 25 proteins found in Robson & Suzuki (1976) as the training set for a network. (Three of the proteins were missing from our database: carp myoglobin, horse cytochrome *c*, and adenylate cyclase. Deleting these proteins from our training set would decrease slightly the performance of the network, as indicated in Fig. 11.) Our testing set was a subset of those found in Table V of Nishikawa (1983). (The following 10 testing proteins were in our database: citrate synthease, erabutoxin B, prealbumin, γ-crystallin II, protease B, subtilisin inhibitor, phospholipase A_2, glutathione peroxidase, rhodanese and alcohol dehydrogenase.) The testing success rate of Robson's method on these ten proteins was 51·2% compared with 61·9% for our method with two cascaded networks. Thus, less than 1% of the 11% improvement in Table 11 can be attributed to differences in the training sets. The relatively small effect of the larger database available to use is consistent with the asymptotic slope of the dependence on training set size shown in Figure 11.

The improvement of our method over that of Robson *et al.* may seem puzzling, since they also use one layer of weights. The difference in performance can be attributed to the observation at the end of Methods, section (c), that first order features are stronger than first order statistics. The information measure in Robson's method depends only on the first order statistics. Therefore, exactly the same information measures would be obtained through the probabilities in Table 5 for the two sets of mappings shown in Table 4. However, two different sets of weights would be obtained by training two first order perceptrons on the two mappings separately. Thus, neural networks can distinguish mappings with same first order statistics but different first order features.

Levin *et al.* (1986) proposed an algorithm for determining secondary structures based on sequence similarity (We thank one of the referees for bringing this paper to our attention). In Table 3 of that paper, they showed that the prediction success rate for nine new proteins (corresponding to our testing proteins) is 63·4%. However, as pointed out by these authors, four out of their nine testing proteins had homologous counterparts in their database (corresponding to our training proteins), and these should be treated separately when the prediction accuracy of the method is assessed. The prediction success rate for these four proteins after the corresponding homologous proteins are removed from the database were given in the legend of their Table 3. The recalculated total success rate for the nine testing proteins falls to 59·7%, which is about 4·6% less than the success rate for our non-homologous testing set. However, this comparison may not be accurate, because the β-sheet content of their nine new proteins is about 17%, while it is 21% in our non-homologous testing set. Because β-sheet is the most difficult part of the structure to predict, we expect that the 4·6% improvement for our method is probably an underestimate. We cannot conduct a better-controlled comparison, as we did with Robson's method in the last section, because we do not have six of the nine proteins they used for testing (we used 6 homologous proteins in our database to estimate the proportion of the β-sheet in their testing proteins shown above). Another observation is that our method should be faster, because a set of weights obtained through training can be used for predicting secondary structures for all new proteins. The method of Levin *et al.* (1986), on the other hand, requires an exhaustive search of the whole database for every seven-amino acid sequence in the new protein.

(xi) *Testing with homologous proteins*

In all of the experiments described above, the testing set was carefully chosen not to have any homology with the proteins in the training set. The results were significantly different when homologies were present, as shown in Figure 14, in comparison with the results from the non-homologous testing set shown in Figure 8. The main difference is that, for the network with 40 hidden units, the performance on the testing set continued to

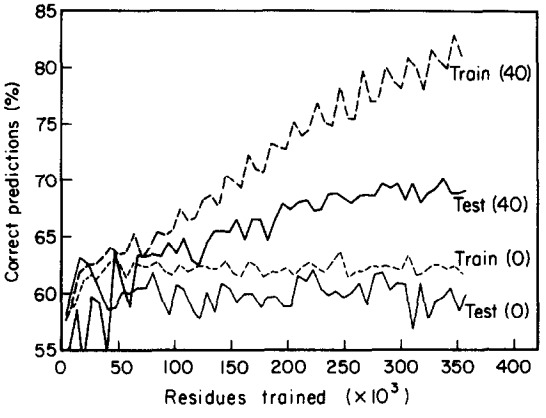

Figure 14. Learning curves for real proteins with testing on homologous proteins using the training and testing sets in Table 3. Results for 2 networks are shown, one with no hidden units (direct connections between input and output units) and another with 40 hidden units. The percentage of correctly predicted secondary structure is plotted as a function of the number of amino acids presented during training.

improve up to about $Q_3 = 70\%$, about 10% better than the network with no hidden units. With two cascaded networks, $Q_3 = 74{\cdot}4\%$. The hidden units were evidently able to exploit the additional information present in the homologous proteins.

