Studies in Protein Secondary Structure Prediction with Neural Network Models

Steven John Hayward

Submitted for the degree of

Doctor of Philosophy

Department of Molecular Biology
University of Edinburgh

February 1991

DECLARATION

All the work in this thesis is my own work, except where otherwise stated.

ACKNOWLEDGMENTS

This work was funded by the Science and Engineering Research Council.

Firstly I'd like to thank my supervisor John Collins. I also wish to thank Andrew Coulson for helpful discussions and Andrew Lyall for his support. Amongst the neural network people in the Physics Department I want to thank Gareth Richards who let me use his program, Frank Smieja, Elisabeth Gardner and Bruce Forrest. In addition my thanks to Tom Smith, Sarah McQuay and James Crook. I also want to thank my sister, Amanda, for typing some of this thesis during her lunch breaks and finally many thanks to my Mum and Dad, and Grandma for their support during the more stressful periods of this work.

ABSTRACT

The aim of this work was to predict protein secondary structure using neural network models. Initially a Hopfield network was used but abandoned in favour of a layered network trained using the back propagation algorithm. In the early stages of this work an exploration of the many different approaches to this problem was undertaken. These included attempts to predict boundaries between secondary structures, the secondary structures of individual residues, and the secondary structures of sequences wholly within a particular secondary structure. Results indicated the latter to be the best approach to continue with. In addition two coding schemes were investigated: a coding scheme based on occurrences of pairs of residues and one based on the positions of residues. It was found that this positional coding scheme was the natural coding scheme for this problem. On segments of whole alpha-helix and whole non-alpha-helix 10 residues in length a prediction success of around 80% with a correlation coefficient of 0.52 was achieved with the positional coding scheme. On whole proteins, where predictions are made for individual residues, alpha-helix prediction drops to 73% with a correlation coefficient of 0.34. The relative predictability of alpha-helices of above and below average accessibility was also investigated showing that those of above average accessibility are more predictable than those with below average accessibility. The main body of this work concerns the apparent limit on predictability of alpha-helices. It was found that test set prediction did not depend on the number of hidden nodes. In fact, a single layer network performed as well as those with hidden nodes showing that the problem is basically linearly separable. In addition, prediction success plateaus well below that of perfect prediction success. During training, test set prediction is seen to peak. The decrease in prediction success was found to be due to non-alpha-helix sequences that the network was unable to distinguish from real alpha-helix sequences. These regions of non-alpha-helix were shown to occur adjacent to actual alpha-helices with statistical significance. It is proposed that potential alpha-helices are disrupted by global constraints during the formation of tertiary structure. The

effect of window size was also investigated as was beta-sheet prediction, but this was found to be limited by the small number of examples available with our approach. However, its distribution in the input space in relation to alpha-helix and coil was determined.

TABLE OF CONTENTS

INTRODUCTION

It is not possible to understate the range of different functions in living things for which proteins are essential. They include catalytic activity in the case of enzymes, transport and storage of ions, coordinated motion as in muscles, mechanical support as in bone, immune protection, generation and transmission of nerve impulses, and control, growth and differentiation in cells.

In order to have such a diverse range of functions, proteins must have an equally diverse range of structures. How these structures are formed from a linear chain of amino acids of which a basic set of only twenty are found in proteins, remains a largely unanswered question.

Proteins are made on ribosomes in the cell. Here the peptide bonds are synthesized in a stepwise fashion. It is generally thought that for globular proteins the linear chain thus synthesized folds under physical forces determined solely by the amino acid sequence itself. Denaturing a globular protein by increasing the concentration of denaturing agent in its surrounding solution and then reducing its concentration has shown that after a sufficient reduction the protein regains its previous function (Anfinsen, 1973). This means that the protein's final folded structure is determined by the amino acid sequence - its primary sequence, and that no external folding agents are necessary. Despite some recently emerging evidence that this may not always be the case (Ellis, 1990) it is thought to be generally true. The consequence of this is that hidden in the primary sequence of a protein is information on its tertiary structure and, thus, function. In other words, the incredible diversity in protein function results from the number of possibilities in combining the basic set of twenty amino acids in a linear chain. To be able to determine tertiary structure from primary sequence would not be

so important if it were easier to determine the structure of a protein rather than its sequence. Determination of a protein's structure, however, is a long and complicated process which involves the purification of the protein, the growing of a crystal, X-ray diffraction and the translation of the resulting diffraction pattern to the protein structure; a far from trivial task. In comparison to this the determination of protein sequence is less laborious. Now most protein sequence comes from DNA sequence which is easier to sequence than protein directly and is emerging at an incredible rate destined to increase in the future in the advent of projects such as the Human Genome Project. This makes the ability to be able to interpret sequence data into protein structure and function all the more important.

1 PROTEIN STRUCTURE AND FOLDING

1.1 THE AMINO ACIDS

The generic formula for an amino acid is NH2-CHR-COOH. Its chemical identity is conferred upon it by its side chain, denoted by R, which extends from the alpha carbon atom C_α (see figure 1.1). All the amino acids are L-isomers apart from glycine which is achiral. The peptide bond links successive amino acids in the chain and has a partial double bond character with very little flexibility. In fact the six atoms linked by the peptide bond lie in a plane and are collectively called the peptide unit. It is the two single main chain bonds at the C_α about which the peptide units rotate that give proteins the required flexibility they obviously have in order to take on so many different conformations. The dihedral angle ψ is the angle of rotation about the C-C_α bond, the dihedral angle ϕ about the C_α-N bond. Flexibility is hindered however by the side chains. In the case of glycine whose side chain is a single hydrogen atom, steric hindrance from the side chain is small.

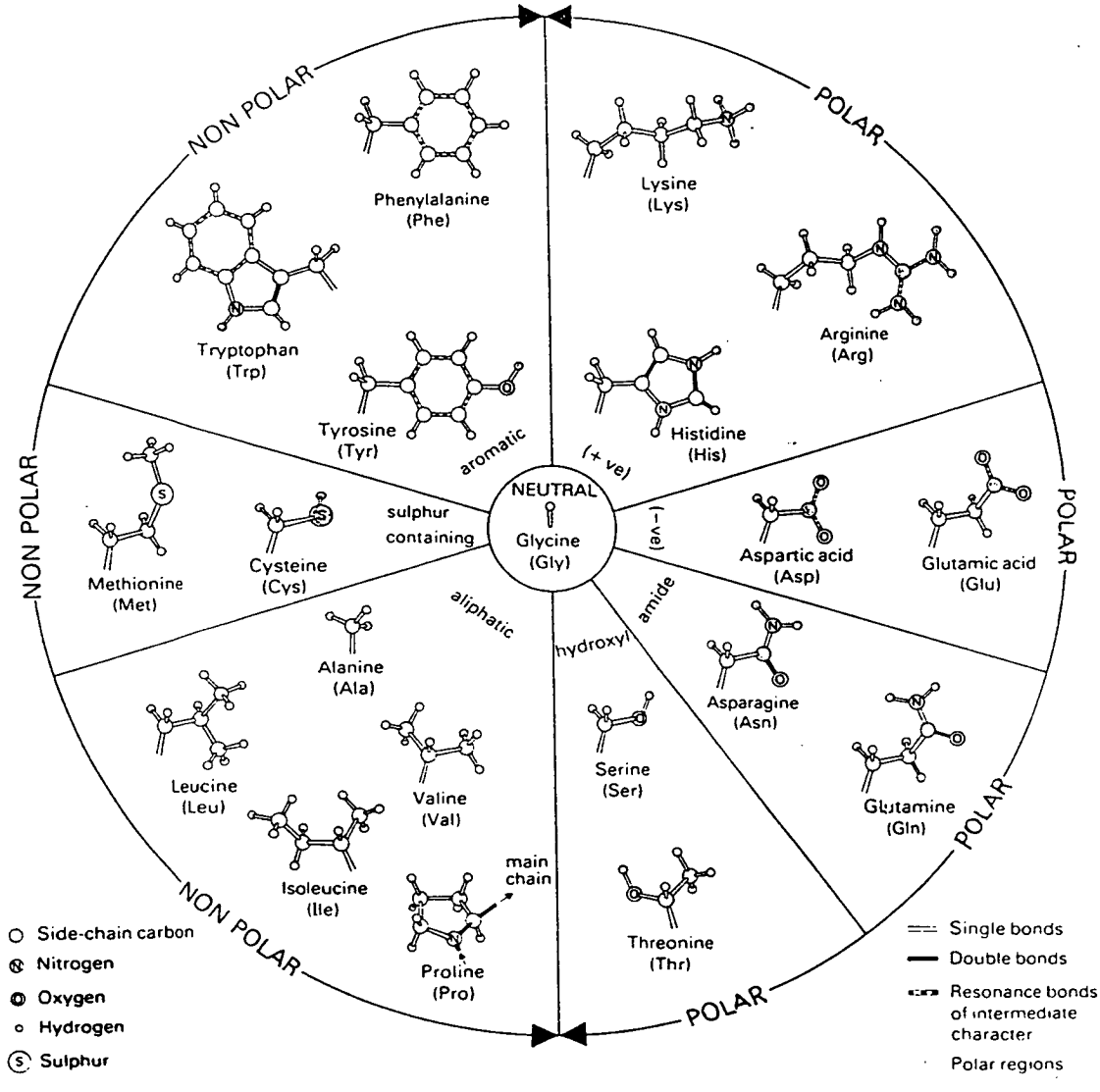


Figure 1.1

Amino acids grouped according to physical properties of the side chain R. Only the side chains are shown. (After Rees and Sternburg, 1984)

For proline, whose side chain curls back and binds to the peptide nitrogen[1] ϕ is constrained to -60°± 20°. For all the remaining amino acids the range of conformational freedom is largely restricted by their C_β atom, the remaining side chain atoms having a comparatively small effect (see figures 1.2 and 1.3). The other main degree of freedom is the C_α side chain bond whose dihedral angle is denoted by χ. Specification of every ψ and ϕ angle in a polypeptide chain determines completely the path of the main chain and consequently along with the χ angles, the relative positions of the side chains. However variations in the χ angles will have a small effect on the relative side chain positions in comparison to variations in ψ and ϕ. Knowledge of the ψ and ϕ angles, therefore, should be sufficient for a possible attempt to be made at the determinination of the protein's function.

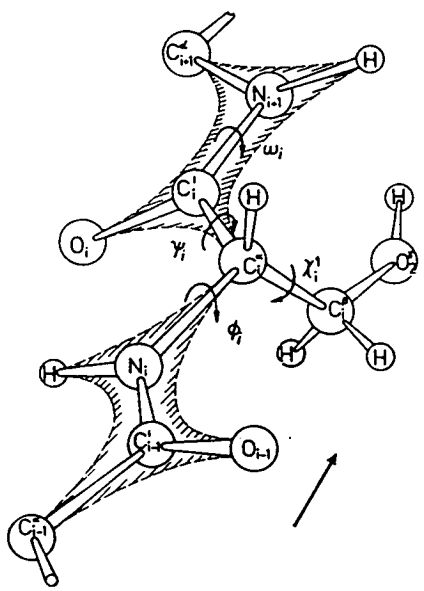
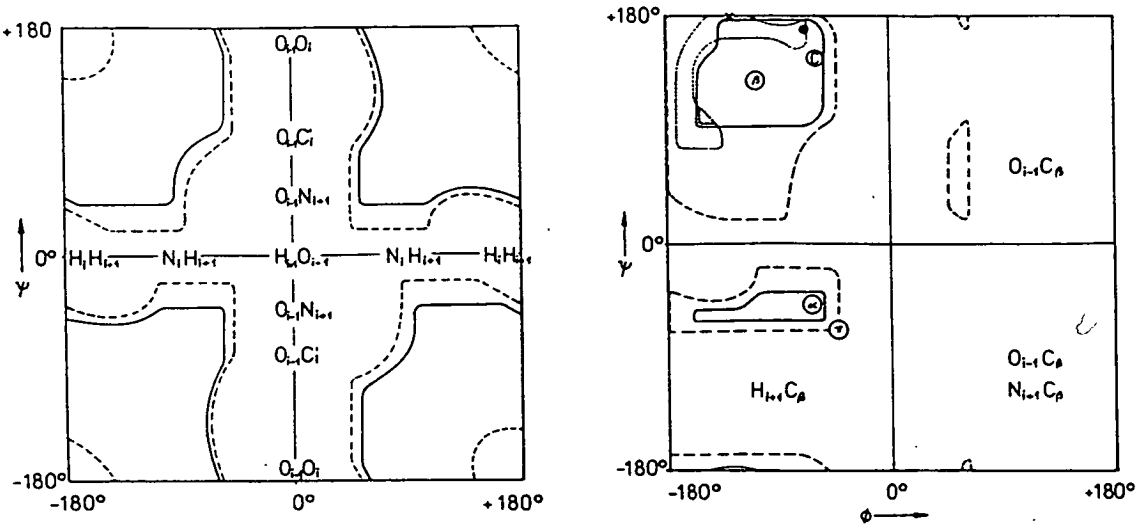


Figure 1.2

Definition of dihedral angles in a polypeptide chain. A serine residue is shown to illustrate the side chain dihedral angle. (After Schulz and Schirmer, 1978)

[1]For this reason proline is an imino acid.



(a) (b)

Figure 1.3

Ramachadran map showing allowed main chain dihedral angles determined using hard sphere models. Map (a) is for glycine which has no side chain. Map (b) is for amino acids with C_β atoms. (After Ramachandran and Sasisekharan, 1968)

1.2 SECONDARY STRUCTURE

Secondary structures refer to regular well-defined local structures of the main chain found in proteins. They are stabilized by hydrogen bonds between the carbonyl oxygen of one peptide and the amide hydrogen of another. There exist three main secondary structures. They are:

1)The alpha-helix (see figure 1.4). This was predicted by Pauling and Cory (Pauling and Cory, 1951) before its eventual discovery and is the most abundant of a number of different helical structures. It is identified by a hydrogen bond between the carbonyl oxygen of peptide i and the amide hydrogen of peptide i+4 and has 3.6 residues per turn. The allowed main chain dihedral angles for alpha-helix are indicated by the α label in figure 1.3. Its abundance suggests it must be a rather stable structure and its stiffness lends a certain amount of rigidity to a protein. The distribution of lengths of the alpha-helix has peaks at 7, 11 and 15 residues with an average of 17. Proline introduces a kink into alpha-helices resulting in a 20° change in direction. Virtually all alpha-helices are right handed with their side chains pointing away from the helix centre. Left handed alpha-helices are disfavoured by side chain interactions and have not yet been observed. This asymmetry between left and right handed alpha-helices is a direct consequence of the amino acids being L-isomers.

2)The beta-pleated-sheet (see figure 1.5). This structure is the most extended structure found in proteins. Being extended, the main chain hydrogen bonds cannot be made between members of the same strand. Hydrogen bonds are made between parallel or anti-parallel running strands so holding them together to form the sheet structure. The pleat arises from the tetrahedral angle at the C_α atom. The side chains point approximately perpendicular to the sheet and alternately to either side. Again the allowed dihedral angles for this structure are

indicated by the β label in figure 1.3.

3)The reverse-turn or beta-bend (see figure 1.6). A typical protein chain must change directions many times. This is particularly true for globular proteins. The structure that enables this is the reverse-turn or beta-bend, and again is made stable by a hydrogen bond, this time between peptide i and peptide i+3.

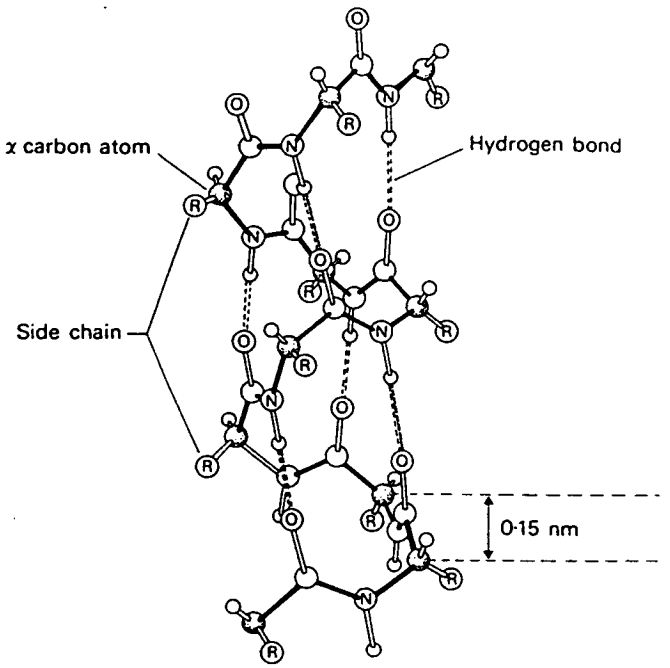


Figure 1.4

The alpha-helix

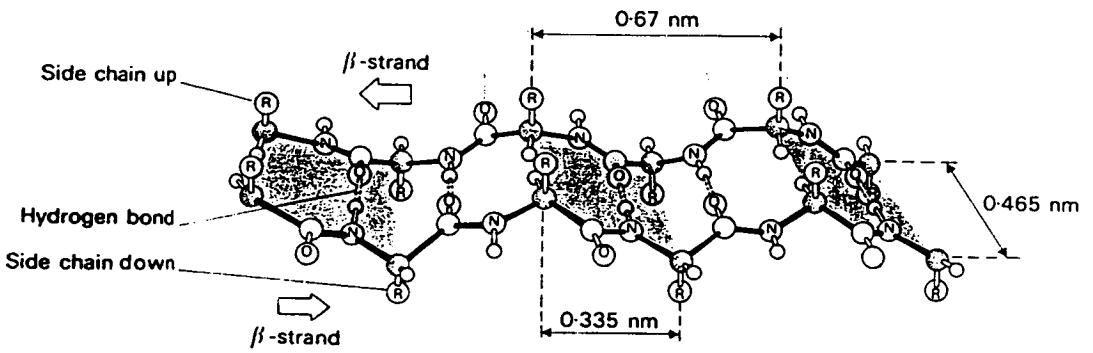


Figure 1.5

The beta-sheet

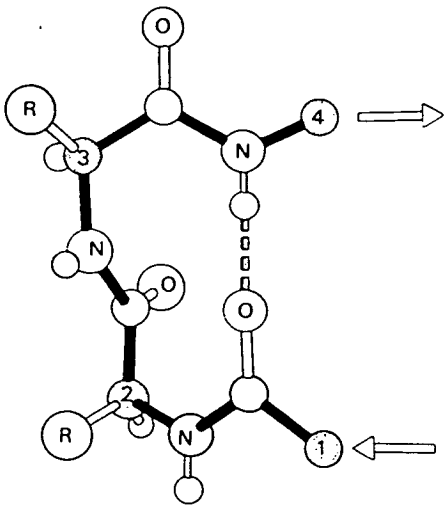


Figure 1.6

The reverse-turn

1.3 PROTEIN FOLDING

There are three non-covalent or van der Vaals forces driving the folding of the amino acid chain. They are:

1)An induced dipole-dipole interaction. This is a short range $1/r^6$ attractive interaction. The dipoles are induced on atoms by their neighbours due to the uneven charge distributions of atoms.

2)A very short range $1/r^{12}$ repulsive interaction that is a consequence of the Pauli Principle. This force takes steric hindrance into account.

3)A relatively long range electrostatic interaction that approximately follows the Coulomb law, ie 1/r. It is a result of the uneven charge distributions caused by covalent bonding. The hydrogen bond is a particularly strong example of this interaction. It occurs if a covalently bound hydrogen, which will have an excess of positive charge, has a contact partner with an excess of negative charge, due to its covalent bond. A highly attractive Coulomb potential results. Figure 1.1 shows side chains grouped according to their physical properties such as polarity.

Along with these, bending, stretching and torsional potentials will also influence the outcome of the final structure. Thermodynamically speaking the folding protein in its aqueous solution should evolve its structure until a minimum of the free energy is reached. It has been argued that this may in fact not be the case, and that the final folded structure is the result of some fast folding pathway. From a thermodynamic point of view this would imply that the protein is fixed in a structure corresponding to a minimum, but one of much higher free energy than the global one. It would, however, have to be sufficiently deep for the structure to be stable. Indeed, evidence is emerging that the free energy surface of a

protein has multiple minima (Noguti and Gō, 1989). Even if the actual minimum is not the global one, it may, however, be sufficiently close to it for thermodynamic arguments to be valid. Assuming this to be the case one can now begin to understand some of the structural features of proteins. Thermodynamically two quantities are competing: the binding energy[2] and the entropy. The free energy equation is:

$$F=E-TS, \tag{1.1}$$

where F is the free energy, E the energy, T the absolute temperature and S the entropy. As the equation shows there is a competition between energy and entropy, as decreases in energy are often accompanied by a decrease in entropy and increases in energy by an increase in entropy. Considering each residue individually, the arguments become relatively simple. Polar side chains can form hydrogen bonds with water so decreasing their binding energies. As these polar side chains in forming hydrogen bonds act simply like water molecules, there is no accompanying reordering of the water and consequent decrease in entropy. This results in an overall decrease in free energy and is therefore a favourable interaction. As a consequence polar residues are hydrophilic and are expected to be found on the outside surface of proteins. Hydrophilic residues have been shown to occur more often in turns than in other regions. As turns are often at the surface, this supports the reasoning given above. Non-polar side chains however cannot form hydrogen bonds with water molecules. The water molecules, unable to form hydrogen bonds, order themselves, resulting in an entropy decrease. Non-polar side chains are therefore hydrophobic, and if they can move to a non-aqueous environment in the interior of the protein, this unfavourable decrease in entropy can be avoided. But a residue in the interior of a protein will leave the carbonyl oxygen and amide hydrogen of its main chain unable to form

[2]As long as there is no appreciable change in volume, the binding energy is equal to the binding enthalpy and the Helmholtz free energy is equal to the Gibbs free energy.

hydrogen bonds with water molecules. This unfavourable situation can be remedied by the main chain atoms forming hydrogen bonds amongst themselves, resulting in characteristic secondary structure. Thus residues with non-polar side chain are expected to be found in the interiors of proteins and to be associated with secondary structure formation. Perhaps alpha-helix and beta-sheet can also be partly explained by the requirement of hydrophilic residues to lie on the protein surface. Given two hydrophilic regions on the protein's surface, and an intervening hydrophobic region whose length is longer than the direct distance between the two hydrophilic regions, then the most ordered way for the hydrophobic chain to structure itself so that the spatial distance it covers is shorter than its actual length would be for it to form a helical structure. A similar argument can be given for beta-sheet; it being the most extended structure. In terms of primary sequence it is possible to deduce the existence of some alpha-helices that occur on protein surfaces. Here hydrophobic residues can be found with intervals placing them all on the interior side of the alpha-helix. An analogous argument can be given for hydrophilic residues (see figure 1.7).

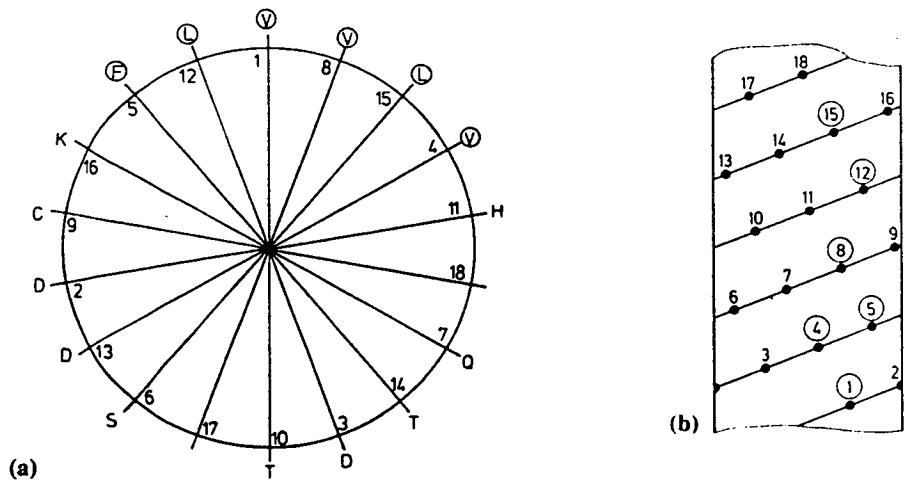


Figure 1.7

Helical wheel representation of C-terminal alpha-helix of adenylate kinase. (a) The wheel is the projection of all side chain positions along the helix axis onto a plane. Non-polar residues are encircled; note they all lie on one side of the helix. (b) A cylindrical plot. (After Schulz and Schirmer, 1978)

Similar arguments to those above pertaining to individual residues and secondary structure formation, can be applied to secondary structures themselves and their packing to form tertiary structure. For example, edges of beta-sheet must not end where the main chain polar atoms are unable to form hydrogen bonds. This means that alpha-helices cannot exist in the same layer as beta-sheet, and that beta-sheet must extend across the whole protein. Alpha-helices are found to be either anti-parallel or perpendicular. This tendency can perhaps be explained by the helix dipole which exists as a consequence of the aligned hydrogen bonds in alpha-helices. They are also expected to form layers as this is a natural way of packing rigid elongated bodies. This sort of reasoning has lead to an understanding of the topology of secondary structure arrangement in proteins (Richardson, 1981; Finkelstein and Ptitsyn, 1987).

The arguments above treating residues and secondary structures as if they were free to move to positions to reduce their own individual free energies cannot tell the whole story. Residues do not act independently but cooperatively, and their own individual free energy contributions may have to be sacrificed to reduce the free energy of the whole. Hydrophobic residues may therefore be found on the surface of a protein, and hydrophilic residues in the interior.

So far only hydrogen bonds have been mentioned but salt bridges may also make a contribution. More importantly, a covalent disulphide bond may occur between cysteine residues holding remote parts of the chain firmly together and having therefore a major influence on overall structure.