We varied the composition of the training set and found that in most cases the best performance was obtained when the training set consisted only of proteins that had homologies with the testing protein. The results for 12 pairs of homologous proteins are shown in Table 12. For each pair of homologous proteins p_a and p_b, we trained a network on p_a and tested it on p_b. The testing success rate was almost always between the sequence homology and the structure homology. However, this is less than the success rate that is obtained by aligning the two proteins and assigning to the amino acids of p_b the corresponding secondary structures in p_a.

When the sequence homology between p_a and p_b is below 65%, the testing success rate can often be improved by adding other unrelated proteins to the training set, but the best result is still not as high as the structure homology.

We attempted to improve on our results for homologous proteins by using different input coding schemes. None of the coding schemes based on the physical properties of amino acids, on Robson's information measures, or on conjunctive encodings, were more effective than the simple local coding scheme with hidden units. Second order conjunctive encoding without hidden units gave results that were similar to a network with local input encoding and 40 hidden units.

4. Discussion

The new method for predicting the secondary structure of globular proteins presented here is a significant improvement over existing methods for non-homologous proteins and should have many applications. We have emphasized the distinction between training and testing sets, between homologous and non-homologous testing sets, and the balance of the relative amount of each type of secondary structure in assessing the accuracy of our method, and have provided objective measures of performance that can be compared with other methods. Tables 13 to 16 in the Appendix contain all the information needed to program our method.

However, the absolute level of performance achieved by our method is still disappointingly low. Perhaps the most surprising result was the conclusion that further improvement in local

Table 12
Results on homologous testing proteins

Homologous pairs Test	Train	Number of residues	Sequence homology(%)	Structural homology(%)	Q_3(%)
1azu	1aza	125	69	84	78
1lzt	1lzl	129	65	96	83
1pfc	1fc2	111	66	62	63
1ppd	2act	212	54	93	83
2gch	1tgs	237	46	87	70
1gfl	1fr2	70	71	94	99
1p2p	1bp2	124	83	91	90
2ape	2app	318	67	80	61
2rhe	1ig2	114	77	92	77
2sga	3sgb	181	65	91	76
3hhb	2dhb	287	85	91	89
5ldh	1ldx	333	71	86	68
Weighted average			68	87	76

Results of networks with hidden units on homologous proteins. The overall weighted correlation coefficients corresponding to $Q_3 = 76\%$ were $C_\alpha = 0{\cdot}70$, $C_\beta = 0{\cdot}58$ and $C_{coil} = 0{\cdot}54$. The sequence and structural homologies, defined as the percentage of identical amino acids or secondary structures between 2 proteins, were estimated by manual inspection.

methods for predicting the secondary structure of non-homologous proteins is unlikely, based on known structures. The fact that networks with no hidden units performed as well as networks with hidden units on the non-homologous training set suggests that there are little or no second or higher order features locally available in the training set to guide the prediction of secondary structure. Could this be due to a database of insufficient size or failure of the network to detect higher order common features?

Two lines of evidence argue against these possible explanations. First, the dependence of the performance on the size of the training set suggests that the addition of more protein structures to the training set will not significantly improve the method for non-homologous proteins. Second, we can definitively conclude that no second order features are present in the database from our experiments with conjunctive input representations of the amino acids (which make 2nd order features available as 1st order features to the output layer). The use of hidden units, which are capable of exploiting higher order features in the data, did not improve the performance either. Experiments with second order artificial structures suggests that our method was capable of detecting second order features. All of these experiments are consistent with the hypothesis that little or no information is available in the data beyond the first order features that have been extracted.

However, it is still possible that our method may not extract all of the information available as first order features from the training set. An estimate for the maximum obtainable accuracy of local methods such as ours can be obtained from our study of artificial structures. We stochastically generated artificial structures that had only information in the first order statistics, as estimated by Garnier *et al.* (1978) from real proteins. The profile of the magnitudes of the weights from different input groups and the increase in the size of the weights was similar to that observed for real proteins, but only when 30% noise was added to artificial structures. This suggests that a theoretical limit of 70% can be obtained with local methods, which is close to our present performance of 64·3%. The pattern recognition method that we used is not effective when the information contained in the statistics of the training set is global. If further statistical analysis of the database of protein structures confirms our results, then a significant fraction of the local secondary structure depends on influences outside the local neighbourhood of an amino acid and significant improvements for non-homologous proteins would require better methods for taking into account these long-range effects.

The prediction accuracy of networks tested with homologous proteins is much better than that for non-homologous proteins. Other methods are also much better when tested with homologous proteins. For a highly homologous testing protein, our best results were obtained by training a network solely with the homologous protein, but the success rate is almost always less than the structure homology. This is not surprising, since a single protein contains little information about amino acid substitutions that do not alter the secondary structure. With a much larger database of homologous proteins, it should be possible for a network to discover the equivalence classes of amino acids in different contexts.

Appendix

The weights in Tables 13 to 15 can be used to program a network that predicts the secondary structure for globular proteins. Each row represents one of the input groups, and each column represents one of the amino acids. There are no hidden units in the network, so each weight is for a connection from one of the 273 input units to one of the three output units. Note that a separate unit in each group is dedicated to the spacer (which appears in the

Table 13
Weights for α-helix

Amino acid	Window position −6			−3			0			3			6
Ala	0·12	0·26	0·64	0·29	0·68	0·34	0·57	0·33	0·13	0·31	0·21	0·18	−0·08
Cys	−0·25	−0·15	0·03	−0·05	−0·15	−0·18	−0·15	−0·03	−0·09	−0·26	−0·12	−0·29	−0·25
Asp	0·01	0·15	0·33	0·11	−0·02	0·06	−0·46	−0·44	−0·71	−0·81	−0·58	−0·32	−0·24
Glu	−0·02	0·21	0·51	0·28	0·44	0·20	0·26	0·21	0·13	−0·06	−0·23	−0·25	−0·19
Phe	0·05	0·12	−0·03	0·24	0·06	0·15	0·03	0·48	0·15	0·10	−0·06	0·05	0·00
Gly	−0·02	−0·37	−0·09	−0·67	−0·73	−0·88	−0·71	−0·46	−0·39	−0·42	−0·15	−0·40	−0·10
His	−0·06	0·10	−0·23	−0·26	−0·14	−0·09	−0·05	0·27	0·32	0·51	0·37	0·28	0·29
Ile	−0·07	−0·03	−0·22	0·00	−0·08	−0·03	0·00	−0·33	0·00	−0·15	0·31	−0·03	−0·01
Lys	0·26	0·12	−0·17	−0·19	0·03	−0·11	0·16	0·23	0·37	0·47	0·28	0·41	0·45
Leu	0·05	−0·02	0·41	0·47	0·61	0·20	0·48	0·57	0·50	0·56	0·70	0·62	0·28
Met	0·00	0·00	0·13	0·27	0·39	0·43	0·41	0·79	0·63	0·58	0·61	0·21	0·11
Asn	−0·10	−0·03	0·09	−0·04	−0·09	−0·33	−0·36	−0·19	−0·07	−0·10	−0·04	−0·03	−0·08
Pro	−0·19	−0·08	−0·43	−0·34	−0·76	−0·81	−1·12	−1·86	−1·40	−1·33	−1·03	−0·84	−0·42
Gln	−0·03	−0·13	−0·23	0·26	−0·15	0·01	0·15	0·19	0·12	0·41	0·13	−0·27	−0·28
Arg	0·04	−0·14	−0·10	−0·03	−0·22	0·22	0·23	0·10	0·08	0·18	0·07	0·21	0·05
Ser	−0·19	0·01	−0·10	−0·17	−0·26	−0·35	−0·47	−0·23	−0·28	−0·49	−0·28	−0·05	0·07
Thr	−0·04	−0·34	−0·07	−0·20	−0·10	−0·37	−0·54	−033	−0·21	−0·44	−0·25	−0·16	−0·33
Val	−0·03	0·02	−0·01	−0·01	0·12	0·13	0·31	0·24	0·17	−0·01	0·00	0·06	−0·13
Trp	−0·06	−0·01	−0·02	0·25	0·20	0·07	−0·10	0·15	0·02	0·14	0·21	0·32	0·36
Tyr	−0·14	−0·29	−0·38	−0·30	−0·04	−0·31	−0·35	−0·19	−0·10	−0·08	0·16	0·11	0·00
	−0·12	−0·15	−0·52	−0·58	−0·64	−0·37	−0·03	−0·47	−0·77	−0·66	−0·56	−0·22	0·24

Tables 13 to 15 show weights for a 1st network without hidden units that predicts secondary structure. Sequences of 13 amino acids are inputs and the structure of the centre residue is the output. The biases for the output units are −1·08 for α-helix −1·50 for β-sheet and 0·41 for coil.