The results of this thermodynamic theorizing has lead to some understanding of the general properties of protein folding but the ultimate goal is to predict the exact tertiary structure of a protein from its amino acid sequence. To this end computer simulations of protein dynamics have been made using mainly molecular dynamics and the Monte Carlo method. Many of these experiments have tried to bring the computing time within reasonable bounds by making simplifying

assumptions. The starting configurations are often chosen randomly. The results of such computer experiments have often been disappointing, probably due to the aforementioned simplifications (Cohen and Kuntz, 1989). Prediction of secondary structure is a less ambitious task. It can not only be approached from a physics angle, but also a statistical one, as there are now plenty of examples of secondary structures in the structural data banks.

2 SECONDARY STRUCTURE PREDICTION METHODS

In the early 1960s' researchers began to analyse the small but growing structural data banks for amino acid preferences of secondary structures. Guzzo (Guzzo, 1965) showed from the analysis of myoglobin and α- and β- haemoglobin that certain amino acids were alpha-helix breakers. Others such as Davies (Davies, 1964), showed correlations between amino acid composition and secondary structure. These results naturally led others to attempt to predict secondary structure purely from sequence data.

Secondary structure prediction can be approached from two fundamentally different directions. The first approach is statistical and uses the information in the structural data banks. For example alpha-helix sequences in the data bank can be analysed for motifs and patterns that all alpha-helices are hoped to have. The other method is a physics based approach. This may involve statistical mechanics or consider the patterns of hydrophobicity and hydrophilicity that would satisfy the energy arguments given in the previous chapter.

2.1 PHYSICS BASED METHODS

2.1.1 The Method of Lewis *et al.*(Lewis *et al.*, 1970)

The method of Lewis *et al.* is based on the Zimm-Bragg model (Zimm and Bragg 1959). This statistical mechanical approach considers a homopolymer whose residues can be in one of two states: the alpha-helical state α or the random coil state c. The partition function is dependent on two parameters σ and s. The σ is introduced into the partition function to model cooperative behaviour. It weights the partition function against boundaries of alpha-helix and random coil. The s is a measure of the residue's preference for the state α to c. In the work of Lewis *et al.* the σ and s values were assigned to all the amino acids. These values can be determined from experiments on homopolymers, although in this work they were given values based upon educated guesswork. From the partition function, probabilities of specific residues being in the alpha-helix conformation can be calculated for a given copolymer. Lewis *et al.* found a correlation between regions that were calculated to have higher alpha-helix probability and these regions actually being alpha-helix. In fact, on the proteins they tested, they found they could predict 68% of residues to be in the correct state.

2.1.2 The Lim Method(Lim, 1974a,b)

This method is based on hydrophobic-hydrophilic interactions determining secondary structure and is somewhat analogous but more general than that of the helical wheel (Schiffer and Edmundson, 1961) (see figure 1.7). It originates from the premise that there is a concordance between short range and long range interactions (Ptitsyn and Finkelstein, 1970) - long range interactions being those

between remote parts of the chain. This means that potential alpha-helix sequence for example will form an alpha-helix, as the long range influence will support the short range influence. The long range interactions will then impose their own properties on the primary sequence of an alpha-helix that will be independent of those imposed by local sequence. The Lim method considers the resulting requirements of these non-local interactions on the primary sequence. It is in essence a general approach to the stabilisation of secondary structure through hydrophobic-hydrophilic interactions. The main principle is that very hydrophobic side chains must be at least partially buried in hydrophobic cores, while hydrophilic side chains must not penetrate into these cores. By using plastic models Lim determined the ordering of side chains in an alpha-helix such that the above requirement is satisfied when a remote leucine side chain is positioned to contact the alpha-helix. He found that hydrophobic residues should be positioned at (i,i+4) if they occur in pairs or (i,i+1,i+4) or (i,i+3,i+4) if they occur in triplets. By designating amino acids to groups of hydrophobicity and hydrophilicity and demanding that these pairs and triplets must overlap, then one has satisfied all the conditions for alpha-helix formation. Similar arguments were also made for beta-sheet formation and random coil. Lim claimed 80% and 85% success in predicting alpha-helix and beta-sheet respectively for the proteins he considered.

2.2 METHODS BASED ON THE STRUCTURAL DATA BANK

In order to predict secondary structure properly there must exist an objective way of relating secondary structure unambiguously to the atomic coordinates derived from X-ray diffraction. The secondary structures alpha-helix, beta-sheet and reverse-turn are well defined in terms of main chain hydrogen bonding. X-ray diffraction, however, cannot reveal hydrogen bonds, and so hydrogen bonds have to be determined from the atomic coordinates. In the past, the crystallographers themselves have made their own assignments of secondary structure based on

various different methods. A more objective approach was introduced by Kabsch and Sander (Kabsch and Sander, 1983a). They proposed an algorithm for the unambiguous assignment of secondary structure from atomic coordinates which is based on the hydrogen bonding definitions of secondary structure. The distance and angle between the main chain carbonyl oxygen of one residue and the amide hydrogen of another residue are calculated. If these two values fall within certain limits, then hydrogen bonding is taking place according to this scheme. The determination of these limits is based upon a plausible physical argument of hydrogen bonding, although in reality there is no clear cutoff. The Kabsch and Sander algorithm, for assessing the presence of alpha-helix for example, defines first a minimal alpha-helix. The minimal alpha-helix has four residues from i to i+3 with hydrogen bonds, determined as described above, between residues i-1 and i+3 and residues i and i+4. Nothing is required of the residues at positions i+1 and i+2. Longer alpha-helices are then defined in terms of overlapping minimal alpha-helices. This definition is sufficiently unrestrictive to allow for certain imperfections such as missing hydrogen bonds or slight bends in the helix. The presence of beta-sheet and reverse-turns is assessed along the same lines.

2.2.1 The Chou and Fasman Method

As this method predates the Kabsch and Sander algorithm, the fifteen proteins used in the original paper by Chou and Fasman (Chou and Fasman, 1974a,b) had the crystallographers' secondary structures assignments. From these fifteen proteins the propensities of all twenty amino acids to occur in alpha-helix, beta-sheet and reverse turn are calculated. This propensity is defined as the ratio of the amino acid's frequency in the structure under question to the average frequency of all amino acids in that structure. On the basis of these propensities, residues are assigned one of six properties. In the case of alpha-helix they were: strong alpha-helix former, alpha-helix former, weak alpha-helix former, alpha-helix

indifferent, alpha-helix breaker and strong alpha-helix breaker. Analogous assignments were made for beta-sheet. Initially all residues in a segment whose secondary structure is to be predicted are assigned one of these six properties. If certain rules are met concerning these properties for a cluster of four residues, then a site of alpha-helix nucleation is said to have been located. This is based on the Zimm-Bragg theory that alpha-helices form at nucleation sites which reside at alpha-helix centres and then propagate outwards until unfavourable residues terminate their progress. In analogy to this, the Chou and Fasman method also propagates a predicted nucleation site of an alpha-helix outwards in both directions until alpha-helix breakers are met. Again, certain rules have to be satisfied for the alpha-helix to be terminated. For a given segment alpha-helix propensities will be competing with those of beta-sheet. The one with the highest propensity value will be the one that is predicted. On the fifteen proteins the individual amino acid propensities were derived from, 86% of alpha-helix and 97% of beta-sheet was predicted correctly.

2.2.2 The GOR Method

The GOR method (Garnier *et al.*, 1978) is an information theory based approach. If P(X/s) is the probability that secondary structure X occurs given sequence s and P(X) is simply the probability that X occurs then the information that s carries on the occurrence of X is defined by:

$$I(X;s)=Log\frac{P(X/s)}{P(X)}. \qquad (2.1)$$

If the sequence of residues has no effect on the occurrence of X, then I=0; if the sequence favours X then I>0, and if it is not favourable to the occurrence of X

then I<0. The measure:

$$I(X:N;s)=I(X;s)-I(N;s), \tag{2.2}$$

where N represents the absence of X, quantifies the preference of sequence s to form structure X, rather than N. In the GOR method one considers the conformational state X_j of a residue at position j in a given sequence of residues from j-8 to j+8. The desired quantity to calculate then is:

$$I(X_j:N_j;R_{j-8}..R_j..R_{j+8}). \tag{2.3}$$

Knowledge of this quantity would give the preferred conformation X or N of residue R_j given the sequence from R_{j-8} to R_{j+8}. Due to the small size of the structural data bank only the information contributions from single residues in the positions j-8 to j+8 were taken in the original GOR paper to estimate (2.3). That is:

$$I(X_j:N_j;R_{j-8}..R_j..R_{j+8}) \approx \sum_{m=-8}^{m=8} I(X_j:N_j;R_{j+m}). \tag{2.4}$$

Values for the individual $I(X_j:N_j;R_{j+m})$'s were estimated from the structural data bank using twenty six proteins whose secondary structure assignments were in this case determined by looking at Ramachandran plots. The conformation with the highest information value estimated by equation (2.4) was chosen as the predicted conformation. They achieved a prediction success of 56% on three states: alpha-helix, beta-sheet and coil, where coil is everything that is not alpha-helix or beta-sheet. The GOR method was updated in 1987 by Gibrat *et al.* (Gibrat *et al.*, 1987) by using the whole existing data bank with the Kabsch and Sander secondary structure assignments. This improved the prediction success only slightly however to 57%. Further to this, Gibrat *et al.* used a better approximation

secondary structure assignments. This improved the prediction success only slightly however to 57%. Further to this, Gibrat *et al.* used a better approximation to (2.3) than (2.4) by taking pairs into account. The resulting expression is:

$$I(X_j{:}N_j;R_{j-8}..R_j..R_{j+8}) \approx I(X_j{:}N_j;R_j) + \sum_{m=-8}^{m=8} I(X_j{:}N_j;R_{j+m}/R_j), \qquad (2.5)$$

where the first term is the information carried by the occurrence of a residue at j on the conformation of the residue at j and the term $I(X_j{:}N_j;R_{j+m}/R_j)$ represents the information carried by the residue at j+m on the conformation of the residue at j, given the residue at j. This latter term can be estimated from the residue pair frequencies in the data bank. Despite the difficulty in estimating these values due to the small number of specific residue pairs in the data bank, this better approximation to (3) improved the prediction accuracy to 63% on the three states.

2.2.3 The Neural Network Approach

Since this work was started, a number of papers on neural network approaches to this problem have been published. All have used the same type of neural network: a feed forward network trained by back propagation (see section 3.2). The first of these approaches (Qian and Sejnowski, 1988) achieved 64% on the three states. The coding scheme used for the input was similar to that used here (see section 6.2) but the structure prediction was for the residue at the centre of the window which was 13 residues in length. This work has recently been extended by Kneller *et al.* (Kneller *et al.*, 1990) by adding an extra input node for each residue to code for its hydrophobic moment. This barely improved overall prediction success. On proteins of different structural class, however, they achieved 79% on all α proteins (those containing α-helices but no β-sheet), 70%

on all β proteins (those containing β-sheet but no α-helices) and 64% on α/β proteins (those containing both α-helices and β-sheet). The similar work by Holley and Karplus (Holley and Karplus, 1989) achieved 63% on the three states. They used the same coding scheme as Qian and Sejnowski but used a window 17 residues in length to predict the structure of the central residue. The approach by Bohr *et al.* (Bohr *et al.*, 1988) managed 73% on two states: alpha-helix and non-alpha-helix. Again, the same coding scheme was used but a window length 51 was taken to predict the central residue's structure. Finally McGregor *et al.* (McGregor *et al.*, 1989) have used a neural network to predict beta-turn residues with a success of 71%. Again the same coding scheme is used, but this time whole beta-turns without any flanking sequence are used for training and testing.

2.3 COMPARISON OF LIM, GOR AND CHOU AND FASMAN METHODS

The Chou and Fasman, Lim and GOR methods are the most frequently used secondary structure prediction methods and have been evaluated independently for comparison. The three methods were tested on sixty two proteins with Kabsch and Sander assignments (Kabsch and Sander, 1983b). The GOR and Lim methods both achieved 56% for the three states, whereas the Chou and Fasman method achieved only 50%.

2.4 MEASURES OF PREDICTION SUCCESS

There seems no established measure for evaluating secondary structure prediction despite the many different measures proposed. For comparison purposes one would ideally like to have one single number, but it seems one measure cannot convey enough information for proper evaluation. The percentages given above,

unless otherwise stated, are the percentages of correctly predicted residues. For a three state prediction this is given by $(f_h+f_b+f_c)\times 100/N$, where f_h, f_b and f_c are the number of correctly predicted alpha-helix, beta-sheet and coil residues respectively, and N is the total number of residues in any of the three states. This measure seems to be the most frequently used, but it is not without its pitfalls. For example, consider a protein that is 70% alpha-helix; a "prediction method" that predicts every residue to be alpha-helix will achieve 70% prediction success with this measure. To remedy this, some measures take the number of falsely predicted residues into account, as in the measure suggested by Ptitsyn and Finkelstein (Ptitsyn and Finkelstein, 1970). This measure does not seem to have established itself, however. In the recent neural network publications, both the overall percentage of correctly predicted residues and the correlation coefficient have been quoted. The correlation coefficient is defined as:

$$Q_h = \frac{w.x - y.z}{\sqrt{(x+y).(x+z).(w+y).(w+z)}} \qquad (2.6)$$

where in the case of alpha-helix for example:

w=number of correctly predicted alpha-helix,
x=number of correctly predicted non-alpha-helix,
y=number of alpha-helix falsely predicted,
z=number of non-alpha-helix falsely predicted.

This measure is equal to 1 for total correct prediction, 0 for random prediction and -1 for total false prediction. This measure, quoted along with the percentage of correctly predicted residues, helps to reveal whether the high percentage is due to an imbalance of the secondary structure assignments. In the extreme example above the correlation coefficient would be 0. However, even these two measures quoted above can hide vital information.

3 NEURAL NETWORKS

The specific functions of the human brain such as recognition and intelligence are ones that even the most powerful serial computer fails to equal. As the individual operations of neurons are many orders of magnitude slower than the modern integrated circuit, the remarkable power of the brain must be achieved by the network of neurons functioning in parallel. This suggests that tasks such as recognition may best be tackled by modelling the brain. Here modelling may mean simply simulating a network of neurons on a sequential computer or by building machines with processors that can operate in parallel, so making use of the natural parallelism of the model. It is hoped that these parallel machines will be able to one day match the performance of the brain. Most neural network models do not attempt to model the precise form and function of neurons connected in the complex architectures found in real brains, but abstract only the essential features, in the belief that it is these that lead to the brain's properties. All neural network models, then, have two basic components: nodes, and synapses that connect some or all the nodes together. Each node has at least two quantities associated with it: the total input at the node or its potential and the output from the node or its activation. A synapse has usually one quantity associated with it: its weight. The total input at a node is usually the sum of all the output messages impinging on that node. The output from a node is a non-linear function of the total input and the output message is the output modified (usually multiplied) by the weight of the synapse along which the message is being sent.

If information is coded as a pattern of the outputs, then a system such as that described above can be made to store this information. Normally computers store information at separate locations in their memories and interaction amongst the

items of data is impossible. In a neural network, however, each item of data has an influence on the values of the weights and it is through the weights that information interacts.

3.1 THE HOPFIELD MODEL(Hopfield, 1982)

In the Hopfield model each node i has an activation S_i whose value can be either +1 or -1. The network is usually totally connected; that is, every node is connected via a synapse to every other node. The total input at nodes i is then:

$$I_i=\sum_{j\neq i} W_{ji}S_j, \tag{3.1}$$

where W_{ji} is the weight of the synapse connecting node j to node i. The activation or output from the node is given by:

$$S_i=SIGN(I_i). \tag{3.2}$$

The model so defined is dynamic, as a change in the activation of one node can in turn affect the activations of other nodes and so on. An information state is the specification of every node's activation. If this state doesn't change with time then it is said to be a stable state or stored. The useful property of the Hopfield network is the retrieval of total information from partial information. Each stored state is at the minimum of an energy well in the state space. This energy well will have a basin of attraction such that any state that falls within this basin will evolve until the state at the minimum is reached. This state, which may be a corrupted version of the stored original, is "captured" or "recognised" by the state at the

minimum. This is the mechanism behind the retrieval of total information from partial information or corrupted information. Whether a state is stable or not will depend on the weights. So by choosing the correct weights one can store information. One of the simplest prescriptions for this is the so called Hebb learning rule (Hebb, 1949). If one wants to store M states $\underline{S}^1$.....$\underline{S}^M$, then the weights should be adjusted as follows:

$$\Delta W_{ij}=\sum_{m}^{M} S_i^m S_j^m. \tag{3.3}$$

A problem with this is that only 0.14×N random states can be stored, where N is the number of nodes. If the states are correlated, then applying the Hebb rule will generally not be sufficient to store the states at all. To remedy this, learning rules have been devised, not only to increase the storage capacity, but also to enable the storage of correlated patterns. These learning rules have to achieve a more subtle set of weight values and involve training. A network trained with the following algorithm will be able to store correlated patterns.

$$\epsilon_i^m=S_i^m\sum_{j\neq i} W_{ji}S_j^m.$$

If

$$\epsilon_i^m\geq 0$$

then

$$\Delta W_{ij}=0 \tag{3.4}$$

If

$$\epsilon_i^m<0$$

then

$$\Delta W_{ij}=S_i^m S_j^m.$$

In one cycle of training, all patterns are considered for weight adjustments according to (3.4). Many cycles of training may be required before all the patterns are stored, as the weight changes that favour one pattern may upset the stablisation of others. Although a network trained in this way is able to store correlated patterns and generally store a greater number of patterns than a network that simply has its weights determined by the Hebb rule, it has a poor recognition ability when many patterns are stored. In other words the stored patterns have small basins of attraction. In fact changing the activation of a single node of a stored pattern may send it tumbling into another stable state. A method is needed that widens the basins of attraction of the stored states. One method devised is training with noise (Gardner *et al.*, 1987). Here the basins of attraction are widened by using noisey versions of the patterns to be stored. In each cycle these noisey versions, $\underline{\Omega}^m$, of the original patterns, $\underline{S}^m$, are generated using random noise and the weights are adjusted such that these noisey versions should be recognised by their original versions in the dynamical phase of the Hopfield algorithm. The training cycle algorithm is similar to that of (3.4).

$$\epsilon_i^m=S_i^m\sum_{j\neq i} W_{ji}\Omega_j^m.$$

If

$$\epsilon_i^m\geq 0$$

then

$$\Delta W_{ij}=0 \qquad (3.5)$$

If

$$\epsilon_i^m < 0$$

then

$$\Delta W_{ij} = S_i^m \Omega_j^m .$$

In each training cycle a new noisey version of each of the original patterns is generated. To ensure that the stabilisation of the original patterns is not upset, this procedure may need to be supplemented by a phase of training with (3.4), i.e. training with zero noise. After training with (3.5), the stored patterns should have relatively wide basins of attraction in comparison with those trained with (3.4) alone. In other words, the network trained according to (3.5) should be much better at pattern recognition than a network trained according to (3.4).

With the Hebb learning rule, the weights are symmetric; that is $W_{ij} = W_{ji}$. With the training algorithms (3.4) and (3.5), the network is naturally asymmetric, i.e. $W_{ij} \neq W_{ji}$. The network can, however, be made symmetric by requiring that the weight change for a weight connecting node j to node i is equal to the weight change for the weight connecting node i to node j. This is achieved by adding these two weight changes, as given in (3.4) or (3.5), together. Although the symmetric and asymmetric networks both develop into stable terminal states, asymmetric networks are more mobile than symmetric ones; if a pattern is not well recognised the network can reach states at greater Hamming distances.

One of the basic features of the Hopfield network is the occurrence of so-called spurious states. These are stable states that arise unavoidably, and, as the name suggests, their relationship to the intentionally stored states is not totally clear, although Amit *et al.* (Amit *et al.*, 1985) have shown that in the limit of large networks they correspond to well defined mixtures of several stored patterns.

Their number increases with the number of patterns successfully stored and they are the main reason for the accompanying worsening of the network's recognition capability. Any pattern that lies outside the basins of attraction of any of the intentionally stored patterns will inevitably be captured by a spurious state.

3.2 LAYERED NETWORKS

Figure 3.1 shows a layered network. The synapses connect nodes between layers but not within a layer. The first layer is the input; the final layer, the output, and the intervening layers are the hidden layers. A network of this type is appropriate to the problem where, for example, one has two sets of data: set 1 and set 2, which are both subsets of their respective total data sets, and one would like the network to use this subset of information to classify correctly any data item from one of the two total sets. The data items appropriately coded are the input patterns at the input layer which pass forward through the network to the output layer. The output from the output layer is the indicator of the presence of a pattern belonging to set 1 or set 2 at the input. In this example, a single output node would be sufficient, with an output of -1 indicating a pattern belonging to set 1 at the input, and +1 a pattern belonging to set 2. Whether all the patterns output the correct output will depend on the weights. It is the object of the training phase to adjust the weights such that the correct outputs or target outputs are achieved. It is through the weights that all items of data interact and the essential difference between the two sets is determined. After training, any pattern that does not belong to the training subsets that outputs -1 is thus being predicted by the network to belong to set 1, and any pattern that outputs +1, set 2.

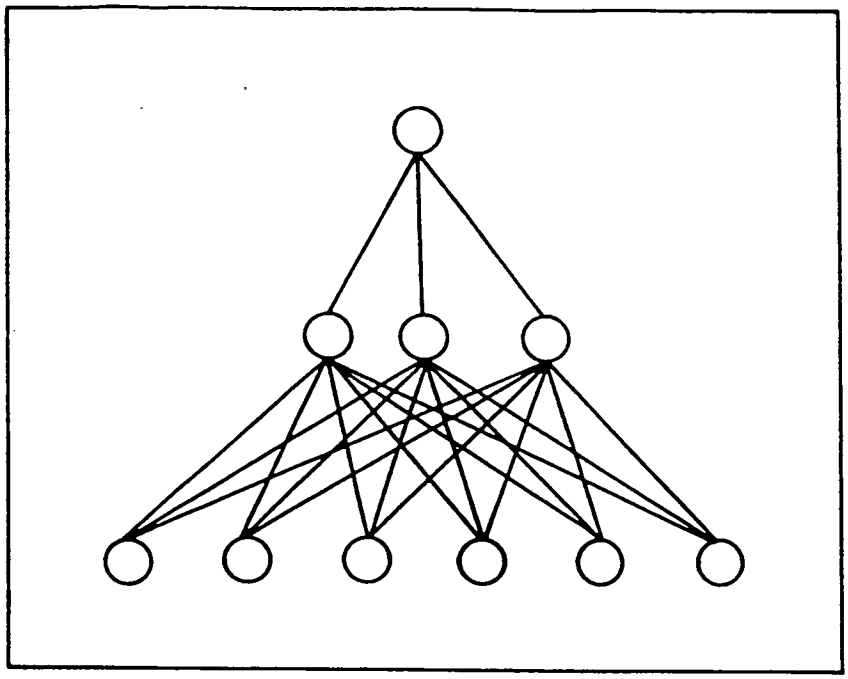


Figure 3.1

A layered network with one hidden layer

The single layer Perceptron is the simplest and will be considered first. In the two set example all the input nodes are connected directly to the single output node. The output O from the output node is given by a hard limiting function of the total input at the output:

$$O=F(\sum_{i=1}^{N} W_i I_i-\Theta)=\pm 1, \tag{3.4}$$

where I_i is the input value at node i for pattern $\underline{I}$ and W_i is the weight connecting the input node i to the output node. For a particular $\underline{I}$ the target output T may or may not have been achieved. If it is not achieved, the Perceptron learning

algorithm is applied (Minsky and Papert, 1969). With this the weights are adapted as follows:

$$\Delta W_i = \eta(T-O)I_i. \tag{3.5}$$

Note that no change in the weights occurs if the target output and the actual output are equal. A single layer network trained this way can only be used on problems that are linearly separable (Minsky and Papert, 1969). That is, the two sets must be able to be separated by a hyperplane in the input space (see figure 3.2). For problems whose sets are bounded by more complicated hypersurfaces, a network with hidden nodes must be used (see figure 3.3). The algorithm most commonly used for training a multi-layered network is called back propagation (Rumelhart *et al.*, 1986). For a network trained by back propagation the hard limiting function for the output above is replaced by a sigmoid function:

$$O_i = \frac{1}{1+e^{-\beta\phi_i}}, \tag{3.6}$$

where β is called the "temperature" and controls how step like the sigmoid is, and ϕ_i is given by:

$$\phi_i = \sum_j W_{ij}O_j + \Theta_i. \tag{3.7}$$

At the output layer the following measure of the error between the actual output

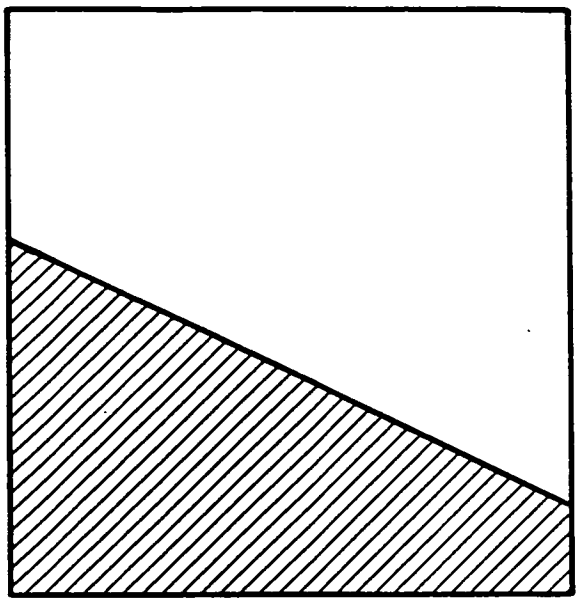


Figure 3.2

A linearly separable region solvable by a single layer network.