Table 14
Weights for β-sheet

Amino acid	Window position −6			−3			0			3			6
Ala	−0·18	−0·01	−0·19	−0·14	−0·31	−0·10	−0·25	−0·26	0·05	−0·44	−0·31	−0·02	−0·06
Cys	−0·26	−0·27	−0·29	−0·64	−0·06	0·13	0·13	0·47	0·36	0·13	−0·11	−0·02	−0·19
Asp	0·05	−0·09	−0·06	−0·10	−0·54	−0·89	−1·01	−0·55	−0·11	−0·20	0·13	0·11	0·24
Glu	−0·06	0·09	−0·10	−0·39	−0·52	−0·34	−0·62	−0·75	−0·35	−0·28	−0·05	0·10	−0·04
Phe	−0·18	−0·12	−0·32	0·08	0·24	0·36	0·48	0·20	0·20	−0·13	−0·04	−0·03	−0·33
Gly	0·23	0·13	0·19	0·46	0·37	−0·45	−0·72	−0·56	0·14	0·08	0·45	0·38	0·17
His	0·24	0·22	−0·16	−0·04	−0·32	−0·34	−0·16	−0·04	0·02	0·09	−0·06	−0·09	0·19
Ile	−0·42	−0·27	−0·08	0·16	0·57	0·95	1·10	0·94	0·47	−0·04	−0·25	−0·48	−0·20
Lys	0·03	0·08	−0·09	0·04	−0·29	−0·46	−0·59	−0·55	−0·51	−0·33	−0·44	−0·39	−0·43
Leu	−0·23	−0·25	−0·42	−0·57	0·09	0·32	0·23	0·25	0·32	−0·12	−0·44	−0·26	−0·46
Met	−0·42	−0·57	−0·38	0·24	0·29	0·43	0·32	−0·05	−0·10	−0·21	−0·28	−0·14	−0·52
Asn	0·28	0·41	0·02	−0·27	−0·53	−0·89	−0·77	−0·34	−0·40	0·05	0·06	0·03	0·10
Pro	−0·13	0·26	0·05	0·02	−0·31	−0·91	−1·24	−1·28	−0·79	−0·48	−0·29	−0·04	0·37
Gln	0·21	0·01	0·02	−0·11	0·07	−0·04	−0·12	−0·33	−0·67	−0·58	−0·47	−0·17	−0·04
Arg	−0·13	0·02	0·03	0·14	0·25	0·19	−0·02	−0·09	−0·11	−0·13	−0·10	0·04	0·02
Ser	0·41	0·44	0·25	−0·12	0·11	−0·12	−0·31	−0·28	0·03	0·27	0·34	0·41	0·43
Thr	0·33	0·35	0·22	0·00	0·03	0·49	0·17	0·08	−0·15	0·47	0·27	0·36	0·50
Val	−0·07	−0·09	−0·15	0·29	0·48	0·76	0·69	0·67	0·58	0·06	0·11	−0·18	0·00
Trp	−0·10	−0·15	−0·19	−0·10	0·15	0·34	0·45	0·22	0·09	−0·22	−0·08	−0·01	−0·32
Tyr	−0·10	0·15	0·05	0·18	0·29	0·42	0·77	0·53	0·34	−0·11	0·06	−0·08	0·35
	0·21	−0·23	−0·32	−0·50	−0·71	−0·61	0·03	−0·58	−0·32	−0·10	0·06	−0·25	−0·12

See Table 13.