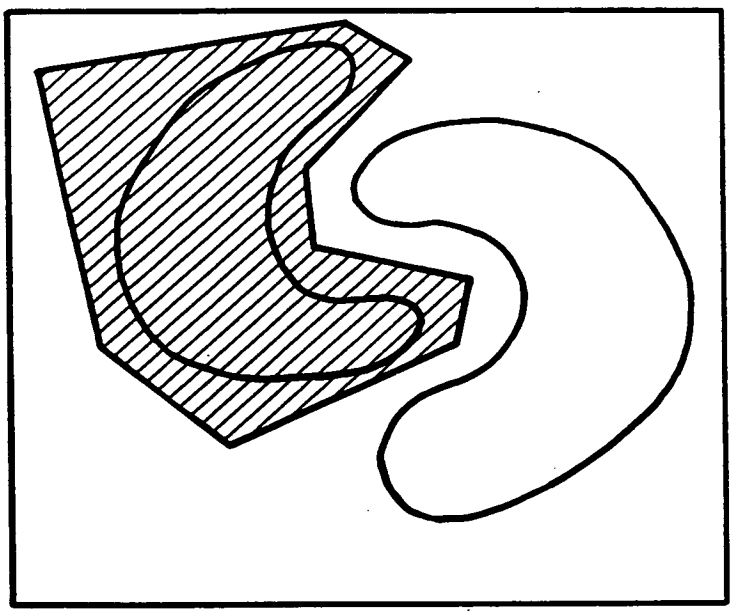


Figure 3.3

A more complicated division of the input space solvable only by a network with hidden nodes.

from pattern p and its target output is used:

$$E_p = \frac{1}{2}(T_p - O_p)^2. \tag{3.8}$$

The total error will be the sum over all the patterns. The object of the algorithm is to minimize the total error by gradient descent. If one imagines the error represented by a surface in weight space, then if one changes the weights according to the following rule:

$$\Delta W_{ij} \propto -\frac{\partial E_p}{\partial W_{ij}}, \tag{3.9}$$

where W_{ij} is the weight between nodes i and j, the error will reduce along its steepest path in weight space. The quantity then on the left of equation (3.9) is calculated and the weight adjustments made. For weights connecting to output nodes this is comparatively trivial, but for weights connecting to hidden nodes it is not immediately clear how this quantity is calculated. However, one can calculate equation (3.9) in terms of the weights connecting to hidden nodes by use of the chain rule. So, weights further back in the network can be calculated in terms of derivatives calculated for the layers above. Adjustments, then, are first made to the weights connecting the last hidden layer and the output layer, followed by weights adjustments in the layer directly beneath, and so on until the input layer is reached. In addition to weights each node has a threshold Θ_i associated with it and these thresholds are updated in much the same way as the weights. So one cycle of training involves a forward pass of one or all the patterns, the calculation of the error and the backward pass involving the weight and threshold adjustments that should reduce this error. This process is repeated until some tolerance criterion concerning the actual outputs and the target outputs is met. In general it will take many cycles of training before the error is reduced

to its global minimum if, indeed, it is able to find it. One of the major snags of this training algorithm is that the network is often likely to get trapped in minima other than the global one. Although equation (3.9) gives the general prescription for back propagation, it gives only the direction of the steepest path in weight space, but does not reveal how far down the slope one can step, i.e. the step size. This is the constant of proportionality in equation (3.9) and in practice finding the right step size is a matter of trial and error. Often in practice two other quantities are also added to equation (3.9) giving:

$$\Delta W_{ij}^{n} = -\eta_{ij} \frac{\partial E_p}{\partial W_{ij}} + \alpha \Delta W_{ij}^{n-1} - \theta W_{ij}. \qquad (3.10)$$

Here η_{ij} denotes the step size for the weight W_{ij}, α the "momentum", and θ the decay rate, and n the training cycle counter. The momentum term ($0 \leq \alpha < 1$) includes in the weight change, a part of the weight change of the previous cycle. In terms of weight space this tends to smooth out the trajectory taken along the error surface. The weight decay factor is sometimes used to keep the weight values from getting too large. Some workers have found that this can sometimes improve the generalising capabilities of the network. If the value of θ is too large it will have diastrous effects and again its optimal value can only be found by trial and error.

3.3 OTHER NETWORKS

The two networks described above are not appropriate for many problems to which neural networks can be applied. For example, one may not even know which sets the training patterns belong to. The network must now find out itself how to group the training patterns. This is a much harder problem for a network as it has much less information to work with. For such a problem, a competitive

learning network (Rumelhart and Zipser, 1985) or a Carpenter/Grossberg Classifier (Carpenter and Grossberg, 1986) may be used. Another type of layered network is the Boltzmann Machine (Hinton and Sejnowski, 1983). In architecture it is identical to the Back-Propagation network above, but is stochastic. During training the network learns to reproduce the probability distribution of its environment which "clamps" the input and output nodes. After training the unclamped network should be able not only to reproduce the environment's probability distribution but also to generalise beyond it and complete partially specified inputs and outputs. Unfortunately, this network is very slow to train.

RESOURCES

4.1 BROOKHAVEN DATA BANK(Bernsten *et al.*, 1977)

The Brookhaven data bank contains the structural information on proteins, whose structures have been determined by X-ray crystallography. The Brookhaven laboratory collects the atomic coordinates and other information from crystallographic studies and makes this data available in files of consistent format. Along with the atomic coordinates, other pertinent information is included in the files e.g. structure factors. At the present time there are 554 coordinate entries in the data bank, although many of these are not unique. The Brookhaven National Laboratory distributes this information to national centres around the world for national distribution. In Britain the SERC's Daresbury Laboratory is the national centre for the distribution of Brookhaven data.

4.2 KABSCH AND SANDER PROGRAM

In this work the Kabsch and Sander secondary structure assignments were determined by running the Kabsch and Sander program directly on Brookhaven files. This gave sequence with secondary structure assignments as output.

4.3 WISCONSIN PACKAGE(Devereux *et al.*, 1984)

This is a package of programs for protein or nucleic acid sequence analysis including secondary structure prediction.

4.4 BIPED

BIPED is a structural data base using the relational data base system, Oracle, and is derived from the Brookhaven data bank. Using the query language SQL it is possible to get answers to specific questions concerning structure, e.g. select all the ψ and ϕ values of all prolines in the fourth position in an alpha-helix.

4.5 FRODO(Jones, 1982)

Frodo is a macromolecular modelling program designed as a tool for the protein crystallographer. It was used simply to display the backbone structures of proteins of interest. The version of Frodo at our disposal was one that had been specifically designed for an Evans and Sutherland PS300 Graphics system with a VAX as a host. The photographs in this thesis are taken directly with an ordinary camera from the screen of the PS300.

4.6 COMPUTING FACILITIES

In the main only two computers were used: the Meiko Computing Surface and a NAS machine using the EMAS operating system. Occasionally a VAX was used to access the Wisconsin Package and the VAX at Daresbury for BIPED and the Brookhaven data bank. As already mentioned an Evans and Sutherland PS300 was used with Frodo. The EMAS system was used for manipulation of data for input to the network and processing its output data.

4.6.1 The Computing Surface

The Meiko Computing Surface is a super computer being developed in Edinburgh by the Meiko Company and the Edinburgh University Computing Surface (EUCS). The computing surface is a transputer based machine, and when it is completed should contain over 1000 transputers. Each transputer is by itself a fairly powerful processor. Its main feature, however, is that it can be connected to four other transputers. Along these connections commands and data can be sent in both directions. This feature allows parallel processing. Given that a task can be divided into parts that can be processed simultaneously, then these parts can be farmed out to different transputers. Each part may not however necessarily be totally independent of each other part, it may require information from another before it can proceed. This information can be sent along the connections. A group of transputers may be connected in different ways depending on the problem. The transputers that do the raw processing are normally called slaves. One transputer is needed to coordinate the slaves and is usually called the master. Further to this, a host transputer is needed to communicate with the user. The transputers in Computing Surface are divided

into domains, of which the largest currently has 132 transputers. Along with the hardware development, software has also been developed in the parallel processing language Occam for problems with inherent parallelism. Neural networks models are an obvious example where there is a large degree of natural parallelism. On implementation of neural network programs on parallel machines, how the parallelism can best be exploited will depend on the exact hardware of the computer. There is not usually a one to one correspondence between a node of a neural network and a simple processor in the computer. This direct mapping of node to processor would be a costly waste of resources if the individual processors were transputers.

4.7 MAIN PROGRAMS USED

Various programs were written to prepare data for the networks used, and to process the results. In the main this work was done on EMAS and the programs were written in fortran.

4.7.1 The Hopfield Program

A program based on the Hopfield model was written in fortran. Although fortran is not a natural language for transputers, it was intended to implement this on an array of transputers, using the software tools available to farm out parts of the program that could run in parallel. The program was run successfully on a single transputer but was abandoned before it was realised on an array (see section 5 for more details).

4.7.2 The Back Propagation Program(Richards and Tolleraere, 1990)

In both the forward and backward passes matrix-vector multiplications need to be performed. In the case of the forward pass, the inputs at the nodes in a layer are calculated by multiplying the matrix consisting of the weights connecting the nodes in the layer concerned with the nodes in the previous layer, by the vector of outputs from the nodes in the previous layer. In the implementation of the back propagation algorithm on a transputer array by Richards (Richards, 1990), the weight matrixes are divided amongst the slaves allowing each slave to do a part of the whole matrix-vector multiplication. Thus these matrix-vector multiplication are carried out in parallel, with each slave contributing its part to the final vector. The transputers are connected in the torus configuration. In this work the program was run on a 17 transputer array; that is 16 slaves and one master. The program is a general purpose and command driven, allowing one to enter commands from the keyboard. One can choose the number of layers, nodes in each layer, connectivity of the nodes, set the weights to either constant or random values as well as setting various other parameters. The training and testing sets can be read from files and during training, all the necessary information to continue training can be filed or read from files. In this program all the training patterns are loaded for the forward pass. The total error is calculated and the weights updated according to the back propagation scheme. This process forms one training cycle.

Even with this considerable computing power a single run such as that depicted in figure 7.1 (see section 7.1) requires approximately 40 minutes to complete. During the course of this work both the back propagation program and the Meiko Computing Surface were under development. The large number of bugs in both the software and hardware resulted in a considerable hinderance to the smooth progress of this work, although in the latter stages both systems stabilised.

EXPERIMENTAL

The object of this work was to predict protein secondary structure. It was decided to concentrate, in the main, on the alpha-helix structure, as this structure is more suited to the neural network approach than the reverse-turn, which involves only four residues, and more modular and abundant than beta-sheet. From here onwards alpha-helix will be simply referred to as helix. Unless otherwise stated a sliding window of length 10 residues has been used.

5 THE HOPFIELD APPROACH

5.1 IDEA BEHIND APPROACH

Due to historical reasons the Hopfield model was the first neural network model to be used in this work. In section 3.1 it was stated that this network could be used to store patterns of nodal activation, where the activation of a single node can be 1 or -1. It was our intention to store sequences of helix and non-helix as patterns of nodal activation. How could a network of stored helix and non-helix sequences be used to predict the structure of unstored or "unknown" sequences? There are two mechanisms by which one can imagine this could happen. The most obvious mechanism is that by which unknown helix sequences would be captured by the stored helix sequences and unknown non-helix sequences by the stored non-helix sequences. Thus for any sequence captured by any one of the stored helix sequences a prediction for helix is being made, and similarly for non-helix. It is quite unlikely, however, that the unknown sequences will all be captured by the stored states. As already mentioned in section 3.1 spurious states abound in this model and it is likely therefore that some sequences will be

captured by spurious states. This leads us to the second possible mechanism by which one could imagine how the structure of unknown sequences could be predicted. The spurious states may not be a problem in this interpretation, but an essential feature. As already mentioned, the actual relation of the spurious states to the stored states is not totally understood, but they do correspond to mixtures of stored patterns. In other words spurious states bear some relation to the states stored, and it is possible that they correspond to generalised features of the stored patterns. Thus any sequence captured by a spurious state may not be lost for prediction. It is feasible that if patterns of activation for helix sequences are in separate regions of the configuration space from patterns of activation for non-helix sequences, then the spurious states associated with the helix sequences may also be separated in the configuration space from the spurious states associated with the non-helix sequences. If one can find then, an association between the spurious states and the stored states, spurious states may be integral for prediction.

5.2 THE CODE

Although Hopfield network is of infinite dimensionality, as every node is connected to every other node, it is convenient to think of the nodes arranged in 2 dimensions located at the lattice points of a simple square lattice. For the purpose of storing sequences a method is needed to convert sequence into a set of activations. That is, there must be a one-to-one correspondence between a sequence and a pattern of activation. It is clear that the coding scheme used for converting sequence into a pattern of activation is of paramount importance for the success of the network. Here a distributed representation (Hinton *et al.*, 1986) was chosen that reflected the physical properties of each of the twenty amino acids. These physical properties were hydrophobicity, polarity, size, and charge. Their values for each of the twenty amino acids are shown in Table 5.1.

These values have to be converted into a series of 1's and -1's if the Hopfield network is to store them.

	Amphiphilicity			Charge	
Residue	Hydrophobic	Polar	Size	Positive	Negative
Ala	0·0	−0·3	−0·3	0·0	0·0
Arg	−0·4	0·3	0·3	0·5	0·0
Asn	−0·5	0·6	0·0	0·0	0·0
Asp	−0·7	0·2	0·0	0·0	0·5
Cys	0·6	0·1	−0·2	0·0	0·1
Gln	−0·4	0·6	0·1	0·0	0·0
Glu	−0·6	0·2	0·1	0·0	0·5
Gly	−0·1	−0·3	−0·4	0·0	0·0
His	−0·2	0·2	0·2	0·3	0·0
Ile	0·9	−0·3	0·0	0·0	0·0
Leu	0·7	−0·3	0·0	0·0	0·0
Lys	−0·5	0·2	0·1	0·5	0·0
Met	0·5	0·1	0·0	0·0	0·0
Phe	0·6	−0·2	0·3	0·0	0·0
Pro	−0·5	−0·3	−0·1	0·0	0·0
Ser	−0·5	0·2	−0·2	0·0	0·0
Thr	−0·3	0·2	−0·1	0·0	0·0
Trp	0·4	0·1	0·6	0·0	0·0
Tyr	0·2	0·1	0·4	0·0	0·0
Val	0·9	−0·3	−0·1	0·0	0·0

TABLE 5.1 (After Bacon and Anderson, 1986)

For each of the physical quantities, a positive constant equal in magnitude to the most negative value of that quantity was added to all values. In this way all negative values were eliminated. In the case of hydrophobicity, for example, the most negative value is -0.7 for aspartic acid. Adding 0.7 to all the hydrophobicity values gives a maximum value of 1.6, for both isoleucine and valine. Hydrophobicity was therefore assigned 16 activation values and the hydrophobic part of the code for isoleucine and valine has 16 -1's. In the case of tyrosine, for example, the value of 0.9, gives 9 -1's, the remaining 7 activation values being 1. In the extreme case of aspartic acid, all 16 nodes have activations of 1. The same procedure was used for the other properties. The resulting code for each amino acid is a row of activations, the first 16 coding for hydrophobicity, the next 9 coding for polarity, the next 10 coding for size, and the final 10 coding for charge. In addition to these four properties a further code of 7 activation values was

acids except for proline, which has 7 -1's. This part of the code is meant to code for "prolineness" ,i.e. it is included to represent proline's special properties in being an imino acid. Figure 5.1 shows the resulting codes for all twenty amino acids. A sequence of ten amino acids, for example, arranged in a column, with their codes, each a row of activation values alongside, gives the 2 dimensional pattern of activation. This pattern of activation reflects the physical properties of the sequence, and, as each amino acid has a unique code of 1's and -1's, every unique sequence will have a unique pattern of activation associated with it.

5.3 DETAILS OF PROGRAM

The first choice one has to make in using the Hopfield program is whether the network is to be symmetrical or asymmetrical. To specify the structure of the network completely, the size must also be specified. This is dependent on two parameters: the length of the code, and the length of the sequence. For input, the program simply requires the sequences for storage and the test sequences for prediction. The structures of the test sequences are known by the user for the purposes of evaluation. In addition, the program can either use the simple Hebb learning rule or learning with noise. For learning with noise it requires the user to specify the percentage of random noise, that is the percentage of activation values that are to be changed form 1 to -1 or vice-versa for each storage pattern, as well as the number of training cycles or weight adjustments for that percentage of noise. In addition the weight values from a trained network can be put into a file making that particular "brain" available for later use. After training, each pattern, irrespective of whether it is a storage pattern or a test pattern, is allowed to evolve until it is captured by a stable state, which could either be a stored pattern or spurious state. Of course, if the network is successful, then all storage patterns will remain unchanged.

	"PROLINENESS"	HYDROPHOBICITY	POLARITY	SIZE	CHARGE
A	1 1 1 1 1 1 1	-1-1-1-1-1-1-1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1	-1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
C	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1-1 1 1 1	-1-1-1-1 1 1 1 1 1	-1-1 1 1 1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1
D	1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
E	1 1 1 1 1 1 1	-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
F	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1-1 1 1 1	-1 1 1 1 1 1 1 1 1	-1-1-1-1-1-1-1 1 1 1	-1-1-1-1-1 1 1 1 1 1
G	1 1 1 1 1 1 1	-1-1-1-1-1-1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
H	1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1-1-1-1-1 1 1 1 1	-1-1-1-1-1-1-1-1 1 1
I	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	1 1 1 1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
K	1 1 1 1 1 1 1	-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1
L	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1-1-1 1 1	1 1 1 1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
M	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1 1 1 1 1	-1-1-1-1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
N	1 1 1 1 1 1 1	-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1	-1-1-1-1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
P	-1-1-1-1-1-1-1	-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1	-1-1-1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
Q	1 1 1 1 1 1 1	-1-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1	-1-1-1-1-1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1
R	1 1 1 1 1 1 1	-1-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1-1 1 1 1	-1-1-1-1-1-1-1 1 1 1	-1-1-1-1-1-1-1-1-1-1
S	1 1 1 1 1 1 1	-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
T	1 1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1	-1-1-1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
V	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	1 1 1 1 1 1 1 1 1	-1-1-1 1 1 1 1 1 1 1	-1-1-1-1-1 1 1 1 1 1
W	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1-1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1-1	-1-1-1-1-1 1 1 1 1 1
Y	1 1 1 1 1 1 1	-1-1-1-1-1-1-1-1-1 1 1 1 1 1 1 1	-1-1-1-1 1 1 1 1 1	-1-1-1-1-1-1-1-1 1 1	-1-1-1-1-1 1 1 1 1 1

Figure 5.1

The program's output consists of four separate parts. The first part gives the Hamming distances between all the storage patterns; that is of the number of activation changes needed to convert one pattern into another. It also gives the Hamming distances between the test patterns and the storage patterns. The second part of the output gives the number of activation changes that occur after training is complete and each pattern is allowed to evolve to a stable state. For storage patterns that have been successfully stored, no activation changes will occur. For test patterns, their activations change as they settle into, or are captured by, stable states. If the storage patterns are not successfully stored, then they will also evolve until a stable state is reached, which could correspond either to another storage pattern, which has been successfully stored, or a spurious state. The third part of the output "translates" the patterns of activations of all patterns, after they have evolved into stable states, back into sequence by calculating the Hamming distances of each row of activation to each of the amino acid codes, and choosing those amino acids whose codes have the minimum distance. Of course, for sequences that have been successfully stored, the patterns of activation will translate directly back to their sequence. An error value, in terms of the Hamming distance between the pattern of activation and the nearest sequence is also given. The fourth and final part of the output gives the Hamming distances between the patterns after they have been allowed to evolve into stable states. If a test pattern is captured by a successfully stored storage pattern, the Hamming distance between the two patterns will be 0.

5.4 RESULTS

This program was implemented on a single transputer using sequences 10 amino acids in length. A training set was constructed of 50 helix sequences together with 50 non-helix sequences from the Kabsch and Sanders output file. Similarly a test set was also constructed from 50 helix sequences and 50 non-helix sequences.

These test sequences were chosen by having little similarity with the training set sequences. Initially the simple Hebb learning scheme was tried. As mentioned in section 3.1 the Hebb learning rule can only store $0.14 \times N$ random patterns, where N is the number of nodes, and it is not surprising that for even a small number of patterns the Hebb learning rule fails miserably to store patterns with such a large overlap. In fact, with the Hebb learning rule, a single spurious state formed which captured every single pattern. Consequently training with random noise was tried. However, the training phase on a single transputer required more time than was reasonable. To remedy this the code was scaled down by half (see figure 5.2). With this code each cycle requires 1.5 seconds per pattern on a single transputer. Training comprised of 10 cycles training without noise, followed by 20 cycles with 10% noise, followed by a 20 cycles with 20% noise and finally a further 10 cycles without noise. Virtually all the training patterns on both the symmetrical and asymmetrical networks were successfully stored. For the asymmetrical network not one of the test sequences was captured by a stored sequence, and for the symmetrical network only one was captured by a stored sequence. This means some other measure of prediction is needed. The most obvious method is to choose the structure of the stored state with the minimum Hamming distance to the spurious state that has captured the test sequence. Doing this for the symmetrical network 36 of the 50 helix sequences were correctly predicted and 22 of the 50 non-helix sequences were correctly predicted. If one assumes that the probability of a helix sequence being correctly predicted is 0.5, the probability that 36 or more are correctly predicted is about 1 in a 1000. Overall, however, 58 out of the 100 test sequences were correctly predicted, which is just on the border of the 5% significance level. For the asymmetrical network 31 out of the 50 helix sequences were correctly predicted and 27 of the non-helix sequences. Again that gives an overall prediction success of 58%. It does not appear, therefore, that there is any significant difference in performance of the two networks. There are many other questions one can ask regarding the spurious states that have captured the test sequences. One of the reassuring features of many of these spurious

```
A   1 1 1-1-1-1-1 1 1 1 1 1 1 1 1 1-1 1 1 1 1-1-1-1 1 1
C   1 1 1-1-1-1-1-1-1-1 1-1-1 1 1 1-1 1 1 1 1-1-1 1 1 1
D   1 1 1 1 1 1 1 1 1 1 1-1-1-1 1 1-1-1 1 1 1 1 1 1 1 1
E   1 1 1-1 1 1 1 1 1 1 1-1-1-1 1 1-1-1-1 1 1 1 1 1 1 1
F   1 1 1-1-1-1-1-1-1-1 1-1 1 1 1 1-1-1-1-1 1-1-1-1 1 1
G   1 1 1-1-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1-1-1-1 1 1
H   1 1 1-1-1-1 1 1 1 1 1-1-1-1 1 1-1-1-1 1 1-1-1-1-1 1
I   1 1 1-1-1-1-1-1-1-1-1 1 1 1 1 1-1-1 1 1 1-1-1-1 1 1
K   1 1 1-1 1 1 1 1 1 1 1-1-1-1 1 1-1-1-1 1 1-1-1-1-1-1
L   1 1 1-1-1-1-1-1-1-1 1 1 1 1 1 1-1-1 1 1 1-1-1-1 1 1
M   1 1 1-1-1-1-1-1-1 1 1-1-1 1 1 1-1-1 1 1 1-1-1-1 1 1
N   1 1 1-1 1 1 1 1 1 1 1-1-1-1-1-1-1-1 1 1 1-1-1-1 1 1
P  -1-1-1-1 1 1 1 1 1 1 1 1 1 1 1 1 1-1-1 1 1 1-1-1-1 1 1
Q   1 1 1-1-1 1 1 1 1 1 1-1-1-1-1-1-1-1-1 1 1 1 1 1 1 1
R   1 1 1-1-1 1 1 1 1 1 1-1-1-1 1 1-1-1-1-1 1-1-1-1-1-1
S   1 1 1-1 1 1 1 1 1 1 1-1-1-1 1 1-1 1 1 1 1-1-1-1 1 1
T   1 1 1-1-1 1 1 1 1 1 1-1-1-1 1 1-1-1 1 1 1-1-1-1 1 1
V   1 1 1-1-1-1-1-1-1-1-1 1 1 1 1 1-1-1 1 1 1-1-1-1 1 1
W   1 1 1-1-1-1-1-1-1 1 1-1-1 1 1 1-1-1-1-1-1-1-1-1 1 1
Y   1 1 1-1-1-1-1-1 1 1 1-1-1 1 1 1-1-1-1-1 1-1-1-1 1 1
```

Figure 5.2

states, is that when patterns are translated back into sequence, the resulting nearest-distance codes are for amino acids that are often substituted conservatively in sequence aligning. For example, alanine and glycine, or serine and threonine.

At this point a decision had to be made as to whether to continue with the Hopfield model or to try the back propagation network that became available. It was obvious that to improve upon the disappointing prediction results above, a much larger training set would have been required. However, in order to bring the running time to within reasonable bounds the Hopfield program would have to have been parallelized, either by rewriting it in Occam or by using one of the Fortran harnesses that at that time were still being developed. As in many ways a layered network, a classification network, seemed a much more natural choice for secondary structure prediction, it was thought to be wise to abandon the Hopfield approach temporarily if not permanently.

6 INITIAL EXPERIMENTS WITH LAYERED NETWORKS

The input patterns for the layered network were the sequences 10 residues in length from the sliding window. The output nodes were used to indicate which structural class the sequence belonged. Before training with the back propagation algorithm begins, the weight values have to be set. Normally the initial weight values were set randomly with values between -0.6 and +0.6. During the initial phase of training the momentum value was always set to either 0 or 0.3, and later, usually after 10 cycles of training, to 0.9. This gave the network a chance to find the right path for gradient descent.

6.1 EARLY RESULTS WITH PAIR CODING

6.1.1 Prediction with Segments of Specific Secondary Structure

For prediction with segments of specific secondary structure, the sequences in the Kabsch and Sander output files were separated into helix, beta-sheet and coil sequences by a sliding window 10 residues in length. No boundaries between helix, beta-sheet and coil regions were included. The first network tried had 66 input nodes, each corresponding to a pair that have been found to either favour or disfavour the helix structure (Gibrat *et al.*, 1987), (see table 6.1). Along with these pairs 5 nodes were assigned to 7 individual residues that were known to particularly favour or disfavour helix. These were alanine, glutamic acid, leucine, proline, glycine, aspartic acid and serine. These 5 nodes simply coded for the

PRO-HELIX PAIRS	ANTI-HELIX PAIRS
P-------K	**V**-------S
G-----A	**V**-------Q
A----H	**V**------S
L----**K**	V------**Q**
A----L	I-----C
Q----K	E----S
S----H	G---**N**
L---**I**	**V**---S
L---**F**	L---E
A---A	**V**--G
S---A	L--P
G---L	G-**A**
V---L	G-L
V---K	S-S
A---K	D-S
D---K	**V**-P
E---K	T-P
A--**K**	K-D
G--A	GA
V--A	GL
D--A	GD
L--L	GE
F--L	PL
A--K	AG
L--F	KG
L-**P**	PG
I-**N**	LP
E-L	TP
S-L	DP
D-L	NP
K-L	
S-I	
D-H	
D-F	
LD	

TABLE 6.1

Bold residues indicate the residue favoured to be in the helix conformation for the first column and in the non-helix conformation in the second.

number of occurrences of the particular residue in the sequence. For example if there were 3 alanines, the first 3 nodes assigned to alanine would output 1, the remaining 2,0. The motivation behind this coding scheme was that the translation of the window would leave the activations of the input nodes unchanged so long as the pair remains within the window. In this way translation of the window would be reflected by proportionate change in the output from the input layer. The target output was 1 0 0 for a helix sequence, 0 1 0 for a beta-sheet sequence and 0 0 1 for a coil sequence. The percentage of correctly predicted helix sequences as a function of training set size was investigated as was the influence of having a similar sequence in the training set. With a training set that comprised of 150 helix sequences and 150 non-helix sequences the network correctly predicted 62 out of 100 helix sequences in the test set, where a prediction for helix was one for which the first output node had the largest output value, a prediction for beta-sheet, where the central node had the largest output value, and a prediction for coil, where the final node had the largest output value. With a training set comprised of 750 patterns prediction rose to 75%. This preliminary result showed that the layered network was already achieving far better results than the Hopfield network.

6.1.2 Prediction Method for Three Structures

For the time being the approach taken above of predicting the structures of segments that did not span boundaries between different secondary structures was abandoned and an attempt was made to predict all the three structures, helix, beta-sheet and coil, using a method that required a network with 30 output nodes. Each of the three structures was assigned 10 output nodes corresponding to the ten possible positions for residues in the input window. In the case of helix, for example, only those of the 10 helix output nodes corresponding to the position of any residue in a helix will output 1 (see figure 6.1). The network had the same

66 input nodes corresponding to pairs in table 6.1, 40 hidden nodes, and the 30 output nodes. An attempt was made to train network to learn 1000 patterns. No amount of variation of the learning parameters could change the disastrous inability of this network to learn. This attempt was in hindsight misguided, as one was presenting very little information at the input, but was demanding a large amount of information at the output. This can only bear fruit if one has a large number of training patterns. This failed attempt does illustrate the following intuitive principle. The higher the ratio of the amount of information one presents for input to the network, to informational content one demands at the output, the better the chances are of good results. For a given number of patterns this ratio may be reflected in the ratio of the number of input nodes to output nodes. Here, for example, we have 66 input nodes and 30 output nodes, making this ratio obviously too small.

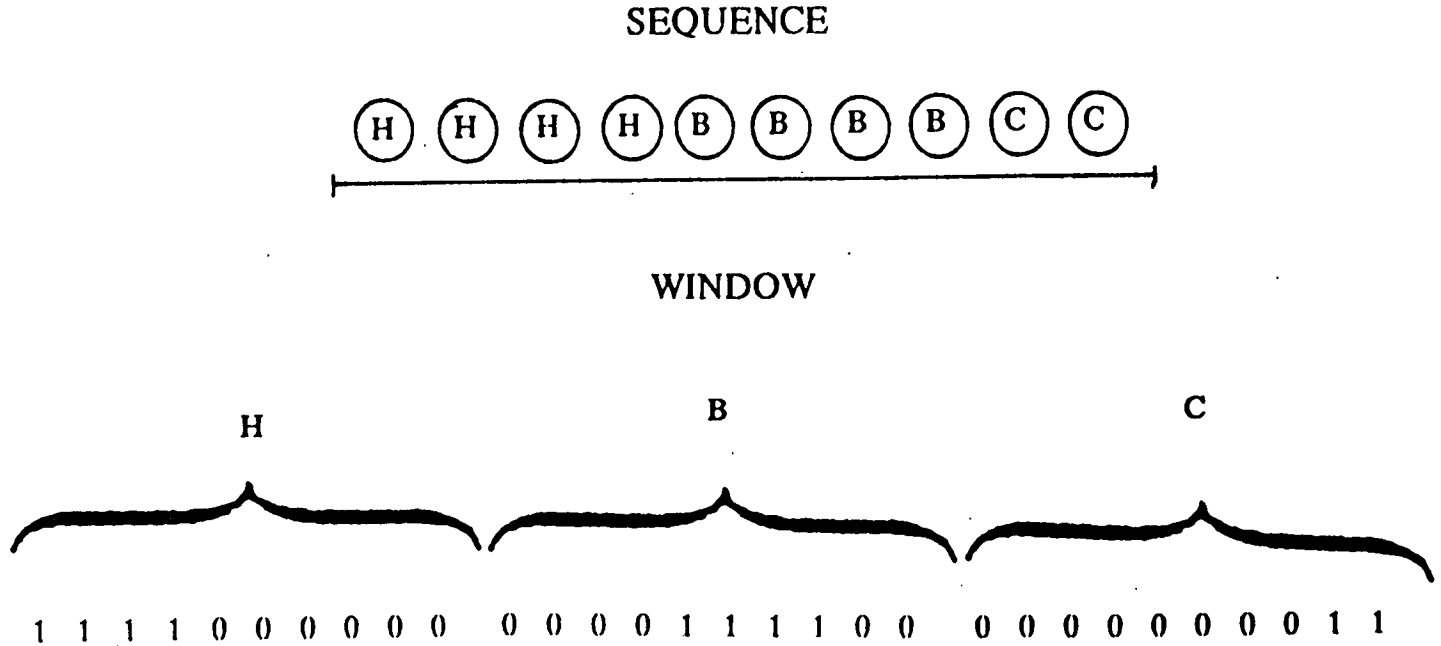

SEQUENCE
H H H H B B B B C C
WINDOW
H
B
C
1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1
OUTPUT CODE

Figure 6.1

6.2 PREDICTION OF HELIX BOUNDARIES USING POSITIONAL CODING SCHEME

At this point the paper by Qian and Sejnowski was published. In this work they selected 106 proteins from the Brookhaven Data Bank. Of these 15 were selected for testing by having little homology with the remaining 91 that were used for training (see Appendix A). For the purposes of comparison the same sets of proteins were used for training and testing. From here onwards all training and test sets were derived from them. It was decided at this point to attempt to predict structural boundaries, rather than the structures of the individual residues themselves. The problem was restricted to predicting only helix to non-helix, and non-helix to helix boundaries; in other words the beginnings and ends of helices. A different coding scheme to the pair coding scheme was used. Each of the twenty amino acids was assigned 10 nodes corresponding to the 10 possible residue positions in the window. If an amino acid x were present at position i, then x's i-th nodes would output 1, 0 otherwise. From here onwards this coding scheme will be referred to as the positional coding scheme. With this coding scheme a network requires 200 input nodes for a window 10 residues in length. Only those sequences for which a helix to non-helix or a non-helix to helix boundary was at the central position were selected for training, i.e. each sequence had 5 residues in both structures. A non-helix to helix boundary was given a target output of 1 0 and a helix to non-helix boundary 0 1 for a network with two output nodes. A network with 100 hidden nodes was trained on 383 patterns and tested on 398 patterns from two test set proteins 2ACT and 2ALP. The 383 training set patterns were derived from roughly half of the whole set of training proteins, the number being restricted by the length of time it took to train such a large network. The test patterns, in contrast to the training patterns, were not restricted to just boundary regions, but contained all sequences, length 10, from the sliding window. The network was trained virtually to 100% with to the low

error value of 2.05. This method was again unsuccessful, as all sequences, irrespective of whether they spanned boundaries, outputted either a 1 0 or 0 1, so making the determination of an actual boundary position impossible. This result illustrates an important point. One should not usually exclude features from training that are to be included in testing. In this case, this would mean that the sequences that do not contain helix boundaries should also be included for training. This presents us with an important problem. If one feature is very much more abundant than the other, such that the number of training examples of one outweighs the number of training examples of the other, then the network will be very difficult or impossible to train, as the gradient descent algorithm will often find the local minimum that corresponds to all patterns having outputs that are the target output of the most abundant feature. In the case of predicting boundaries this problem is particularly acute as the number of sequences length 10 that do not contain a helix boundary far outweighs those that do. To overcome this a network was trained to output a special sequence as the window slid across a boundary of a helix. Three output nodes were selected for this code. This method was tried first on non-helix to helix boundaries. Figure 6.2 shows the target codes for 5 positions of the window as it slides across the non-helix to helix boundary. It was hoped that only real boundaries from the test set would output the sequence 1 1 0, 1 0 0, 0 1 0, 0 0 1, 0 1 1, as the window slides across them, and regions not containing boundaries would not output such a sequence. The training set contained 304 patterns and could be trained to an error of 4.006; indicating that most of the patterns achieved their target outputs. The results from this on the same test set of 398 patterns were again disappointing. Sometimes the actual target sequence occurred where no boundary existed, but on the whole, the sequence did not occur at all, and the results were difficult to interpret. Although these results did not fulfil expectations, some boundaries were correctly recognised and so it was decided not to abandon this approach at this stage. Changing the code sequence for the output to 1 0 0 0 0, 0 1 0 0 0, 0 0 1 0 0, 0 0 0 1 0, 0 0 0 0 1, and increasing the size of the training set did not improve the situation, however. A final attempt was made to improve upon these results

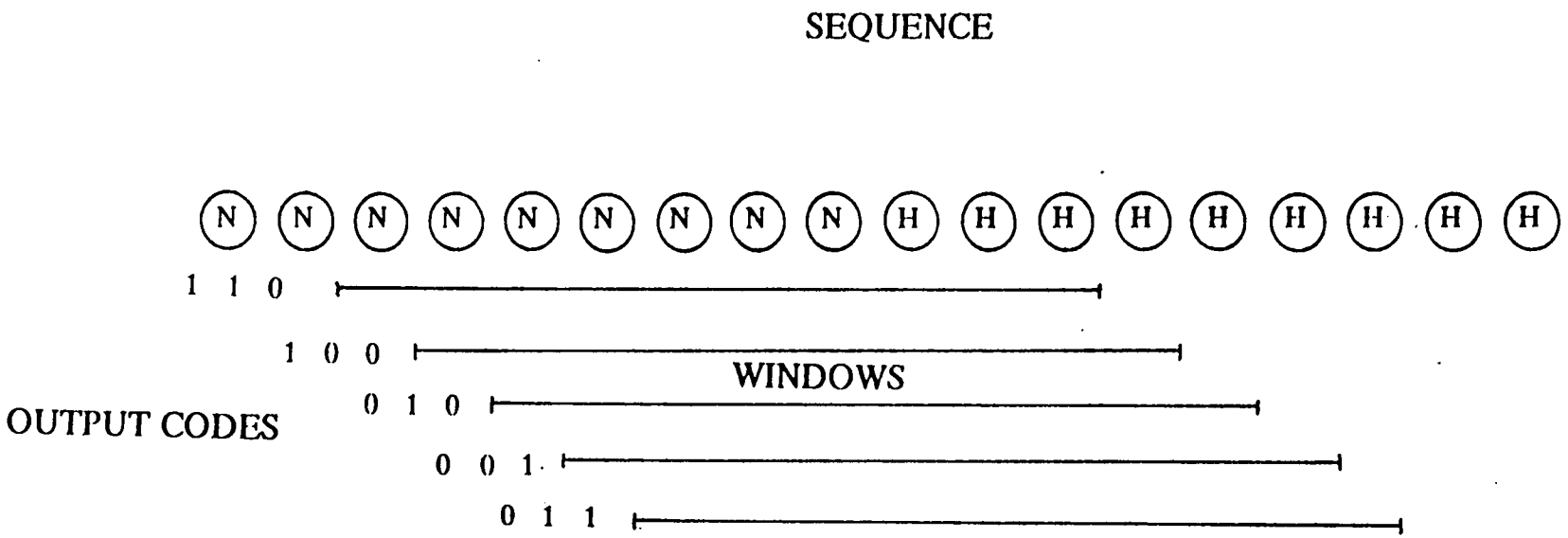

SEQUENCE
N N N N N N N N N H H H H H H H H H
1 1 0
1 0 0
WINDOWS
OUTPUT CODES
0 1 0
0 0 1
0 1 1

Figure 6.2

by considering a longer sequence of codes at the output. Given that a network will output one of the 5 possible codes only, it is possible that the sequence of 5 could occur by chance so indicating a boundary where there was none. Although the probability of this exact sequence occurring by chance is only 10% for 300 sequences, if, for the purposes of prediction, we are to allow a certain error of 3 correct out of the 5, for example, then this will occur 15 times by chance for 300 sequences. The use, therefore, of a longer sequence was investigated by calculating the probability of a sequence with a given number of errors occurring by chance and comparing this to the probability that the sequence with these errors would occur at the boundary, given that the network was trained to output the sequence at the boundary. These calculations did show that the use of a longer sequence than 5 would improve the situation, but the real results from trained networks suggested that this approach was not going to bear fruit.

6.3 PREDICTION WITH SEGMENTS OF SPECIFIC SECONDARY STRUCTURE WITH PAIR CODING SCHEME REVISITED

At this point it was decided to return to the pair coding scheme and to predicting the structure of sequence segments of specific secondary structure. The return to this problem was encouraged by the relative simplicity of this approach and the lack of any success in tackling the more complicated problems above. The training and test sets were now derived from the Qian and Sejnowski training and test proteins. It was realised at this point that the pair coding scheme was deficient. The 66 nodes in the input layer, representing 66 possible pairs in a sequence length 10 was far too few, as some segments may not have contained any of these 66 pairs. Some sequences, therefore, would have had input codes that contained 0's only, or had a code that only indicated the number of occurrences of each of the 7 residues (see section 6.1.1). This was partly remedied by discarding the 35 nodes coding for the 7 individual residues and

randomly choosing 104 nearest neighbour pairs, bringing the total number of input nodes up to 170. Even so, some sequences still had inputs codes of 0. Now, instead of coding for beta-sheet individually, two output nodes were selected with target outputs of 1 0 for helix and 0 1 for non-helix. Leaving out the sequences with all 0 inputs, training and test sets were constructed as follows. The training set was constructed from the training set proteins and comprised 938 helix sequences, and 938 non-helix sequences. The 938 non-helix sequences comprised all the non-zero beta-sheet sequences with the rest being non-zero coil sequences. The training set contained as many non-helix sequences as helix sequences, even though there are many more non-helix sequences because, as already mentioned in the previous section, a training set with the number of examples of one feature outweighing that of the other is difficult or impossible for the network to learn (see section 6.6 for a more detailed analysis). This problem does not apply to the test set which comprised of 615 sequences from all the helix, beta-sheet and coil sequences from the test set proteins. Again the target output was 1 0 for helix and 0 1 for non-helix. A sequence for which the first node gave the greater output value was taken to be a helix, and one for which the second output node gave the greater output value, a non-helix. Three networks were trained and tested on these sets: a single layer network, a double layer network with 20 hidden nodes, and a triple layer network with 40 nodes in the first hidden layer and 8 nodes in the second hidden layer. For the single layer a network was trained using a stepsize value of 0.05. The best result on testing was 65%. The error stopped decreasing at a value of 125.3, when 91% of the training set had been learnt; that is, 91% of training sequences had achieved their target output values to the tolerance of 0.5. For the network with 20 hidden nodes, trained using a stepsize value of 0.01, the maximum prediction value was 71% on testing and after 200 training cycles 99% of the training set had been learnt. For the triple layer network, again trained with a stepsize of 0.01, the maximum prediction value was 70% on testing, and after 217 training cycles 96% of the training set was successfully learnt. It is noticeable that both the networks with hidden nodes have comparable results, but the single layer network does significantly worse than

both. The single layer network was also unable to learn more than 91% of the training set and its final error value of 125.3 was much higher even after 1660 cycles of training than for both the networks with hidden nodes, whose error values were 20.9 for the network with 20 hidden nodes, and 70.0 for the network with 40 and 8 hidden nodes, after only around 200 cycles of training. One can conclude that the single layer network is behaving qualitatively different from the networks with hidden nodes. It was realised at this point that it was wrong to leave out the all 0 input codes in testing, as these will inevitably occur if any proper secondary structure prediction is to be attempted. In training, however, these all 0 input codes cannot be allowed, as some of them will have a target of 1 0 and others 0 1, which will cause conflict during training. Testing on all the sequences, 841, instead of 615, gives a maximum prediction success of 72.5%, the correlation coefficient being 0.39. The question that now arises is whether this coding scheme can be improved upon. So far only 170 pairs were used and much vital information is probably lost. Of course for a segment length of 10, the number of possible pairs is too large to be practically implemented on the network. If one makes the restriction of using only nearest neighbour pairs, then one will need 400 input nodes. It is sensible to use nearest neighbour pairs rather than those a number of residues apart as they are more likely to occur than any other pairing in a sequence of finite length. What is more, this method codes for 230 more pairs than our previous method and must consequently contain more information. However, a run using this coding scheme did not improve prediction results. This could be due to the fact that this network had far more weights than our previous network. It is believed that an increase in the number of weights can be detrimental to a network's ability to generalise by allowing the patterns to influence separate regions in weight space thus preventing the interaction of patterns that leads to generalisation. But here we have already seen that the results from a network with 20 hidden nodes and one with a first layer of 40 hidden nodes and a second with 8 hidden nodes did not produce significantly different results. We must conclude therefore the 170 pairs in the previous coding scheme contain on average more significant information than the 400 nearest

neighbour pairs. This makes sense, as 66 of the 170 pairs were selected for their particular favourability for the helix or non-helix structure.

6.4 COMPARING RESULTS

One problem in comparing results from different runs is that all runs with different sets of initial weight values and different stepsizes will be at different stages of learning. This means one cannot simply compare results at specific cycle values, but one must choose either an error value or the percentage of training set learnt. But even these two values may not be appropriate as they both do not reveal the exact percentages of helix and non-helix learnt which may be important if one is comparing prediction results. It is very obvious during training that cyclic fluctuations occur. That is, in one cycle a high percentage of helices and a low percentage of non-helices are successfully learnt, with the next cycle seeing the opposite and so on. This is particularly true during the early stages of learning when fluctuations are large. These fluctuations are also reflected in the test set predictions on the two structures. Overall prediction success on the whole test set is a possible measure in this case, but as the number of non-helices outweighs the number of helices in the test set, fluctuations in favour of non-helices will always have a greater influence on this measure than fluctuations that favour helix prediction. This makes the correlation coefficient a particularly appropriate measure of prediction success during runs. It means that the best method for comparing prediction performance is to compare maximum correlation coefficients.

6.5 PREDICTION WITH SEGMENTS OF SPECIFIC SECONDARY STRUCTURE WITH POSITIONAL CODING SCHEME

Altering the number of hidden nodes and rerunning the same network with different sets of random weights, as well as altering the other parameters did not have a great effect on prediction success with the pair coding scheme on whole structure sequences. Given the obvious importance of the coding scheme used, it was natural at this stage to try the positional coding scheme originally used to predict boundaries. As already mentioned, this coding scheme requires 200 input nodes for a window size 10. Apart from the sequences that gave all 0 input codes, the training set sequences were those used with the pair coding scheme. It contained 1161 helix sequences and 1161 non-helix sequences. The non-helix sequences were derived from the 130 beta-sheet sequences, the remainder being coil. The test set contained 244 helix sequences and 597 non-helix, only 25 of which were beta-sheet. The test set sequences were those used with the pair coding scheme. Again the two output nodes had targets of 1 0 for helix and 0 1 for non-helix. A preliminary result with a 20 hidden node network gave an overall prediction of 76%, with a correlation coefficient of around 0.5, which was significantly better than any result achieved with the pair coding scheme. All the subsequent results are with this same training set and the positional coding scheme.

6.6 BALANCED AND UNBALANCED TRAINING SETS

It has already been pointed out that if the training set does not contain equal numbers of helix and non-helix sequences, then the network is difficult or impossible to train. In this work the number of helix sequences is outweighed by

the number of non-helix sequences and if we are to use only as many non-helix sequences as helix, then all this important information is lost. Despite the likelihood of the network of simply getting stuck in the local minimum that corresponds to all input patterns outputting the target value of the non-helices (0 1), it was decided to attempt to train such a network. The training set comprised of 1161 helix sequences and 2839 non-helix patterns, i.e. there were more than twice as many non-helix sequences in the training set than helix sequences. In fact with a network with 10 hidden nodes, and stepsize value of 0.01 it proved to be relatively easy to train the network such that 99% of the training set was learnt. The maximum prediction success achieved was 77%, with a correlation coefficient of 0.468. Although there was a risk that the network would be unable to learn the training set, the benefit in using a balanced training set is that it actually performs better. Although the overall prediction success is on par with that of the balanced training set, the correlation coefficient is significantly worse: it was 0.5 for the balanced training set. For helix prediction alone the balanced training set achieves a high value of around 78%. The unbalanced training set here, however, only achieves 63% on helix alone. This means that the overabundance of non-helices in the training set is detrimental to helix prediction. This must be because the non-helix examples, due to their greater number, have more influence on the weights than helix examples. This is not the only possible explanation as the balanced training set was learnt by a network with 20 hidden nodes instead of 10. As at this point the effect of hidden nodes with this positional coding scheme had not been investigated, the reason for the difference in our two results could have been due to the number of hidden nodes used. The effect of the number of hidden nodes is investigated thoroughly at a later stage, but to deduce whether a network with 20 hidden nodes could produce results significantly different from that with 10 and on par with those of the balanced training set, two runs with 20 hidden nodes were attempted. At this point the risk that an unbalanced training set would prove difficult to learn was confirmed. Initial attempts with a stepsize of 0.01 all ended in the local minimum mentioned above. However, reducing the stepsize from 0.01 to 0.001 helped overcome this and the network learned the

training set successfully. Again, in both cases a high value of 78% for the overall prediction success was achieved, but the correlation coefficient was at most 0.47, again due to the low prediction value of 60% attained for helices alone. This shows that the extra hidden nodes are not the reason for the balanced training set achieving a better correlation coefficient and that the balanced training set performs better than the unbalanced for the reason given above. It also suggests that hidden nodes do not have a great effect on prediction success as both runs with the unbalanced training set, one with a network with 10 hidden nodes and the other with 20, gave similar prediction values.

At this point it was clear that the variation of the learning parameters and the number of hidden nodes was not going to help attain the desired correlation coefficient of 1. It seemed sensible, therefore, to try to find a way of using all the non-helices that were not being used in the balanced training set, without unbalancing the training set. To this end a training set was constructed of all the helix sequences repeated twice, together with as many non-helices. This meant the training set comprised of 2×1161 helix sequences together with 2322 non-helices sequences, leaving just a few hundred non-helix sequences unused. This is a large training set containing 4644 patterns and a network with 5 hidden nodes, for example, takes roughly 80 minutes to train to 200 cycles. Figure 6.3 shows the correlation coefficient plotted against percentage of training set learnt for six runs with this training set, on networks with 0, 5, 10, 20, 30 and 40 hidden nodes. The most surprising result here is that it does not improve, to any great extent, on the result from the original balanced training set, which achieved a correlation coefficient of 0.5 with a network with 20 hidden nodes. Again there is no obvious dependence on the number of hidden nodes. A network with 5 hidden nodes does better than a network with 30 hidden nodes, but worse than one with 40 or 20 hidden nodes. The network with 0 hidden nodes does worst of all (in section 7.2 this is not found to be significant). Another interesting feature is that all networks achieve a peak when roughly 87% of the training set has been learnt (the reason for this peak forms the main part of later work, see section 7). One

final feature to notice is that the network with 0 hidden nodes shows peculiar peaking behaviour. It is unable to learn more of the training set with further training but actually unlearns it to a slight extent resulting in a comparatively large drop in prediction success (see Discussions and Conclusions page 140). In terms of trying to achieve the best correlation coefficient, the result here shows that the balanced training sets do significantly better than the unbalanced, but the use of more information in including an extra 1161 non-helices in the training set did not produce any significant increase in prediction success. The exact effect of the number of hidden nodes and the size of the training set is investigated thoroughly in section 7.2.