Table 15
Weights for coil

Amino acid	Window position −6			−3			0			3			6
Ala	−0·05	−0·19	−0·43	−0·19	−0·25	−0·27	−0·42	−0·24	−0·14	0·01	−0·30	−0·23	0·08
Cys	0·30	0·41	0·19	0·42	0·18	0·00	−0·18	−0·38	−0·09	−0·31	0·03	0·19	0·37
Asp	0·15	0·09	−0·31	−0·27	0·60	0·54	0·95	0·65	0·66	0·78	0·44	0·34	0·04
Glu	−0·02	−0·20	−0·41	−0·22	−0·12	−0·12	−0·09	0·07	0·06	0·09	0·18	0·28	0·36
Phe	0·09	0·07	0·25	−0·31	−0·29	−0·47	−0·39	−0·61	−0·25	−0·20	0·11	−0·02	0·34
Gly	−0·14	0·28	−0·21	0·17	0·09	1·14	1·24	0·85	0·36	0·14	−0·12	0·14	−0·02
His	−0·07	−0·19	0·21	0·17	0·42	0·18	0·05	−0·21	−0·31	−0·56	−0·20	−0·22	−0·45
Ile	0·26	−0·06	0·29	−0·34	−0·54	−0·74	−1·17	−0·65	−0·51	−0·09	−0·07	0·42	0·09
Lys	−0·42	−0·20	0·33	0·00	0·14	0·45	0·09	0·17	−0·14	−0·43	0·06	−0·15	−0·27
Leu	0·04	0·34	−0·10	−0·22	−0·55	−0·54	−0·69	−0·80	−0·80	−0·81	−0·18	−0·36	0·24
Met	0·25	0·45	−0·01	−0·53	−0·47	−0·76	−0·86	−0·71	−0·56	−0·49	−0·44	−0·19	0·16
Asn	0·00	−0·38	0·00	0·17	0·61	0·71	0·81	0·45	0·35	−0·11	−0·12	0·06	−0·06
Pro	0·31	0·04	0·28	0·14	0·89	1·40	1·77	2·27	1·59	1·14	0·77	0·78	0·16
Gln	−0·08	0·04	0·14	−0·29	0·09	−0·08	−0·01	0·01	;·11	−0·13	0·24	0·47	0·48
Arg	0·06	0·17	0·06	−0·07	0·12	−0·40	−0·23	−0·04	0·21	−0·13	−0·09	−0·20	−0·01
Ser	−0·11	−0·23	−0·23	0·22	0·24	0·40	0·63	0·33	0·32	0·13	−0·09	−0·29	−0·35
Thr	−0·06	−0·02	−0·26	0·10	0·16	−0·10	0·29	0·13	0·21	−0·02	−0·27	−0·30	−0·04
Val	0·04	0·05	−0·10	−0·33	−0·45	−0·86	−1·32	−0·99	−0·70	−0·11	−0·06	0·29	0·18
Trp	0·19	0·16	0·15	−0·15	−0·44	−0·46	−0·37	−0·44	−0·17	−0·20	−0·09	−0·18	−0·06
Tyr	0·33	0·22	0·09	−0·02	−0·19	−0·05	−0·41	−0·49	−0·35	0·10	−0·25	0·07	−0·20
	−0·33	0·01	0·54	1·00	1·04	0·76	−0·21	0·84	1·05	0·41	0·49	0·24	−0·20

See Table 13.

Table 16
Weights in the second network

α	Window position −6			−3			0			3			6
h	0·09	0·04	0·52	0·36	0·30	0·35	0·73	0·60	0·33	0·57	0·09	0·29	0·12
e	−0·04	−0·11	−0·26	−0·32	−0·30	−0·73	−0·81	−0·50	−0·55	−0·46	−0·24	0·13	−0·19
–	0·19	−0·03	0·10	−0·09	−0·19	−0·48	−0·97	−0·49	−0·21	0·12	0·16	0·12	−0·20

β	Window position −6			−3			0			3			6
h	−0·56	−0·33	−0·09	−0·11	−0·51	−0·73	−0·50	−0·39	−0·10	−0·18	0·02	−0·25	−0·37
e	0·09	−0·08	0·42	0·44	0·64	1·15	1·58	0·78	0·46	0·08	−0·03	0·01	0·28
–	0·11	0·15	0·07	−0·01	−0·02	−0·28	−1·12	−0·56	−0·20	−0·10	0·13	−0·06	0·11