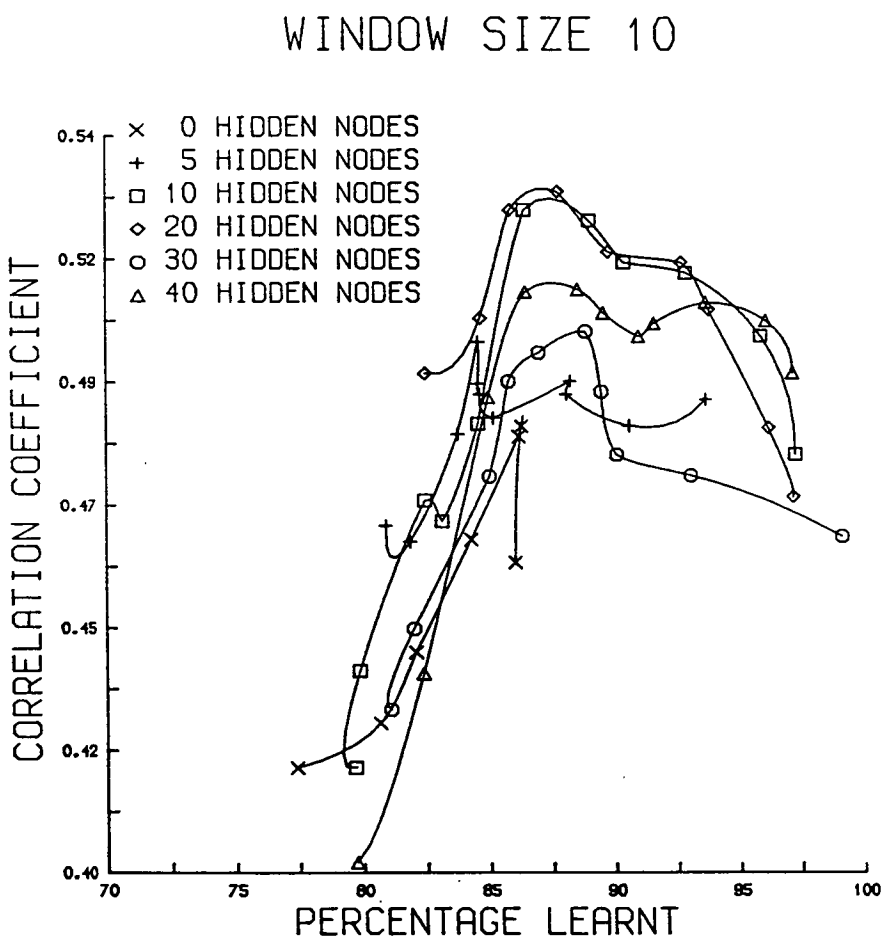


Figure 6.3

Correlation coefficient plotted against overall percentage of training set learnt for networks with 0, 5, 10, 20, 30 and 40 hidden nodes with the artificially balanced training set.

6.7 BETA-SHEET PREDICTION

The main problem in predicting beta-sheet with this method, whereby a whole window length 10 has to be beta-sheet, is that there are very few examples of beta-sheet: 130 in the training set and only 25 in the test set. In order to ascertain how predictable beta-sheet is, it is informative first to train a network to simply distinguish helix from beta-sheet. Again the problem of balancing the training set arises, but as the ratio of beta-sheet to helix or non-helix is far smaller than for helix to non-helix, so the problem of balancing the training set is even more acute. If a network is unable to accomplish this simple task, then it is clear that it will be unable to cope with the more difficult task of distinguishing helix, beta-sheet and coil. The target output for beta-sheet was 0 1, and 1 0 for helix and the test set contained 25 beta-sheet and 244 helices. Trying to train a network to learn the unbalanced training set failed on every attempt. A network with 10 hidden nodes was trained on the balanced training set comprising of the 130 beta-sheet sequences and 130 randomly selected helix sequences. This network could learn to 100%, but as expected only achieved a very poor prediction success, with an overall prediction value of 74%, but with only 52% of beta-sheet being correctly predicted. This best result occurred when 96% of the training set had been learnt and roughly 92% of beta-sheet. Further to this a balanced training set was constructed from beta-sheet sequences repeated 8 times giving roughly the same number as all the helix sequences. Unfortunately a network trained on this training set improved on this result only slightly and behaved rather erratically during training. This result dashes any hopes of doing beta-sheet prediction using this approach and shows that the result in distinguishing helix from non-helix is mainly due to the network's ability to distinguish helix from coil (see section 7.7 for further work on beta-sheet prediction).

6.8 RECIPES FOR LEARNING

Although no thorough investigation of how the learning parameters affect learning was made, in the course of the work up to this point, a fairly successful strategy for the setting of the learning parameters evolved. The initial weight values were set randomly between the values of -0.6 and 0.6. A narrower range of values often made training more difficult, possibly because low weight values put the state of the network in a more distant region of weight space from the region of the ideal solution. As already mentioned the initial setting for the momentum value should be a fairly low one. Certainly an initial value of 0.9 would encourage further any initial step in the wrong direction. An initial setting of 0.3 or lower gives the network a chance to use the pure gradient descent algorithm to find the correct direction for successful learning. After a few cycles of successful learning have taken place it was often sensible then to increase the momentum value to a value of 0.9 or more. Another important parameter is the stepsize. Large stepsize values can send the error spirally upwards. However, setting the stepsize at a too small value can increase learning time by a large factor. It was found here that small networks such as one with 5 hidden nodes could learn quite quickly with a stepsize value of 0.01. However, larger networks, with 40 hidden nodes or more, were unable to learn with such high stepsize values. For these a stepsize of 0.005 or 0.001 was used. For single layer networks a stepsize of 0.1 was possible. Some networks were trained with weight decay to find out whether this improved generalization. If the weight decay factor was set too high learning was totally disrupted. If it was set at a lower value such that learning was not upset, no effect could be seen on prediction performance. In the subsequent work the weight decay factor was set at 0.

6.9 ARE ACCESSIBLE HELICES MORE PREDICTABLE?

Up to this point two output nodes have been used, with a target output of 1 0 for helix and 0 1 for non-helix. Two output nodes were originally chosen so that beta-sheet prediction could be incorporated at a later stage. From the results above, however, it is clear that beta-sheet prediction cannot be realistically incorporated and so a single output node is sufficient to code for helix and non-helix. In subsequent runs a single output node will be used with a target output of 1 for helix and a target output of 0 for non-helix. Here any sequence whose output was greater than 0.5 was taken to be helix, and any sequence whose output was less than 0.5 was taken to be non-helix. In other words a tolerance of 0.5 was used.

At this juncture it was decided, rather than to try to achieve better prediction values, to use the network more as an investigative tool. Initially the network was used to determine whether the accessibility of the helices and non-helices effects predictability. Later the reason for the false prediction of some sequences is investigated.

The lack of success so far in achieving the desired correlation coefficient of 1 may be simply because helices are not predictable from local sequence. In other words remote regions of the polypeptide chain may influence secondary structure in the folded protein. If this is the case some helices may be more predictable than others. Consider an accessible helix on the surface of the protein. This helix is competing with water molecules to form the main chain hydrogen bonds. Now consider an inaccessible helix in the interior of a protein. This is not competing with water molecules, and therefore the formation of a helix will be encouraged, so that main chain hydrogen bonding is satisfied. This argument suggests that accessible helices may contain sequence that more naturally forms helix than inaccessible helices. In other words accessible helices may be more predictable.

The same case can be made for beta-sheet. The opposite argument, however, can be made for coil. Inaccessible coil in the interior of the protein that does not form helices to satisfy main chain hydrogen bonds may contain more naturally coil forming sequence than accessible coil on the surface of the protein, where coil is more likely to found as water molecules can satisfy main chain hydrogen bonding. In other words inaccessible coil may be more predictable.

Using BIPED the relative accessibilities for all the sequences in the training and test sets were determined. The relative accessibility being defined as the accessibility of the residue relative to its accessibility in an extended chain conformation (Chothia, 1976). Figure 6.4 shows the distribution of these accessibilities for helix together with beta-sheet, and coil. The helix sequences together with beta-sheet sequences were then divided into two groups, those of above the average accessibility and those below. The same was done for the coil sequences. The average accessibility in the case of helix and beta-sheet was 194

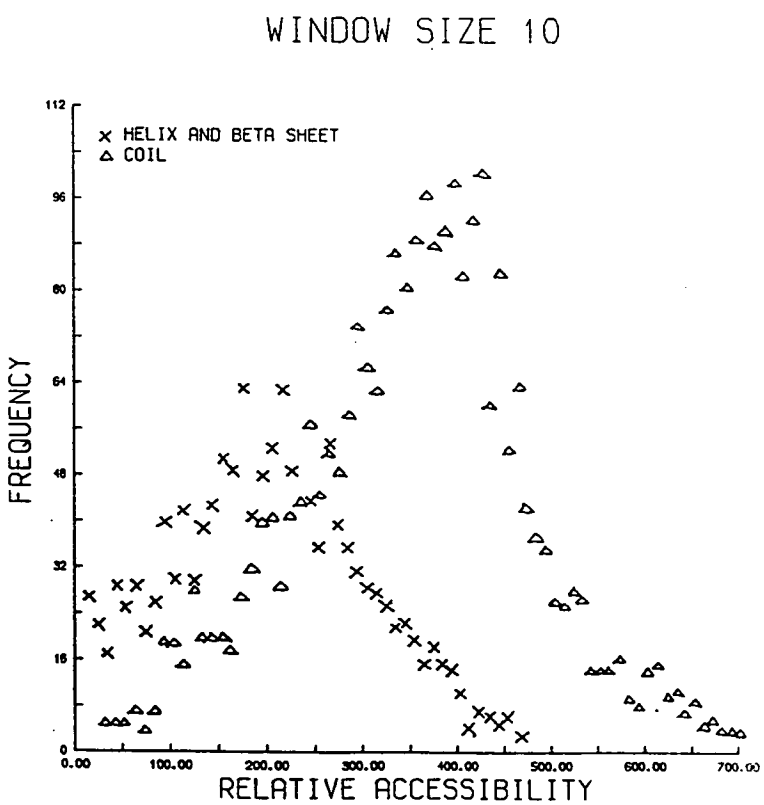


Figure 6.4

and for coil 348. For the training set the above average set of helices and beta-sheet contained 619 sequences and the above average set of coil contained 1251. From the test set there were 127 helix or beta-sheet sequences of above average accessibility and 169 coil sequences. A training set was constructed from the 619 helix or beta-sheet sequences together with 619 of the 1251 coil sequences. The test set contained all the 127 helix or beta-sheet sequences and all the 169 coil sequences. Training a network with 5 hidden nodes gave an overall prediction value of 81.5% with a correlation coefficient of 0.65. The maximum value for helix and beta-sheet prediction alone was 91%. Is this result significant? Later in section 7.7 a network achieves a correlation coefficient of 0.56 on a test set that also groups beta-sheet sequences with helix sequences. This network was trained on twice as many sequences, however, and the expected correlation coefficient from a training set of the size used here would be somewhat than 0.56. To work out whether our result of 0.65 really is significant let us assume that the true correlation coefficient is 0.56, even though the true value is expected to be somewhat lower. From Fisher (Fisher, 1958), the correlation coefficient is normally distributed for large samples and moderate correlations, with a standard deviation given by:

$$\sigma = \frac{1-\rho^2}{\sqrt{n}}, \tag{6.1}$$

where ρ is the correlation coefficient for the whole population, and n is the sample size. Substituting 0.56 for ρ and 296 (=127+169) for n we get a value of 0.04 for σ which means our value lies two standard deviations away from 0.56 and is therefore a significant result. A similar result was obtained, when instead of selecting for accessible coil, a random unselected set of coil sequences was chosen. On testing the percentage of correctly predicted helix and beta-sheet sequences was 92%, the percentage of correctly predicted coil sequences remained at roughly 74%. A training set was then constructed from 620 helix and beta-sheet sequences of below average accessibility together with 620 unselected coil.

Similarly a test set was constructed from 132 helix and beta-sheet sequences of below average accessibility together with the 572 coil sequences. This training set achieved a maximum value of 75% for helix and beta-sheet alone and 80.5% on coil. This means that accessible helix and beta-sheet sequences are more predictable than inaccessible helix and beta-sheet sequences and confirms the idea above. Further work in this area to test whether inaccessible coil is more predictable than accessible proved inconclusive due to the small size of the training and test sets.

7 MAIN RESULTS WITH LAYERED NETWORKS

7.1 PREDICTION DURING TRAINING

From here onwards only the positional coding scheme is used. In addition all training sets are balanced by simply including as many non-helix examples as helix examples in the training set. Again, unless otherwise stated, a window length 10 residues was used, with which the training set comprised of 1161 helix and 1161 non-helix examples of which 130 were beta-sheet. The test set contained 244 helix examples and 597 non-helix examples of which 25 were beta-sheet. A single output node was used with a target of 1 for helix and 0 for non-helix.

Figure 7.1 shows how the error and the prediction success on the test set behave during training for a network with 5 hidden nodes. The error initially decreases rapidly and then more slowly. As the network switches over from the rapid learning phase to the slow learning phase, prediction success peaks. During slow learning prediction success decreases. Figure 7.2 shows test set prediction success for helix and non-helix plotted against, respectively, the percentage of helix and non-helix learnt; where a helix sequence with an output value greater than 0.5 was taken as being learnt, and similarly for a non-helix sequence with an output value less than 0.5. During the rapid learning phase the test set prediction success for the two structures has a linear relationship with the percentage learnt. During the slow learning phase the prediction success for non-helix remains relatively constant, but for helix it decreases significantly.

Figure 7.3 is an analogous plot to figure 7.1 but for a single layer network. Here the correlation coefficient does not decrease appreciably. The error decreases to about the same value at which networks with hidden nodes switch from the fast learning phase to the rapid learning phase, but the single layer network cannot

decrease the error any further and it remains constant during further training.

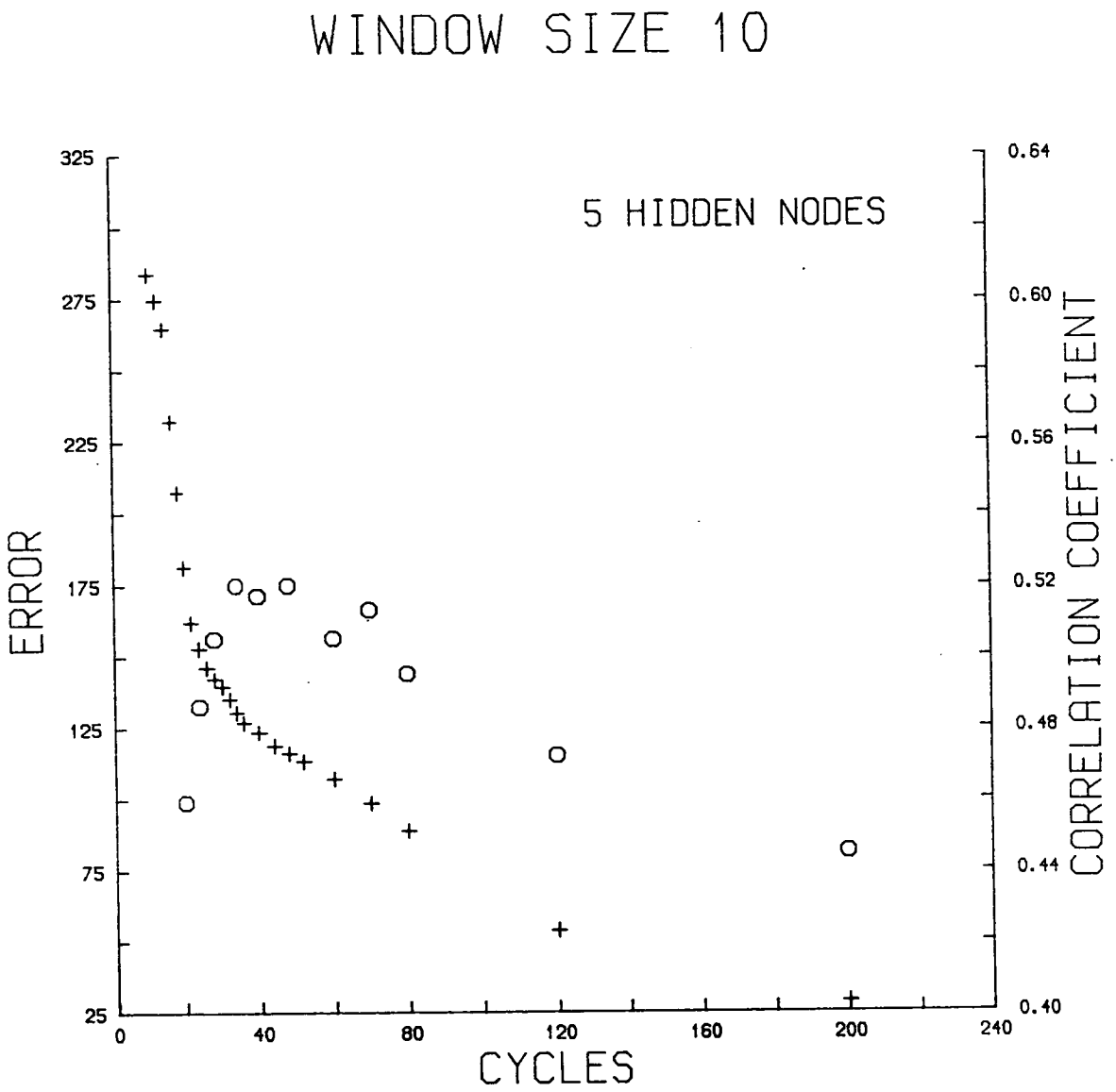


Figure 7.1

Error(+'s) and correlation coefficient of test set prediction(o's) plotted against training cycle for a network with 5 hidden nodes.

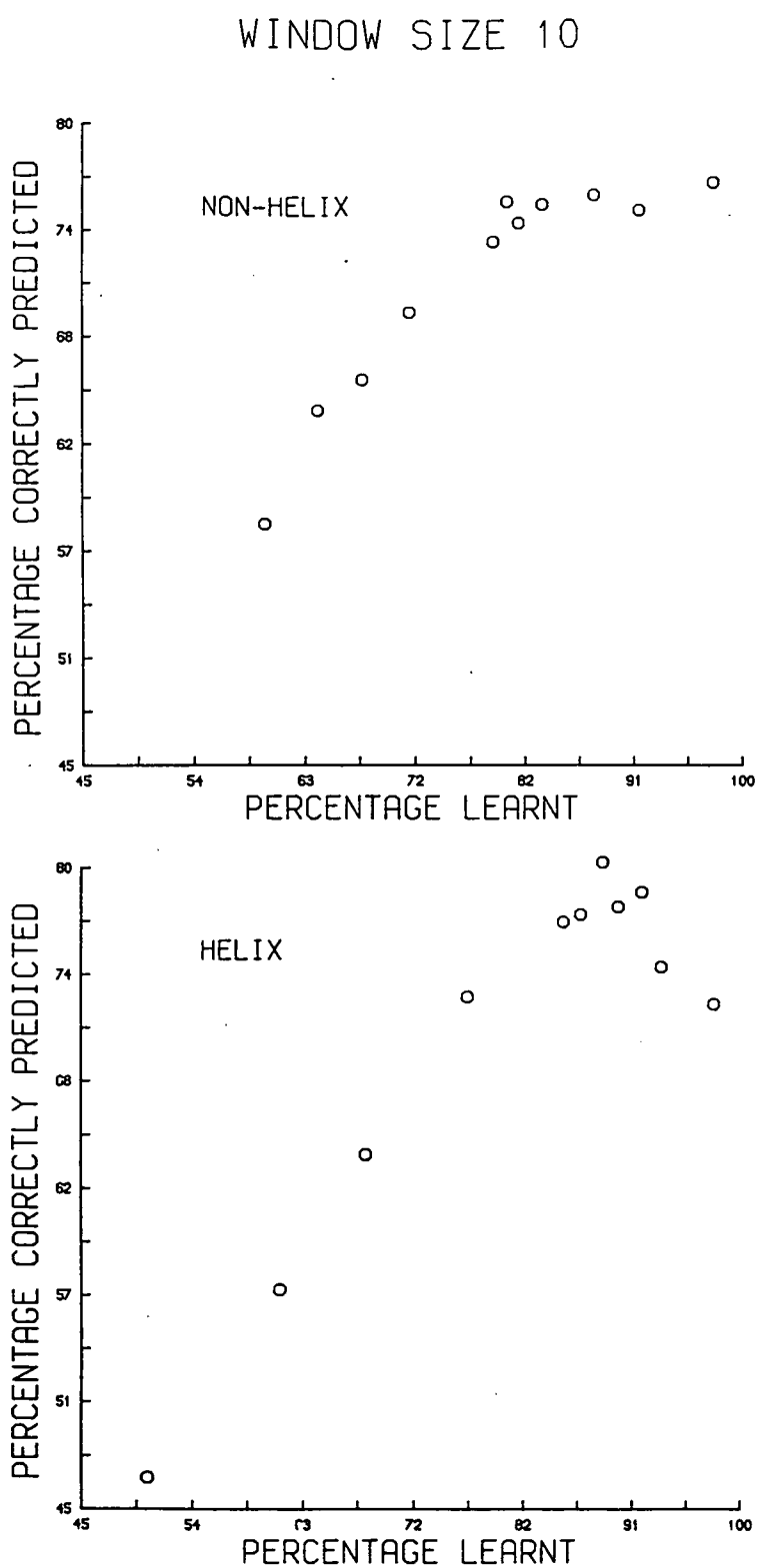


Figure 7.2

Test set prediction success plotted against percentage of training set learnt for non-helices and helices.

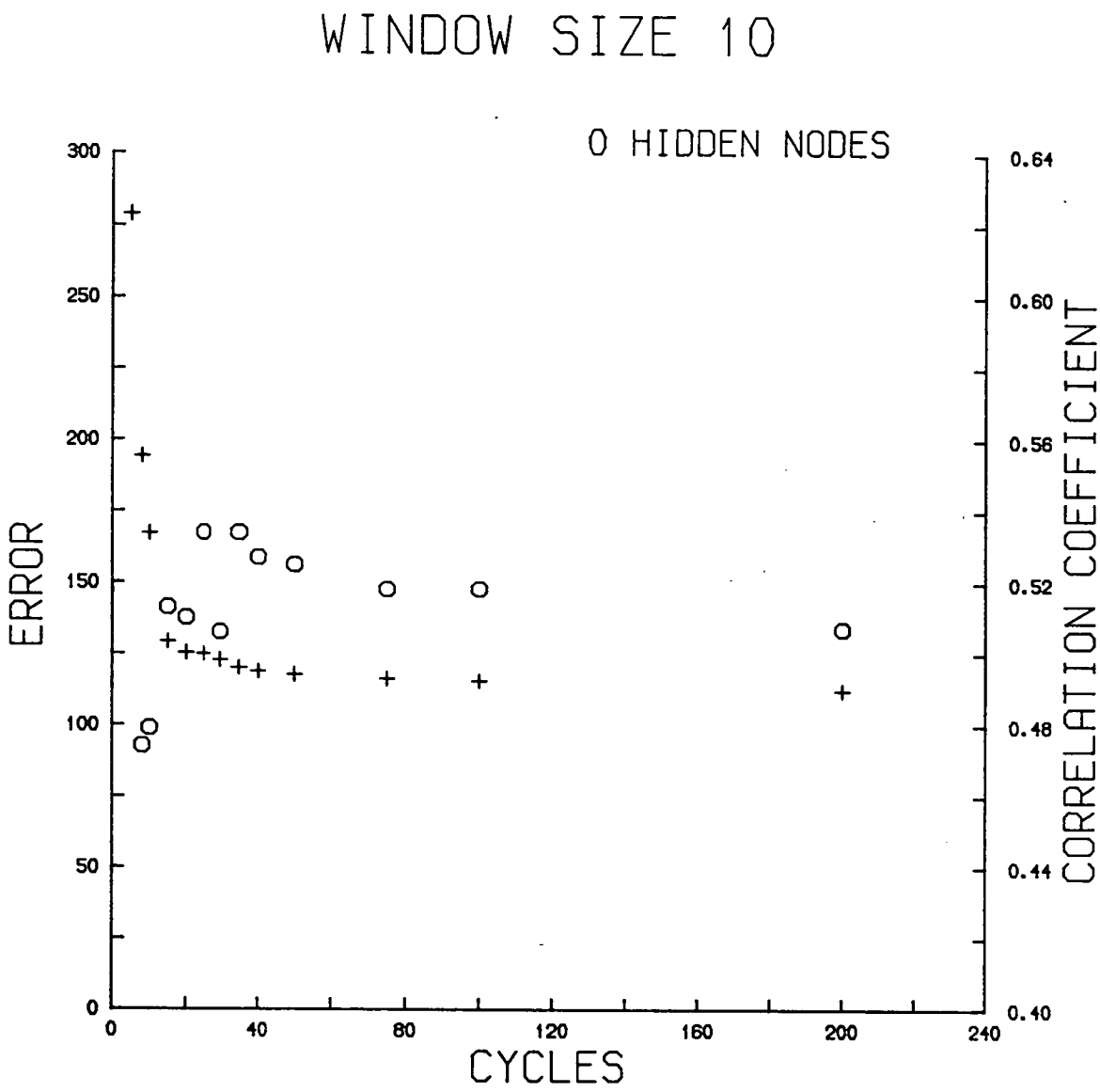


Figure 7.3

Error(+'s) and correlation coefficient of test set prediction(o's) plotted against training cycle for a single layer network.

7.2 EFFECT OF TRAINING SET SIZE AND NUMBER OF HIDDEN NODES

Figure 7.4 shows the effect of the size of the training set for networks with 0, 5 and 20 hidden nodes. Each point in this figure is the correlation coefficient at its maximum value during training. The maximum correlation coefficient in figure 7.1 is represented by the plus for training set size 1 in figure 7.4.

Two main conclusions can be drawn from figure 7.4. Firstly, prediction success does not depend to any discernable extent on the number of hidden nodes. In fact, a single layer network does as well as those with hidden nodes. Secondly, the prediction success plateaus at a correlation coefficient well below the desired correlation coefficient of 1 corresponding to perfect prediction.

7.3 THE WEIGHT VALUES

Given that the prediction success does not depend on the number of hidden nodes, one can easily analyse the weights of the single layer network to determine the degree of helix forming potential for each residue in the ten possible window positions. Figure 7.5 shows the weight values for the 20 amino acids plotted against window position. Negative weight values are anti-helix as they will reduce the input to the sigmoid function at the output, and so help push the output towards the target output for non-helix, which is 0. Similarly positive weight values are pro-helix as they will increase the input to the sigmoid function so pushing the output towards the target output for helix, which is 1. As one would expect, both proline and glycine are strongly anti-helix. Most residues, with the exception of histidine, remain roughly either pro-helix or anti-helix in all window

positions. Histidine is anti-helix at the start of the window and pro-helix at the end.

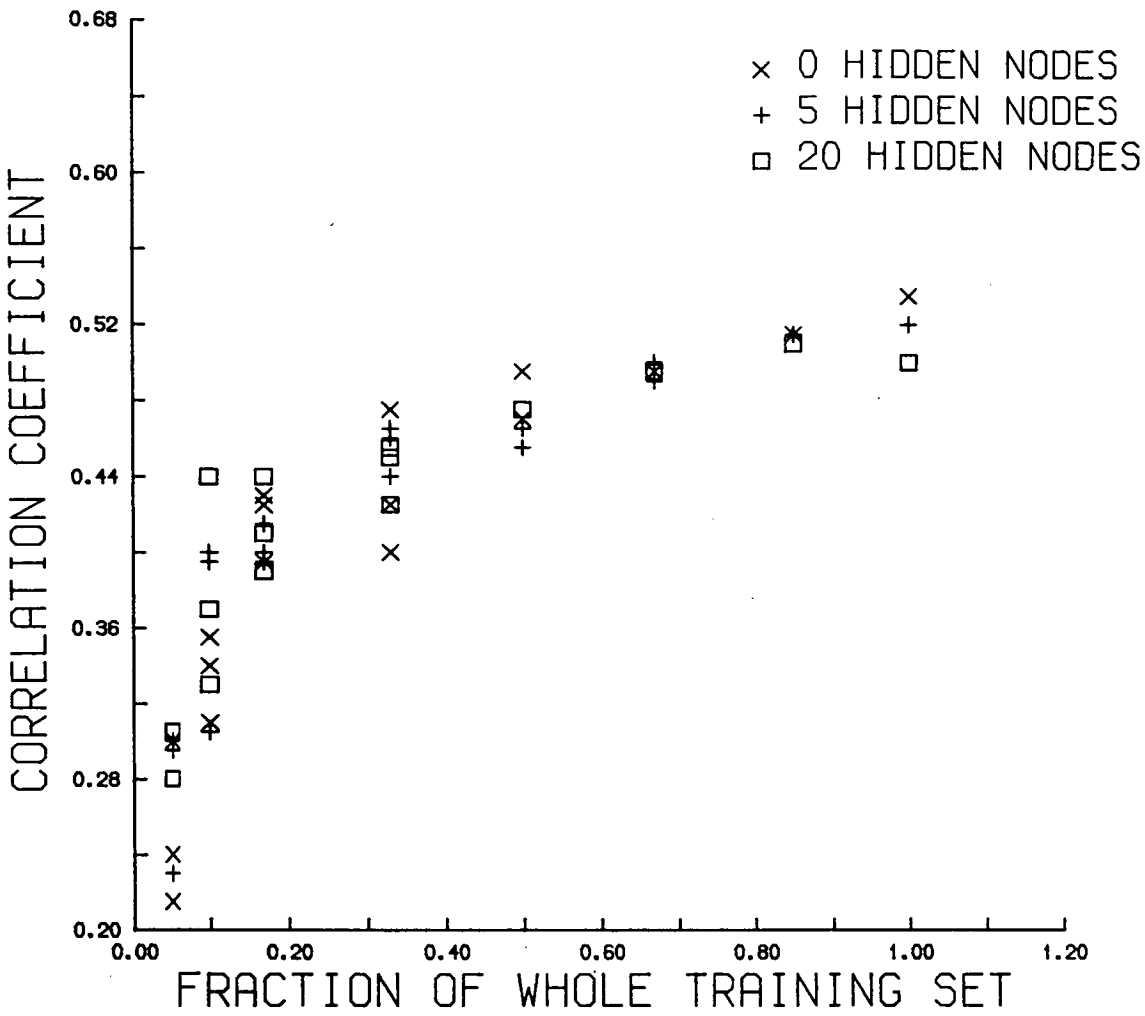


Figure 7.4

Effect of size of training set on test set prediction success. Below training set size 0.5 each network was trained on 3 different training sets derived from the whole training set. At training set size 0.5 each network was trained on the 2 halves of the original training set. At training set size 0.66 each network was trained on two training sets having one half of their patterns in common.

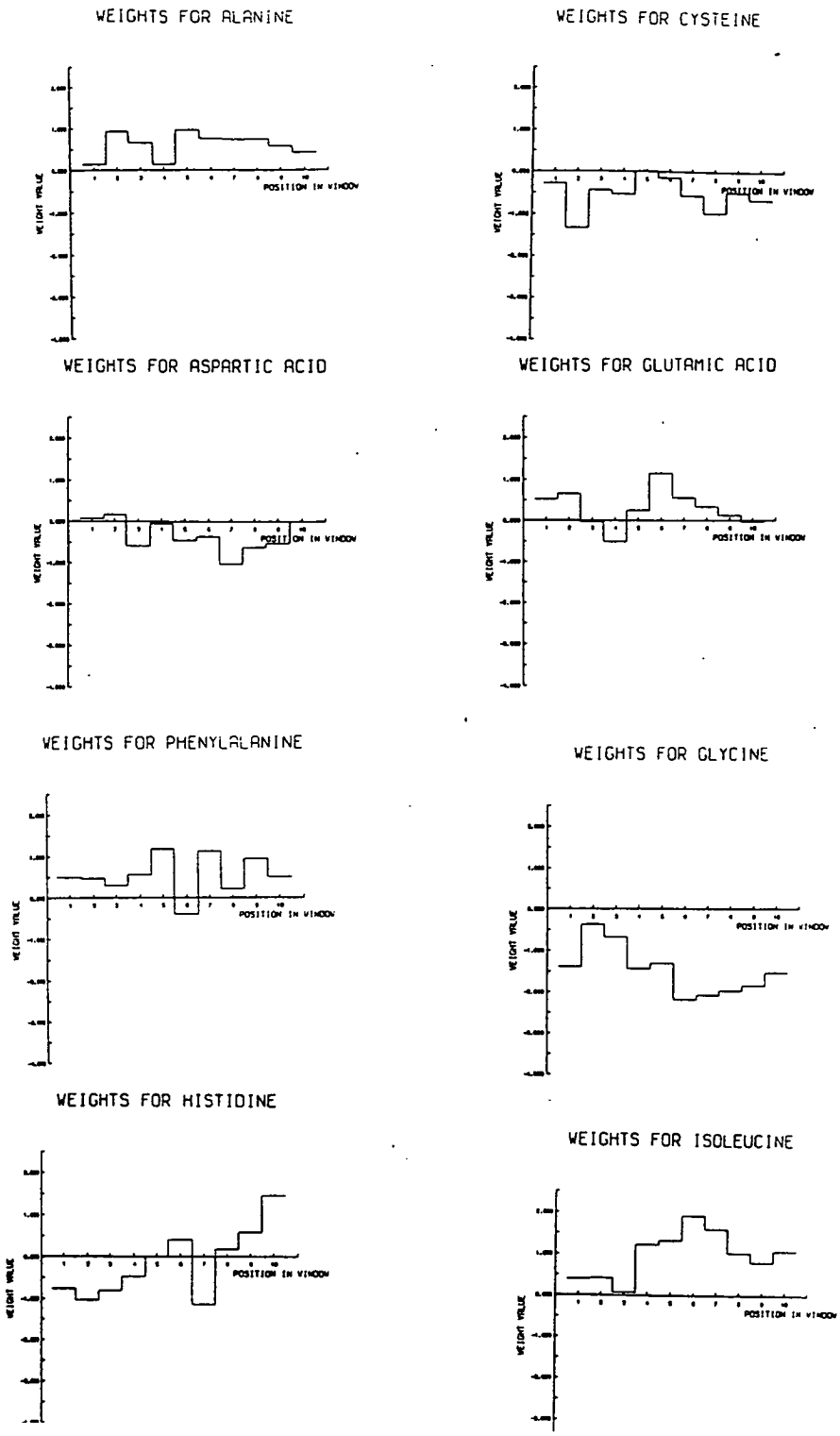
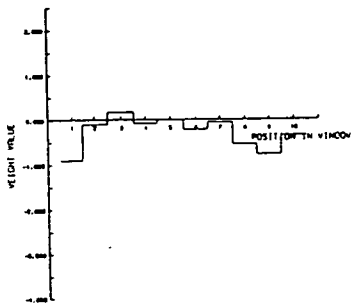


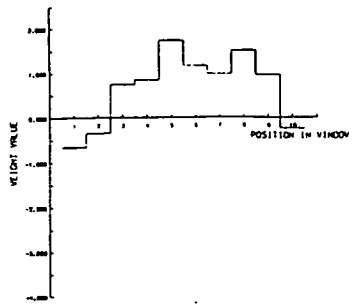
Figure 7.5

Weight values from a single layer network plotted against window position for each of the 20 amino acids.

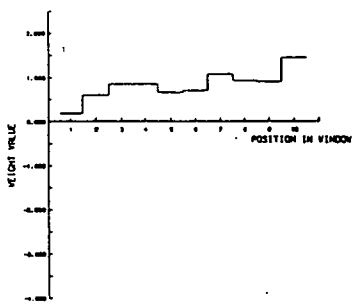
WEIGHTS FOR THREONINE



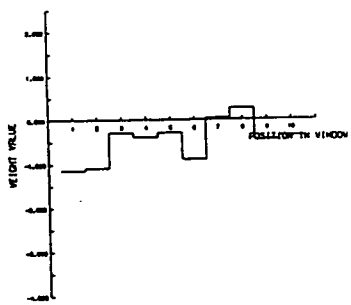
WEIGHTS FOR TRYPTOPHAN



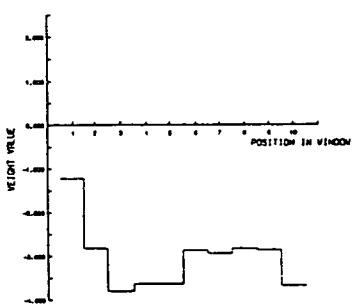
WEIGHTS FOR VALINE



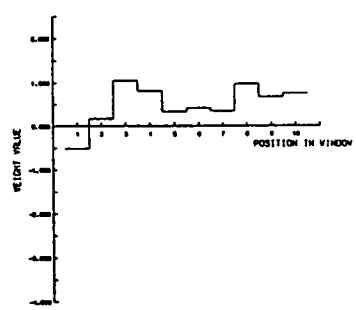
WEIGHTS FOR TYROSINE



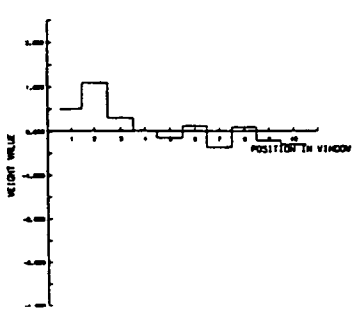
WEIGHTS FOR PROLINE



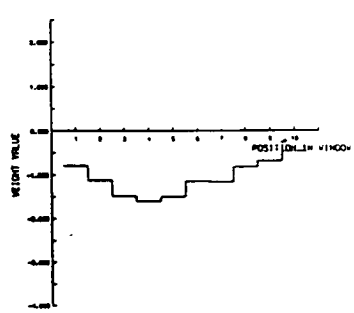
WEIGHTS FOR ARGININE



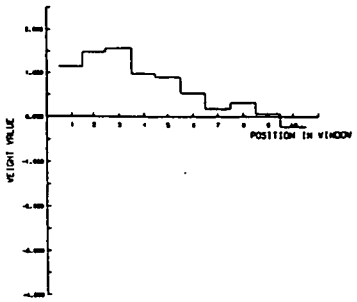
WEIGHTS FOR GLUTAMINE



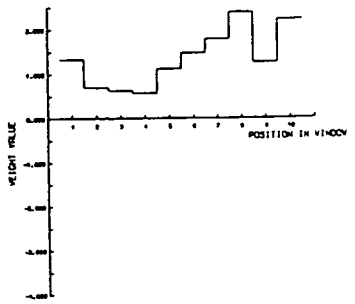
WEIGHTS FOR SERINE



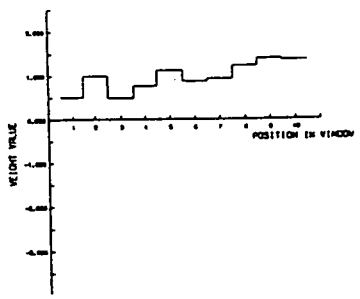
WEIGHTS FOR LYSINE



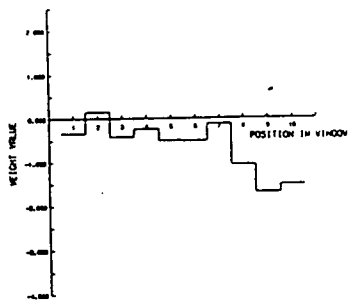
WEIGHTS FOR METHIONINE



WEIGHTS FOR LEUCINE



WEIGHTS FOR ASPARAGINE



7.4 PSEUDO-HELIX AND PSEUDO NON-HELIX SEQUENCES

In neural network circles, over-specialization in the training data is often cited as the cause for a decrease in prediction success. This is sometimes called over-learning (see Discussions and Conclusions page 139). Initially this over-specialization of the weights on some or all of the training examples to the consequent detriment of the network's overall predictive capabilities, was also thought to be the reason for the decrease in prediction success with further learning. However, at the peak of figure 7.1 about 91% helix sequences and 80% of non-helix have been successfully learnt; that is, 9% of helix sequences and 20% of non-helix sequences have not been learnt. For the test set around 80% of helix and 75% of non-helix sequences were correctly predicted or "recognised". The correlation coefficient was 0.52. The fact that a significant percentage of the training set examples are unlearnt at the peak, suggests that the unlearnt examples are the cause of the decrease in test set prediction success with further learning. Although this may seem to be the obvious conclusion to reach, one must bear in mind the following. Even if a pattern is learnt this does not mean that it ceases to effect a change in the weights. Here a training example is taken as being learnt when it has achieved its target to within a tolerance of 0.5. Only after achieving its exact target value, however, will a training example cease to contribute to the error and so have no direct influence on the weight changes. To test whether the unlearnt examples are really the cause of the decrease, the unlearnt examples in the training set were separated from those that were successfully learnt. Figure 7.6 shows the performance of a network trained on only the successfully learnt sequences and tested on test set, which was left untouched. Here prediction success does not decrease and one can achieve only a small increase in the correlation coefficient; it reaches the value of 0.54. This shows that it *is* the unlearnt sequences that are causing the decrease in prediction success. Although figure 7.6 depicts a run with a network with 5 hidden nodes, the successfully learnt

sequences could also be learnt to 100% by a single layer network. This means that the successfully learnt sequences, which are in a large majority, are separated in the input space by a simple decision plane. In the case of helix, this means that the unlearnt and unrecognised helix sequences lie on the opposite side of the decision plane to the majority of helix sequences. Similarly for the case of non-helix, the unlearnt and unrecognised non-helix sequences lie on the opposite side of the decision plane to the majority of non-helix sequence. From here onwards the non-helix sequences on the majority helix side of the decision plane will be referred to as pseudo-helix sequences, and similarly the helix sequences on the non-helix side of the decision plane will be referred to as pseudo-non-helix sequences.

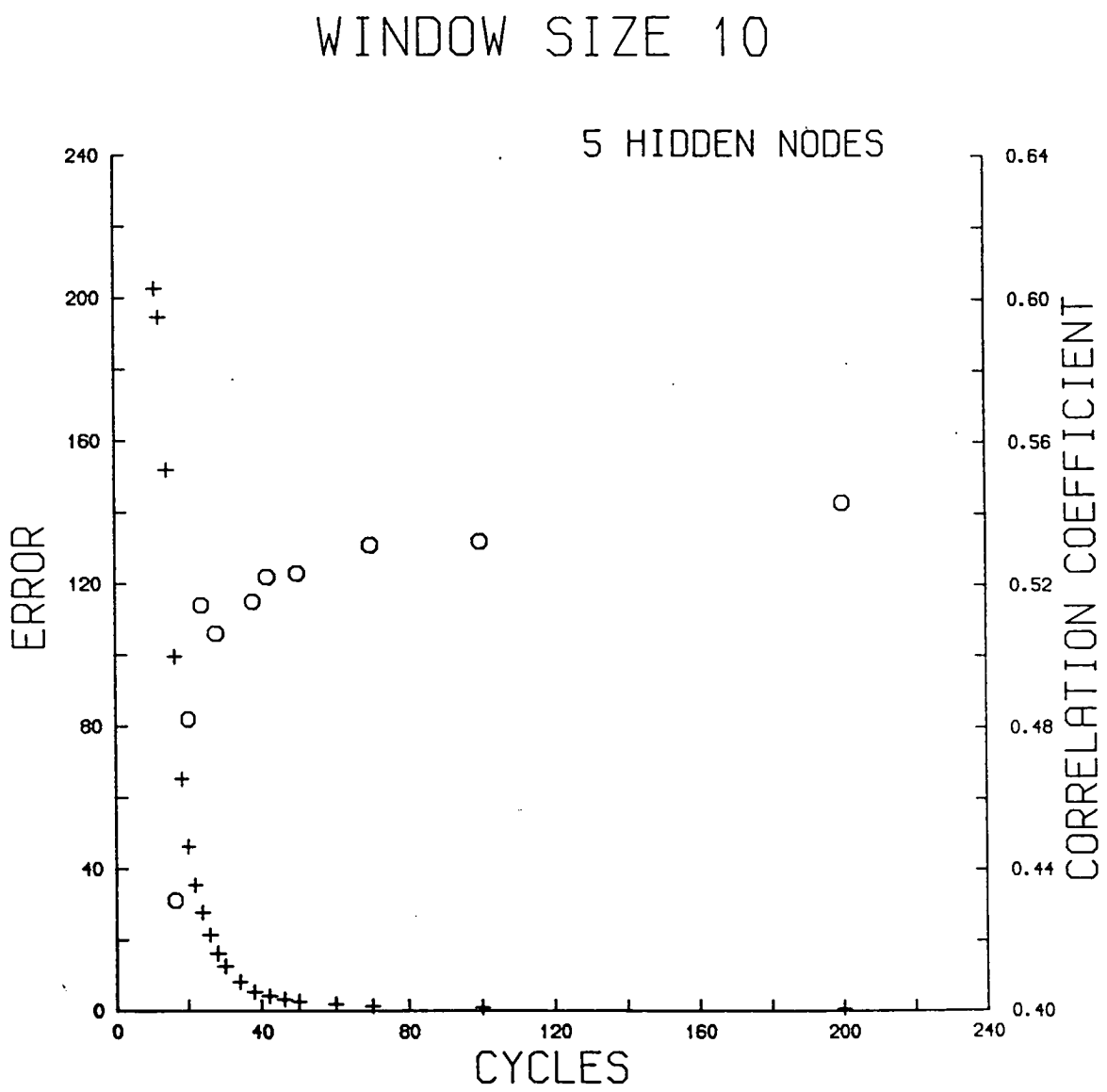


Figure 7.6

Error(+'s) and correlation coefficient of test set prediction(o's) plotted against training cycle for a network with 5 hidden nodes. The training set contains only successfully learnt sequences at peak in figure 7.1; the test set being left unaltered.

7.4.1 Effect of Unlearnt Sequences on Prediction

Although it is now clear that the unlearnt sequences are the cause of the decrease in prediction success, it is not obvious how they effect this decrease. In the case of helix, for example, is it the learning of non-helix sequence in the majority helix region, the pseudo-helix sequence, that causes the decrease, or is it due to the minority of helix sequence, the pseudo-non-helix sequence in the majority non-helix region? It could of course be a combination of both possibilities. The former case would seem to be the more likely as the learning of the pseudo-helix sequence will establish regions of non-helix prediction in the majority helix region. But this still begs the question why helix sequences from the test set occur in these regions of non-helix prediction and not exclusively non-helix sequences. This could be explained if the training set of helix examples occupied a different region in the total input space for helix sequences to the test set examples. If this were so, then learning previously unlearnt helix sequences, the pseudo-non-helix sequences, may shift the decision boundary away from the region occupied by the test set helix sequences and so effect a decrease in helix prediction success. This same question can of course be asked of non-helices. To answer this, a training set was constructed of successfully learnt helix sequences with target outputs of 1 and a joint set of successfully learnt non-helix sequences together with pseudo-helix sequences with target outputs of 0. The test set was constructed from successfully recognised helices and non-helices with target outputs of 1 and 0 respectively. As one can see from figure 7.7 the increase in the number of learnt pseudo-helix sequences in the joint set coincides with the decrease in helix prediction success. This means the learning of the pseudo-helix sequences disrupts helix prediction. The learning of these non-helices, however, leaves non-helix prediction unchanged and so it is reasonable to assume that the learning of the pseudo-non-helix sequences will not affect helix prediction. One can now

conclude that the learning of pseudo-helix sequence is the sole cause of the decrease in helix prediction.

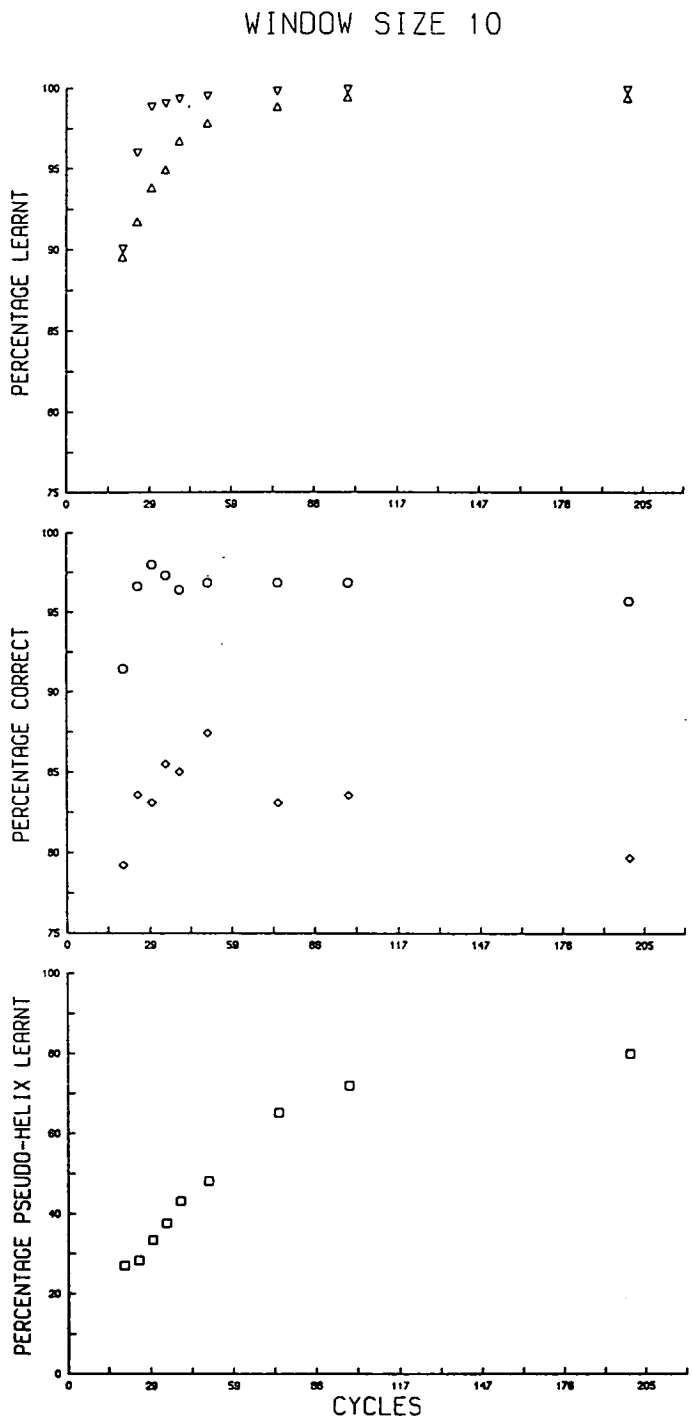


Figure 7.7

Upper plot: percentage of successfully learnt helices(△'s) and non-helices(▽'s) (at peak in figure 7.1) learnt, plotted against training cycle. Middle plot: correctly predicted helices(◇'s) and non-helices(○'s) (at peak in figure 7.1) predicted correctly, plotted against training cycle. Lower plot: percentage of pseudo-helix sequences in training set learnt, plotted against training cycle.