Coil	Window position −6			−3			0			3			6
h	0·04	0·15	−0·32	−0·33	−0·02	−0·15	−0·58	−0·53	−0·36	−0·37	−0·17	−0·08	−0·04
e	−0·03	0·17	−0·27	−0·17	−0·36	−0·83	−1·40	−0·60	0·02	0·31	0·09	−0·08	0·02
–	−0·23	0·14	−0·09	−0·18	0·09	0·60	1·42	0·60	0·28	0·08	−0·37	−0·05	0·17

Weights for 2nd network without hidden units in a 2-network cascade. The sequence of 13 outputs from the 1st network are inputs to the 2nd network, whose output is the corrected secondary structure of the centre amino acid. The biases for the output are −0·19 for the α-helix, −0·73 for the β-sheet and −0·04 for the coil. Performance of the network on the testing set of non-homologous proteins (Table 2) was $Q_3 = 64$ °C%, where $C_\alpha = 0{\cdot}36$, $C_\beta = 0{\cdot}31$ and $C_{coil} = 0{\cdot}42$.

window only when the leading or trailing edge of the protein is present).

The weights in Table 16 can be used to program the second network in a two-network cascade. The input to the second network is the value of the three output units from the first network given in Tables 13 to 15. The overall performance of these cascaded networks is $Q_3 = 64\%$, 0·3% less than the figure quoted in Table 11, which was based on networks that had 40 hidden units.

We thank Dr Kevin Ullmer for helping with the database and for many discussions during the course of the research. Drs Carl Pabo and Richard Durbin suggested important improvements in the presentation. Drs Warner Love, Richard Cone and Evangelos

Moudrianakis provided helpful advice on many aspects of protein structure. We are grateful to Paul Kienker for discussions and the use of his network simulator. T.J.S. was supported by a Presidential Young Investigator Award (NSF BNS-83-51331).

References

Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. (1985). *Cong. Sci.* **9**, 147–169.

Argos, P., Schwartz, J. & Schwarz, J. (1976). *Biochim. Biophys. Acta*, **439**, 261–273.

Chou, P. Y. & Fasman, G. D. (1978). *Advan. Enzymol.* **47**, 45–148.

Garnier, J., Osguthorpe, D. J. & Robson, B. (1978). *J. Mol. Biol.* **120**, 97–120.

Kabsch, W. & Sander, C. (1983*a*). *FEBS Letters*, **155**, 179–182.

Kabsch, W. & Sander, C. (1983*b*). *Biopolymers*, **22**, 2577–2637.

Karplus, M. (1985). *Ann. N.Y. Acad. Sci.* **439**, 107–123.

Levin, J. M., Robson, B. & Garnier, J. (1986). *FEBS Letters*, **205**, 303–308.

Levitt, M. (1983). *Cold Spring Harbor Symp. Quant. Biol.* **47**, 251–262.

Lim, V. I. (1974). *J. Mol. Biol.* **88**, 873–894.

Mathews, B. W. (1975). *Biochim. Biophys. Acta*, **405**, 442–451.

Minsky, M. & Papert, S. (1969). *Perceptrons*, MIT Press, Cambridge MA.

Nishikawa, K. (1983). *Biochim. Biophys. Acta*, **748**, 285–299.

Robson, B. & Pain, R. H. (1971). *J. Mol. Biol.* **58**, 237–259.

Robson, B. & Suzuki, E. (1976). *J. Mol. Biol.* **107**, 327–356.

Rosenblatt, F. (1959). In *Mechanisation of Thought Processes*, vol. 1, pp. 421–456, HM Stationery Office, London.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). In *Parallel Distributed Processing*, vol. 1, pp. 318–362, MIT Press, Cambridge, MA.

Scheraga, H. A. (1985). *Ann. N. Y. Acad. Sci.* 439, 170–194.

Sejnowski, T. J. & Rosenberg, R. R. (1987). *Compl. Syst.* **1**, 145–168.

Staden, R. (1982). *Nucl. Acids Res.* **10**, 2951–2961.

Taylor, W. R. (1986). *J. Mol. Biol.* **188**, 233–258.

Webster, T. A., Lathrop, R. H., & Smith, T. F. (1987). *Biochemistry*, **26**, 6950–6957.

Widrow, R. M. & Hoff, M. E. (1960). In *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record*, part 4, pp. 96–104. IRE, New York.

Edited by S. Brenner