7.4.2 Pseudo-Helix Sequences

As already stated, the pseudo-helix sequences lie on the same side of the decision plane to the majority of helix sequences, i.e. those that were successfully learnt and recognised. The question that naturally arises now is whether the pseudo-helix sequences can be distinguished by a network from this majority of helix sequences. To this end a training set was then constructed from the 236 unlearnt non-helices and 236 successfully learnt helices. A test set was also constructed from the 207 correctly recognised helices and the 153 incorrectly recognised non-helices. At the peak in figure 7.1 these four sets of sequences all had outputs greater than 0.5. A 5 hidden node network was trained to output 1 for the successfully learnt helix sequences and 0 for the unlearnt non-helix sequences. This network could be trained to 98%; that is, 98% of the training set sequences could be learnt. When tested on the test set with the target outputs of 1 for the correctly recognised helices and 0 for the incorrectly recognised non-helices, the result was that only 49% of sequences were correctly recognised, the correlation coefficient being -0.017. The purpose of this experiment was to train a network to distinguish the correctly recognised helix sequences, that represent the majority, from the incorrectly recognised non-helix sequences, the pseudo-helix sequences. It failed. Therefore, the pseudo-helix sequences, which are a subset of non-helix sequences, coil and beta-sheet, are indistinguishable to the network from the majority of helix sequences. If one trains a network on two groups of randomly generated patterns and then tests it on another two groups of randomly generated patterns, one does typically get this result; that is, virtually all the training set can be learnt and the correlation coefficient is 0 on testing. This suggests that the pseudo-helix sequences are randomly distributed amongst the majority of helix sequences (see Discussion and Conclusions pages 126,127, 129-132). Indeed, if one trains a network with the same parameters, but replaces the pseudo-helix

sequences in the training set with randomly chosen successfully learnt helix sequences, and the pseudo-helix sequences in the test set with randomly chosen correctly recognised helix sequences, again the training set can almost be totally learnt, but the correlation coefficient is near 0 on testing. In other words, the experiment to train a network to distinguish pseudo-helix sequence from helix sequence, gives the same result as training a network to distinguish helix sequence from helix sequence. Together, these experiments strongly suggest that the pseudo-helix sequence is scattered randomly amongst the majority of real helix sequence, and is therefore intrinsically identical in sequence to the majority of real helix sequence. These results have two main consequences. Firstly, the fact that pseudo-helix sequences, which are in fact non-helix sequences, are indistinguishable from real helix sequences, puts a real and unavoidable limit on the success of helix prediction with neural networks. Secondly, one is tempted to conclude that these sequences are naturally helix forming but are in non-helix conformations because of global constraints during the formation of tertiary structure. The first of these conclusions is unavoidable, but the second is slightly more speculative given the low number of examples we have used in training and testing (see Discussion and Conclusions for a fuller discussion).

7.4.3 Pseudo-Non-Helix Sequences

The analogous experiment to the one above is to train a network to distinguish the pseudo-non-helix sequences from the majority of non-helix sequences. The training set comprised of 104 unlearnt helices and 104 learnt non-helices. The test set comprised of 37 incorrectly recognised helices and 444 correctly recognised non-helices. These four sets of sequences have outputs less than 0.5 at the peak in figure 7.1. Again a 5 hidden node network was chosen. The result was that, overall, the network correctly recognised 68% of sequences with a correlation coefficient of 0.25. So the network can to some extent distinguish the pseudo-non-

helix sequences from recognised non-helix sequences. Successful non-helices are those that have been learnt or correctly recognised to with the chosen tolerance of 0.5 from the target output of 0. It is noticeable that if one looks at the outputs achieved by the successful non-helix sequences, there are those that very nearly achieve their target of 0, and those that are some way off with much higher output values. Those that do nearly achieve their targets of 0 are those that are rich in anti-helix residues such as glycine, proline and serine. In other words the value of the output is a measure of the degree of helicity or non-helicity of the sequence being presented at the input. Looking at the achieved outputs for the pseudo-non-helix sequences, it is obvious that these, on the whole, have outputs that would put them alongside those successful non-helix sequences with the higher output values rather those with output values near 0. It is likely therefore that the pseudo-non-helix sequences are less distinguishable from the successful non-helix sequences with higher outputs and more distinguishable from those with outputs near 0. The average output value for the pseudo-non-helix sequences is 0.27 with a standard deviation of 0.17. Most pseudo-non-helix sequences therefore will have outputs greater than 0.1 and this threshold was chosen to divide the successful non-helix sequences in both the training and test sets into those that have nearly achieved their target value of 0 and those that have not. Training and a network, again with 5 hidden nodes, to distinguish the pseudo-non-helix sequences from those successful non-helix sequences with output values greater than 0.1, one finds as expected that the network is less able to distinguish the pseudo-non-helix sequences from the selected non-helix sequences: the correlation coefficient being -0.13. Again, as expected, the pseudo-non-helix sequences were distinguishable, to some extent, from the non-helix sequences with outputs less than 0.1, the correlation coefficient being 0.22.

From these results one can now picture the form of the input space. It is illustrated in figure 7.8. This is a 2-dimensional representation of what is in reality a 200-dimensional space with the patterns at the vertices of the unit hypercube. Going from left to right in this figure represents the degree of helix forming potential of the sequence, and the bottom line of the figure can consequently be regarded as an axis representing the output strength. The shaded areas represent areas of non-helix sequence, the white areas helix sequence. The sequences on the extreme left are those containing a high proportion of anti-helix residues, such as proline, glycine and serine, and have outputs close to their target of 0. Moving further right the outputs increase. At the decision boundary, represented by the continuous straight line in the figure, the output value is 0.5 which is the decision value of the output. Moving further right still, one reaches those sequences, which favour helix formation with outputs increasing from 0.5 at the boundary up to 1, the target value for helix sequences, at the extreme right hand edge of the figure. The sequences here are strongly pro-helix. In the predominantly white area to the right of the decision boundary, representing the majority helix region, the pseudo-helix sequences are scattered in a random fashion, in accordance with the result of section 7.4.2. As these are non-helix sequences they are represented by shaded islands. The majority non-helix region is to the left of the decision boundary. Here the pseudo-non-helix sequences, which are helix sequences, are represented by white islands. In section 7.4.3 it was shown that the pseudo-non-helix sequences could be distinguished to some extent from those non-helix sequences with outputs less than 0.1. Therefore the region on the further left of the figure is much less densely populated by the pseudo-non-helix sequences than the region nearer the boundary, representing those non-helix sequences with outputs greater than 0.1, but less than 0.5. During training the simple decision boundary is first established and then, provided the network has some hidden nodes, the pseudo sequences are bounded. The bounded pseudo sequences are the islands in the

figure. For the full explanation as to why the bounding of the pseudo sequences leads to a decrease in prediction success as depicted in figures 7.1 and 7.2, please refer to the Discussion and Conclusions pages 127 and 128. Figure 7.8 should be regarded as the major result of this work. It can explain, in conjunction with the manner in which a back propagation network learns, figures 7.1, 7.2, 7.3 and 7.4. If one accepts that the pseudo-helix sequence is indeed potential helix forming sequence, figure 7.8 also has a very plausible physical explanation. This explanation is also given in the Discussion and Conclusions page 135.

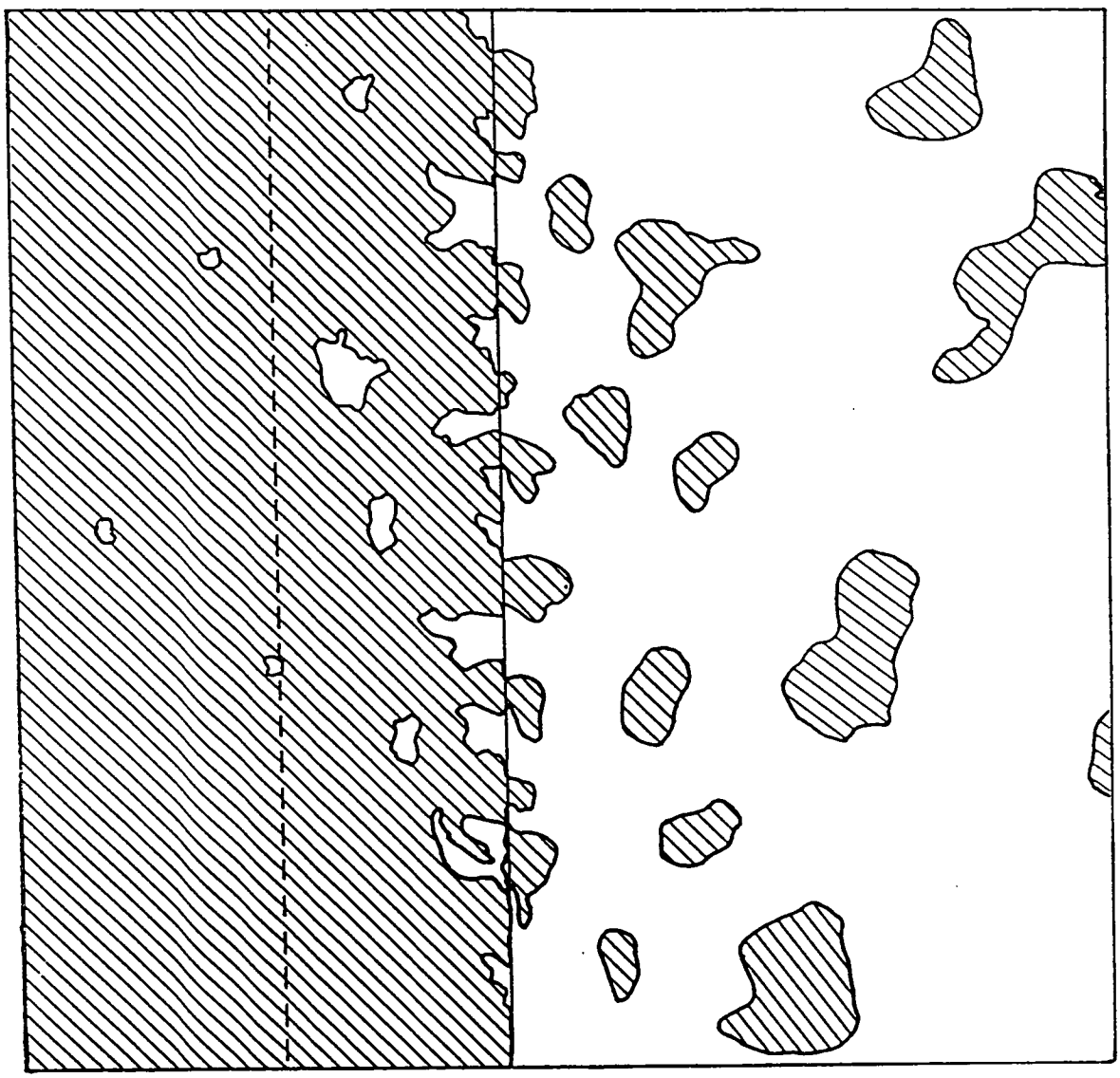


Figure 7.8

Schematic 2-dimensional representation of input space for helices and non-helices in training set. Shaded areas show non-helix regions, white areas helix regions. During training the continuous straight line is established first. After further training, islands of non-helix prediction are established in the majority helix region and islands of helix prediction are established in the majority non-helix region. See text for further explanation.

7.6 SHIFTED DECISION BOUNDARY

These results put a limit on the success of helix prediction. One can, however, achieve some certainty in prediction, but at a cost. If one moves the boundary to the left past the majority of pseudo-non-helix sequences then virtually all the sequences to its left will be non-helix (see dashed line in figure 7.8). This means that whenever the prediction is on the left of this line one can be virtually certain that one has predicted non-helix correctly. The cost, however, is that one will be very uncertain of the result when the prediction is on the right of the line. To achieve this shift in the boundary the following sequences were trained with a target output 1: successfully learnt helix sequences, pseudo-helix sequences, pseudo-non-helix sequences and those successful non-helix sequences with outputs greater than 0.1 in the original training. The remainder of the non-helices were trained to output 0. This single layer network was tested on the original test set. It predicted 94% of helix correctly, but only 50% of non-helix correctly. Before these two figures were 80% and 75%. As well as the percentage of correctly predicted helix or non-helix, another useful measure is the accuracy of the prediction which, in the case of helix, is the percentage of helix predictions that are indeed helix. This measure can give a false impression as the number of non-helices is greater in the test set than the number of helices (see section 2.4) but is useful for comparison purposes. In the original case the prediction accuracy for non-helix is 91%, for helix 57%. With the new shifted boundary it is 95% for non-helix, 43% for helix. So for this 4% increase in prediction accuracy for non-helix quite a large 14% loss in prediction accuracy for helix has been made. One can, now however, be fairly certain that whenever non-helix is predicted it is indeed non-helix.

7.7 BETA-SHEET PREDICTION AND DISTRIBUTION IN THE INPUT SPACE

As already mentioned part of the problem in predicting beta-sheet is that there are relatively few examples of beta-sheet 10 residues in length: 130 in the training set, and only 25 in the test set. It is very noticeable that the network has greater difficultly in predicting beta-sheet correctly. By looking at the percentages of all three structures either side of the original decision boundary and the shifted decision boundary one can get some idea of how they are distributed in relation to each other in the input space. Table 7.1 summarises the percentages of each of the three structures in the three regions bound by the two decision planes and figure 7.9 illustrates this. These values were determined from the training set. Beta-sheet seems to have a distribution in the input space that lies roughly between helix and coil and is more abundant than either in the region between the two planes.

SECONDARY STRUCTURE	LEFT OF SHIFTED BOUNDARY	BETWEEN SHIFTED AND ORIGINAL BOUNDARIES	RIGHT OF ORIGINAL BOUNDARY
HELIX	2%	7%	91%
BETA-SHEET	25%	37%	38%
COIL	60%	25%	15%

TABLE 7.1
See text for explanation

These results together with the few number of examples makes efficient beta-

sheet prediction impossible.

In addition to this, the beta-sheet were grouped with the helices, rather than coil. The training set consisted of 1291 structured sequences and 1291 coil or unstructured sequences. The test set consisted of 269 structured sequences and 572 unstructured. A network with 5 hidden nodes was trained and tested with these sets and a maximum correlation coefficient of 0.56 was achieved. The result obtained when the beta-sheet sequences were grouped with coil has an average correlation coefficient of 0.52 with a standard deviation of 0.02. So the value 0.56 is significantly better (outside two standard deviations from the mean). At the peak of prediction success in the original case only 80% of non-helix has been learnt. In this case around 85% of coil has been learnt. This is because such a large proportion of unlearnt non-helix consisted of beta-sheet. Similarly in the original case 91% of helix was learnt at the peak, whereas here roughly 88-89% of the structured sequences has been learnt. In addition further learning causes a larger decrease in prediction success than here with the structured case. This all suggests that beta-sheet are more naturally grouped with helices than coil.

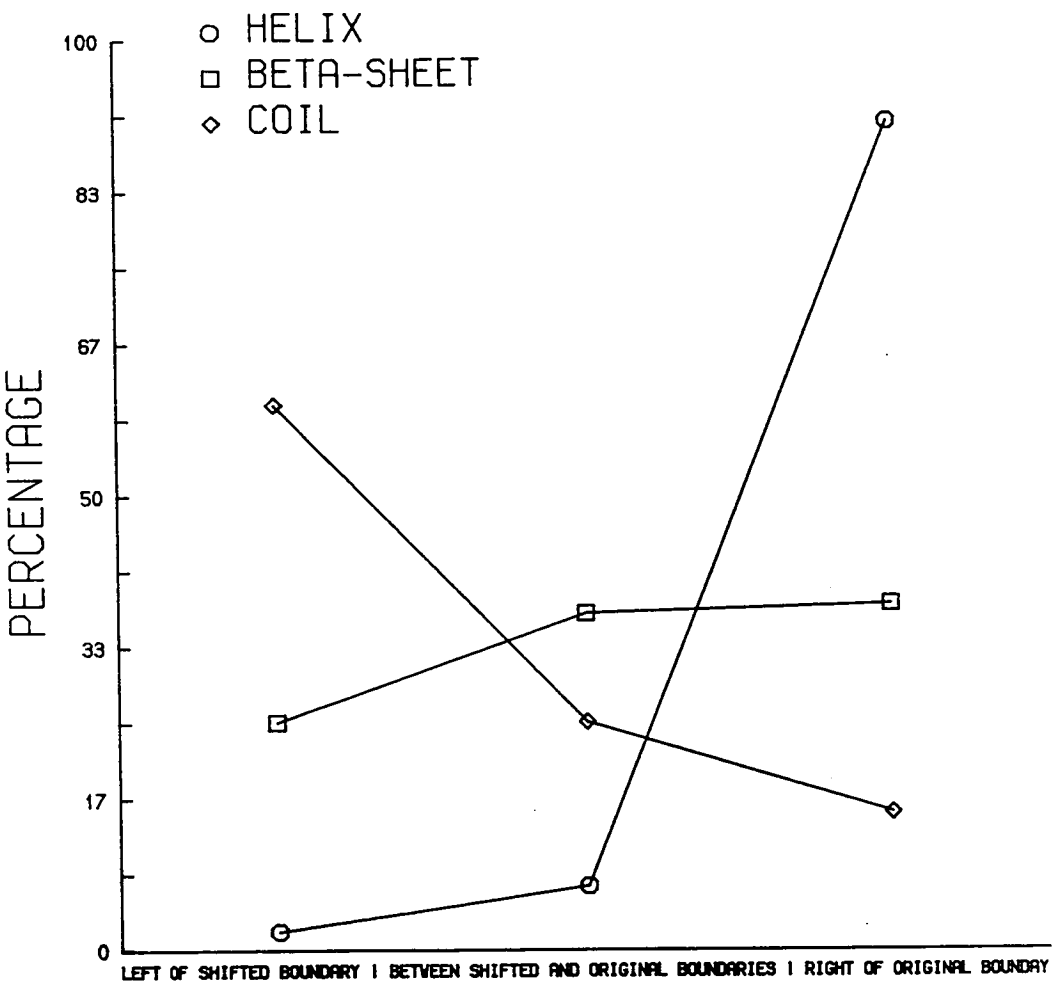


Figure 7.9

Distribution of helix, beta-sheet, and coil in the input space as depicted in figure 7.8.

7.8 PREDICTION FOR INDIVIDUAL RESIDUES

Up till now a rather artificial case has been dealt with: prediction of segments 10 residues in length that are wholly helix or non-helix. Ideally one wants prediction for a single residue. As the window slides along the protein, each residue will be within the window on ten occasions unless it is at or near the beginning or end of the protein. The window will also now contain boundaries between helix and non-helix. As long as the residue is not at or near the end or beginning of the protein, one can imagine that it has 10 predictions associated with it as the predictive window slides across it. These ten predictions can be used to make a prediction for the single residue alone.

7.8.1 Simple and Biased Averaging

The first method tried was simply to average these ten predictions. If the average was greater than 0.5 this was taken as a prediction for helix, less than 0.5 for non-helix. Other methods involved biasing in favour of the predictions where the residue is at or near the centre of the predictive window. To deal with the residues at the N- and C-termini of the proteins, they were imagined to be flanked by 9 residues. Any window containing these flanking residues was assigned a prediction of 0.5, which is not preferential to the prediction of either structure. Biased averaging always did slightly worse than simple averaging. The best result obtained was 73% with a correlation coefficient of 0.34 (see Appendix B). More explicitly 76.5% of non-helix and 61% of helix was predicted correctly. In terms of prediction accuracy this gives 45% for helix and 86% for non-helix. For the network with the shifted decision boundary 40% of non-helix and 90% of helix were predicted correctly and in terms of accuracy this gives 92% for non-helix

prediction and 32% for helix.

Other training runs sometimes did better on predicting helix at the sacrifice of predicting non-helix. For example one network achieved 67% on helix and 69.5% on non-helix. This corresponds to 69% overall with a correlation coefficient of 0.32. The over-prediction of one structure to the sacrifice of the other can be remedied by changing the threshold from 0.5 to some other value. If it is non-helix being over-predicted then lowering this threshold will reduce non-helix prediction and increase helix prediction.

7.8.2 Second Network Method

To try to improve on these results a second network was used. The ten outputs from the first network used for averaging were now used as input for the second network and the target output was 1 or 0 depending on whether the residue was helix or non-helix. Training the second network was done using the training set outputs from the trained first network as input, the helix and non-helix assignments in the training set determining the target outputs. To test, the outputs of the test set from the first network were used as input for the trained second network. The resulting outputs from the second network were the final predictions. The idea behind this method is that as the window slides across boundaries a second network can be trained to recognise the resulting signatures. Unfortunately it was impossible to train the network even if it had a large number of hidden nodes. The best result was achieved by using a single layer network starting with constant weights. Rapid learning took place until the prediction success on each of the structures was comparable: 67% helix, 69% non-helix. Thereafter the results worsened. This method barely improved upon simple averaging with the threshold altered to balance prediction.

7.9 EFFECT OF WINDOW SIZE

7.9.1 Training and Testing

All the work described above was done using a window length 10 residues. To see how window size effects these results window sizes 7 and 13 were also used. Using a longer window causes a problem in that by demanding that the segments be wholly helix or non-helix, the fewer examples give poorer statistics and so we did not go beyond a window size 13. During training the same peaking effects were seen for both window sizes (see figures 7.10 and 7.11). For the window size 7, 87% of helix and 71% of non-helix have been successfully learnt at the peak of prediction success. For the window size 13 these two figures are 94% and 90%. These values lie either side of those for window size 10. The reason that the peak for window size 13 is less pronounced is that there are fewer pseudo-helix sequences and pseudo-non-helix sequences in the training set to upset prediction success. The larger number of pseudo-helix sequences and pseudo-non-helix sequences is also the reason that the decrease in prediction success for window size 7 is greater. On the whole the same regions of pseudo-helix sequence and pseudo-non-helix sequence were found as with the window size 10. Figure 7.12 shows the effect of the size of the training set on prediction success for the three window sizes. Again, no dependence on the number of hidden nodes was found. Window size 13 does better than 10 for the same number of sequences in the training set and 7 does worse than both. The difference in performance is far greater between 7 and 10 than it is between 10 and 13.

7.9.2 Is Difference in Prediction Success due to Boundary Regions?

One possibility for the large difference between prediction success for windows 7 and 13 is that some windows of length 7 will contain proportionally more boundary residues of helix and non-helix than a window length 13, and that these residues will effect prediction detrimentally. In other words, if one takes those sequences of 7 residues that are within 3 residues of the C- and N- termini of helices out of the training and test sets, does this increase prediction success to a value comparable with that for window size 13? To test this, the middle 7 residues from each window of 13 were taken to train and test. This network did significantly worse than the network trained and tested with the original sequences of 13, giving a value comparable with the original set of sequences length 7. This shows that boundary residues are not the reason the results with window size 7 are worse than those for window size 13. This experiment shows that taking the 3 residues either side of the middle 7 into account has a large effect on prediction success. But as one can see from figure 7.12 the inclusion of the extra 3 residues in going from window size 7 to 10 has a much larger effect than then further inclusion of another 3 residues in going from window size 10 to 13.

7.9.3 Prediction on Individual Residues

For individual residues simple averaging and the second network give fairly similar results. Simple averaging with a threshold of 0.5 under-predicts helices for a window size 13. This is because short helices are being overwhelmed by the rest of the residues in the window being in the non-helix region. This problem does not arise for window size 7. For window size 7 simple averaging with a threshold 0.52 achieves 68% on helices and 67% on non-helices. Using the second neural

network gives 67% and 66% for helix and non-helix respectively. For window size 13 the results for simple averaging were 67% and 68% with a threshold of 0.435 and using the second network 67% and 69% for helix and non-helix respectively. It should be noted that all three window sizes yield roughly the same prediction values when tested on individual residues.

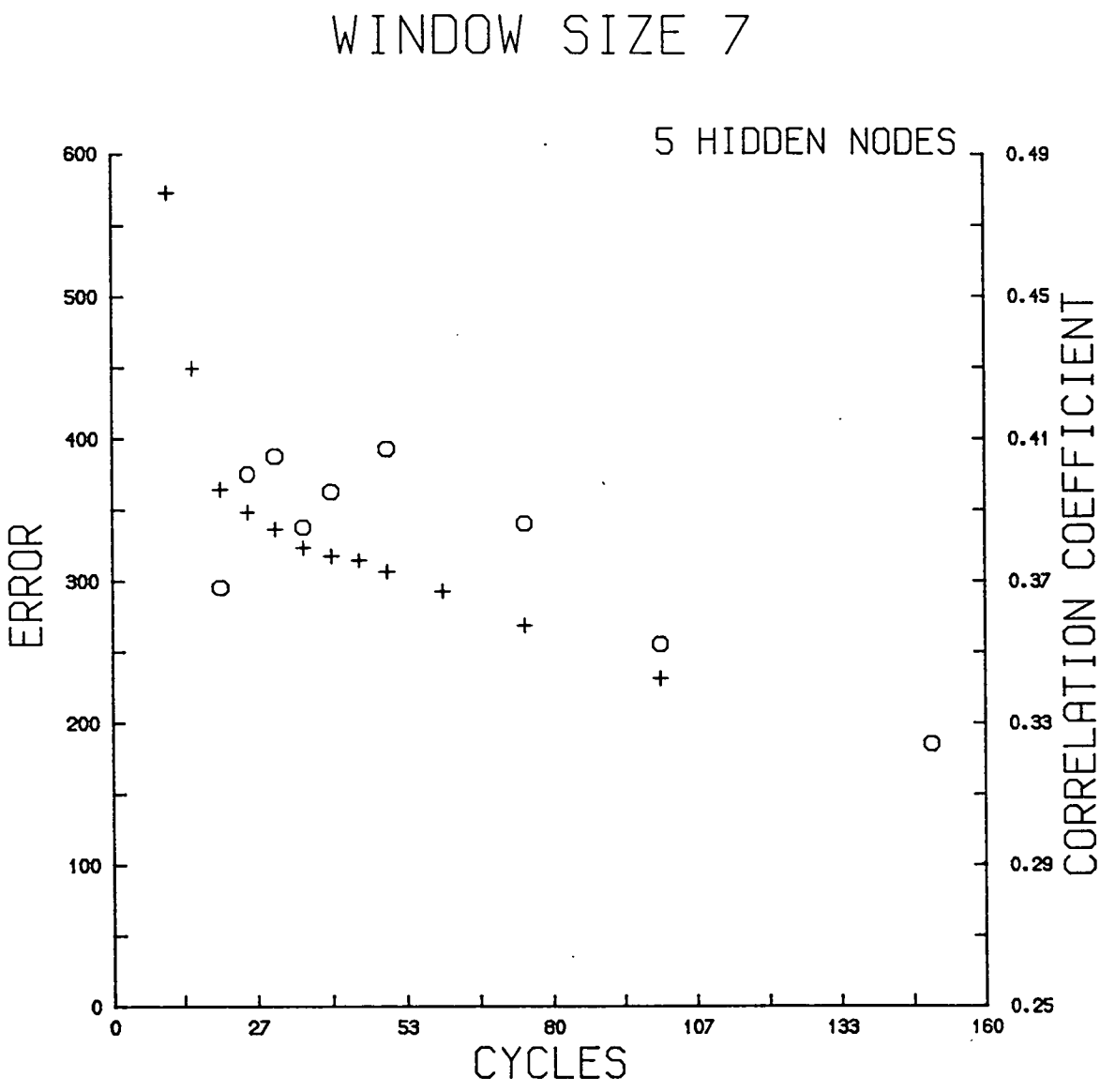


Figure 7.10

Error(+'s) and correlation coefficient of test set prediction(o's) plotted against training cycle for a network with 5 hidden nodes with data from a window size 7.

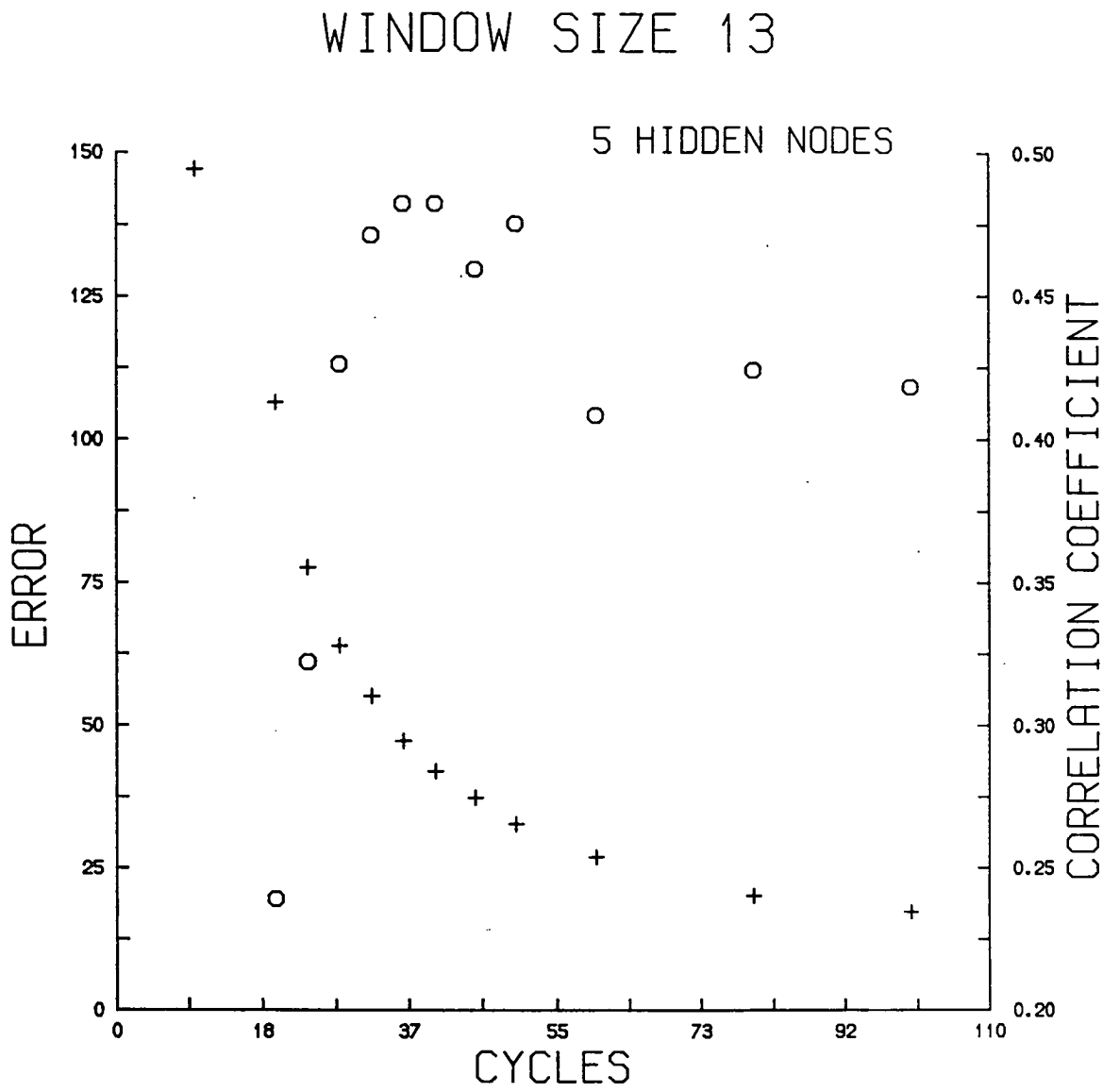


Figure 7.11

Error(+'s) and correlation coefficient of test set prediction(o's) plotted against training cycle for a network with 5 hidden nodes with data from a window size 13.

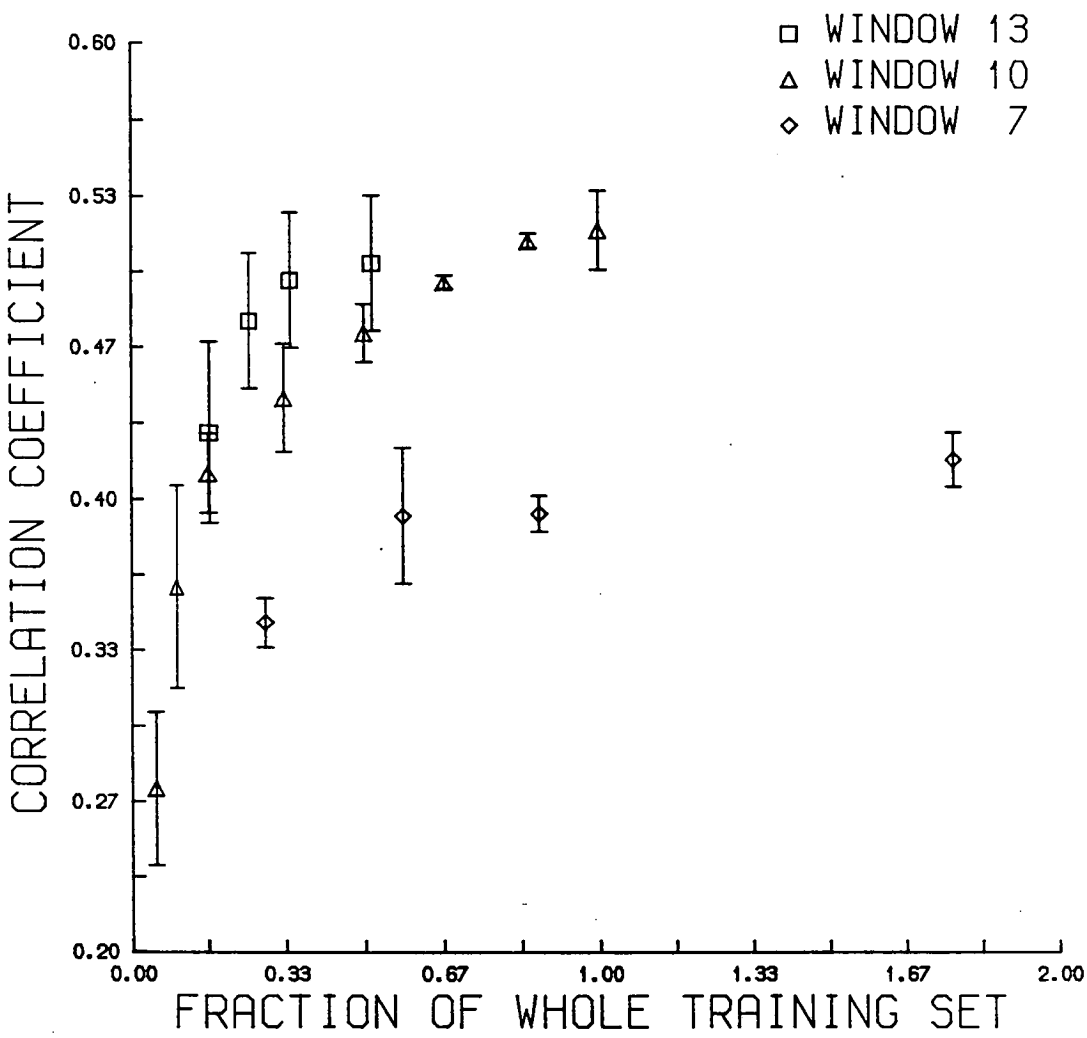


Figure 7.12

Effect of size of training set (fraction is based on training set for window size 10) on test set prediction success for the 3 window sizes tried: 7, 10, and 13. Error bars are two standard deviations in length.

7.10 SIGNIFICANCE OF PSEUDO SEQUENCE

7.10.1 Are Pseudo Sequences due to Errors in Structure Determination?

The main result of this work is the discovery of sequences of non-helix that are indistinguishable from actual helix sequences and the form of the input space as represented in figure 7.8. One possible explanation for these regions is that they are caused by errors in the structural data. Indeed one region of pseudo-helix sequence was found to be actually in the helix conformation in an improved determination of lysozyme (2LZM) in a more recent edition of the Brookhaven data bank. However, pseudo-helix sequences for window size 10 are found in around 70% of the proteins used in training and testing, which would suggest that these regions are not due to structural errors. To investigate this further, only those proteins whose structures were determined to a resolution of 2.5 Å or higher were selected for training and testing. Also, proteins whose sequences in Brookhaven showed a large discrepancies with their sequences in the protein sequence database NBRF (Lesk *et al.*, 1989) were rejected. The Brookhaven codes for these proteins and their resolutions are shown in table 7.2. Training and testing with this set showed again the tell-tale peaking effect, thus showing that pseudo sequences are probably not due solely to errors in the structural data.

There is a further important feature that enforces the idea that pseudo-helix sequence is potential helix sequence. As the predictive window slides along the protein sequence, windows of pseudo-helix sequence are quite often predicted consecutively. Thus regions of strong pseudo-helix sequence are established. What is more, many of these regions are found with all the three window sizes used. The pictures in figure 7.13 show such regions. These often resemble helices in structure, suggesting that pseudo-helix sequence is potential helix sequence that

TRAINING SET		TEST SET	
1INS	1.5	2LHB	2
1HIP	2	2CDV	1.8
1GP1	2	2ALP	1.7
1FX1	2	2ACT	1.7
1FDX	2	1PPD	2
1FDH	2.5	1NXB	1.3
1EST	2.5	1HMQ	2
1ECD	1.4	1ACX	2
1CYC	2.3	1ABP	2.4
1CY3	2.5		
1CRN	1.5		
1CPV	1.8		
1CCR	1.5		
1BP2	1.7		
2LH1	2		
2KAI	2.5		
2GN5	2.3		
2CYP	1.7		
2CCY	1.6		
2CAB	2		
2APP	1.8		
1TIM	2.5		
1TGS	1.8		
1SN3	1.8		
1RN3	1.4		
1RHD	2.5		
1REI	2		
1MLT	2		
1MBS	2.5		
1MBD	1.4		
1LZT	1.9		
1LZM	2.4		
1LZ1	1.5		
8CAT	2.5		
5RXN	1.2		
5PTI	1		
5CPA	1.5		
4FXN	1.8		
4DFR	1.7		
451C	1.6		
3TLN	1.6		

3SGB	1.8	
3PCY	1.9	
3HHB	1.7	
3FXC	2.5	
3CNA	2.4	
3C2C	1.6	
2STV	2.5	
2SOD	2	
2SNS	1.5	
2SGA	1.5	
2PAB	1.8	
2MT2	2.3	

TABLE 7.2

Brookhaven codes with resolutions

is unable to form helix due to global constraints (see Discussion and Conclusions for further discussion).

7.10.2 Sequence Comparison with Pseudo Sequences

In order to investigate pseudo sequence more closely comparisons were made between the pseudo sequences and the successfully learnt sequences for window size 10. This was done to find real helix sequences that closely matched pseudo-helix sequence and real non-helix sequences that closely matched pseudo-non-helix sequence. Using a 100 PAM table (Dayhoff *et al.*, 1978) (see table 7.3) all the successfully learnt sequences were searched for the best match to each of the pseudo sequences. Roughly 46% of pseudo-non-helix sequences (helix sequence predicted non-helix) had their best matches with non-helix sequences and 61% of pseudo-helix sequences (non-helix sequence predicted helix) had their best matches with helix sequences. These comparisons with their scores are shown in Appendix C. Table 7.4 shows a selection of high scoring sequence pairs from

Appendix C. One also gets broadly the same results using a 50 PAM table. Looking at the sequences in table 7.4 or Appendix C, one may think one sees the reason for one sequence being a coil when its partner is helix. For example, the sequence KWERPFEVKD has a proline in position 5, whereas its partner in the helix conformation has not. However, there are plenty of examples of helices containing prolines, as first sequence in table 7.4 shows. Looking at the weights in figure 7.5 one can see that many of the substitutions between the sequence pairs are between pro-helix residues and are not therefore expected to disrupt the structure from helix to coil or beta-sheet. For examples the substitutions between the helix PEELKGIFEK and the coil LDDLKGAFAQ do not seem sufficient to cause this change in structure. Indeed, the helix sequence with its proline seems to be the more likely coil candidate. The pseudo-non-helix sequences are even more extreme examples of unlikely helix sequences and have consequently been predicted as non-helices by the network. These contain significantly more anti-helix residues than the pseudo-helix sequences, but unlike the pseudo-helix sequences they are in the helix conformation. For example, the sequence YPIYYPLLNA is a helix yet the sequence KANAKDIKLV is a coil (see Discussions and Conclusions pages 131 and 132 for further discussion).

	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Z
A	6	-1	-5	-1	0	-7	1	-5	-3	-4	-5	-3	-1	1	-2	-5	2	2	0	-11	-6	0
B	-1	8	-8	8	5	-6	-1	2	-4	1	-6	-5	7	-3	1	-3	2	0	-5	-8	-3	5
C	-5	-8	14	-11	-11	-10	-8	-6	-5	-11	-12	-11	-8	-6	-11	-6	-1	-5	-4	-13	-2	-11
D	-1	8	-11	8	5	-11	-1	-1	-6	-2	-9	-8	4	-4	1	-6	-1	-2	-6	-13	-9	5
E	0	5	-11	5	8	-11	-2	-2	-5	-2	-7	-6	1	-3	4	-5	-2	-3	-5	-14	-7	8
F	-7	-6	-10	-11	-11	12	-8	-4	0	-11	0	-2	-6	-9	-10	-7	-5	-6	-5	-2	6	-10
G	1	-1	-8	-1	-2	-8	8	-7	-7	-5	-8	-8	-1	-3	-5	-8	1	-3	-4	-13	-11	-2
H	-5	2	-6	-1	-2	-4	-7	11	-7	-3	-5	-7	2	-2	4	1	-4	-5	-6	-7	-1	4
I	-3	-4	-5	-6	-5	0	-7	-7	9	-4	2	2	-4	-6	-5	-4	-4	-1	5	-12	-4	-5
K	-4	1	-11	-2	-2	-11	-5	-3	-4	8	-6	1	1	-4	-1	3	-2	-1	-6	-9	-10	-1
L	-5	-6	-12	-9	-7	0	-8	-5	2	-6	9	4	-6	-5	-3	-7	-7	-5	1	-7	-5	-3
M	-3	-5	-11	-8	-6	-2	-8	-7	2	1	4	13	-5	-6	-2	-2	-4	-2	1	-11	-8	-2
N	-1	7	-8	4	1	-6	-1	2	-4	1	-6	-5	7	-3	-1	-3	2	0	-5	-8	-3	1
P	1	-3	-6	-4	-3	-9	-3	-2	-6	-4	-5	-6	-3	10	-1	-2	1	-1	-4	-11	-11	-1
Q	-2	1	-11	1	4	-10	-5	4	-5	-1	-3	-2	-1	-1	9	1	-3	-3	-5	-11	-9	9
R	-5	-3	-6	-6	-5	-7	-8	1	-4	3	-7	-2	-3	-2	1	10	-1	-4	-6	1	-10	1
S	2	2	-1	-1	-2	-5	1	-4	-4	-2	-7	-4	2	1	-3	-1	6	2	-4	-4	-6	-2
T	2	0	-5	-2	-3	-6	-3	-5	-1	-1	-5	-2	0	-1	-3	-4	2	7	-1	-10	-6	-3
V	0	-5	-4	-6	-5	-5	-4	-6	5	-6	1	1	-5	-4	-5	-6	-4	-1	8	-14	-6	-5
W	-11	-8	-13	-13	-14	-2	-13	-7	-12	-9	-7	-11	-8	-11	-11	1	-4	-10	-14	19	-2	-11
Y	-6	-3	-2	-9	-7	6	-11	-1	-4	-10	-5	-8	-3	-11	-9	-10	-6	-6	-6	-2	13	-7
Z	0	5	-11	5	8	-10	-2	4	-5	-1	-3	-2	1	-1	9	1	-2	-3	-5	-11	-7	8

TABLE 7.3

100 PAM table.

PSEUDO-NON-HELIX SEQUENCES (H) MATCHED WITH NON-HELIX SEQUENCES (C,B)	PSEUDO-HELIX SEQUENCES (C,B) MATCHED WITH HELIX SEQUENCES (H)

H FAYPDTHRHR
  | |
34 C YSYTDANKSK
	B VSFEATFAFL
	  || || |
	43 H VSIATAFAML

H FSQVCTHLDT
  | | |||
32 C FTQGLKHLDD
	B PDVLKALKAP
	  || ||| |
	41 H PDALKAQAAA

H TGSMDALKAA
  || || |
39 B SGSVTALNAT
	B AHGQAVQAAQ
	  |||| | |
	38 H AHGQKVANAL

H TTGSMDALKA
  || || |
32 B HSGSVTALNA
	C LTRTNGQLAQ
	  | | |||
	36 H VSRALGVLAQ

H CIDCHALKKK
  | || |
41 C CAQCHTVDKG
	C LDDLKGAFAQ
	  ||| |
	38 H PEELKGIFEK

H YNMINTVKSD
  ||||
34 C FSSINTVQGS
	C KEELEKKGLG
	  | || ||
	34 H AEALERMFLG

H TRTRLSFQTS
  | ||| |
33 C QRMFLSFPTT
	C DEAAVNLAKS
	  | | |||
	33 H DDEAQTLAKW

H RTRLSFQTSM
  | ||| |
37 C RMFLSFPTTK
	C NDIEDVEKYF
	  || | |
	40 H DEIENVIAYL

H GATLDTFFGM
  | | ||
37 C GPNLNGLFGR
	C KWERPFEVKD
	  || | ||
	37 H KWMRDFEERM

H GQQEAARAGE
  | || | |
42 C GHQENAKNEE
	C TKGKLRFVRN
	  ||| ||
	38 H AKGKKTFVQK

H SRLNAIYQNN
  | | ||
32 C ETLNPIIQNT
	B TEAAGAMFLE
	  || |||
	36 H AEALERMFLS

H KSIVDFVKNH
  | | | |
40 B ASVVBFLNNF
	B SKAVHKAVLT
	  | ||| |
	34 H WKEVHKMVVE

TABLE 7.4 (see Appendix C for more examples)

7.10.3 Distribution of Pseudo-Helix Sequences

Looking at these pictures and the general distribution of the regions of pseudo-helix sequence gives one the impression that they occur quite often at the C- and N-termini of helices and that they are potentially longer helices that have been disrupted. The distribution was tested statistically. The probability p of a window of pseudo-helix sequence was simply calculated by dividing the number of windows of pseudo-helix sequence by the total number of non-helix windows. If n is the number of windows of non-helix in the first position adjacent to the C-terminus of a helix, then the population average μ will be np, and the population variance σ^2 will be np(1-p), as the number of windows of pseudo-helix sequence is expected to be distributed binomially. As n is in the order of hundreds it will be safe to approximate this binomial distribution by a normal distribution with the same population average and variance. If there are x actual windows of pseudo-helix sequence in the first position adjacent of the C-termini of helices, then one can test its statistical significance by transforming to the standard normal variate z using the transformation:

$$z=\frac{x-\mu}{\sigma}. \qquad (7.1)$$

One can look up the z value in the standard normal distribution table to find the probability of x or more occurrences happening by chance. The null hypothesis is that the beginnings of windows pseudo-helix sequence are equally likely to occur at the first position adjacent to the C-termini of helices as anywhere else, and the alternative hypothesis is that they are more likely to occur near the C-termini. For the first position the value of z is 1.8, which gives a probability of 3.6%, which is well within the often chosen 5% significance level for accepting the alternative hypothesis. Table 7.5 shows these probabilities for the beginnings of windows

from 0 (the first position) to 9 residues from the C-terminus of a helix.

0	1	2	3	4	5	6	7	8	9
3.6%	3.8%	0.7%	0.6%	10%	14%	100%	10%	82%	99%

TABLE 7.5

At the N-terminus of a helix, the alternative hypothesis that the ends of windows of pseudo-helix sequence are more likely to occur near the N-termini of helices. Table 7.6 shows the probabilities analogous to those in table 7.5 for the ends of windows from 0 to 9 residues from the N-terminus of a helix.

0	1	2	3	4	5	6	7	8	9
8.8%	2%	10.7%	17%	9%	6.5%	27%	59%	17.5%	35%

TABLE 7.6

As one can see from table 7.5 the alternative hypothesis is upheld at the 5% significance level up to the fourth residue from the C-terminal of helix. However, the alternative hypothesis is only roughly upheld at the 10% significance level up to the sixth residue from the N-terminal (table 7.6). It was also tested whether windows of pseudo-helix sequence are more likely to occur at C-termini of helices than at N-termini. The average number of windows, x_n, of pseudo-helix sequence beginning within four residues of the N terminus of a helix was calculated as was the standard deviation σ_n. Similarly the average number of windows, x_c, of pseudo-helix sequence ending within four residues of the C-terminus of a helix was calculated as was its standard deviation σ_c. In this case the null hypothesis is that the population means are the same. The alternative hypothesis is that they are different; more specifically, windows of pseudo-helix sequence are more likely to occur at the C-termini rather than the N-termini of helices. In this case our z value is given by:

$$z = \frac{x_c - x_n - 0}{\sqrt{\frac{\sigma_c^2}{N_c} + \frac{\sigma_n^2}{N_n}}}, \qquad (7.2)$$

where N_n(=92) is the number of examples of ends of windows of pseudo-helix sequence within four residues of the N-terminus of a helix and N_c(=85) is the number of beginnings of windows of pseudo-helix sequence within four residues of the C-terminus of a helix. The z value calculated using equation (7.2) is 0.6, giving 30%, a value that is well outside any acceptable significance level and so the alternative hypothesis cannot be accepted.

7.10.4 Comparison with Trichotomous Classification Methods

Most secondary structure methods are not restricted to simply predicting the presence of helix, but attempt to predict all three structures: helix, beta-sheet and coil. In neural network terms, the dichotomous classification is achieved by the network establishing a single boundary between helix sequences and non-helix sequences. In the trichotomous case a further boundary between coil and beta-sheet has to be established. Unless the establishment of this further boundary affects the original boundary there is no reason for helix prediction alone to be better or worse than for the dichotomous case. However, this problem is not well bounded and so the establishment of the boundary between coil and beta-sheet may upset prediction of helix. Unfortunately all the neural network papers published so far have omitted to state the percentages of residues in the individual structures that they predict correctly. This information is needed to know if one wants to make a direct comparison between the trichotomous and dichotomous prediction methods. In these papers the overall percentage of correct predictions

is stated along with the individual correlation coefficients for the three structures. The question that one naturally asks at this stage is whether the individual percentages are uniquely determined by the overall percentage of correct predictions P and the individual correlation coefficients: Q_h, Q_b, and Q_c (see section 2.4, equation (2.6) for definition of correlation coefficient). Taking helix as an example one can put this question more succinctly: is w, the number of correctly predicted helix residues, determined by Q_h and P? Given p=P/100, then one has the following equations relating x,y and z to w (see section 2.4 for definitions of x, y and z):

$$x=Np-w, \tag{7.3}$$

$$y=N_h-w, \tag{7.4}$$

$$z=N_n-x, \tag{7.5}$$

which gives substituting for x:

$$z=N_n-Np+w, \tag{7.6}$$

where N is the total number of residues, N_h is the number of helix residues and N_n is the number of non-helix residues. Substituting these expressions into equation (2.6) in section 2.5 and rearranging the many terms one finally arrives at the following quadratic equation for w:

$$\begin{aligned}&[(N_n-N_h)^2+4Q_h^2N_nN_h]w^2\\&+[2(N_n-N_h)(N_hNp-N_nN_h)-Q_h^2N_nN_h(4Np-2N_n+2N_h)]w\\&+[(N_hNp-N_nN_h)^2-Q_h^2N_nN_h(N_nNp+N_nN_h-N_hNp-N^2p^2)]=0\end{aligned} \tag{7.7}$$

This means that, given the overall percentage of correct predictions, the correlation coeffcients, and the numbers of residues in the individual secondary structures, w is specified to two values.

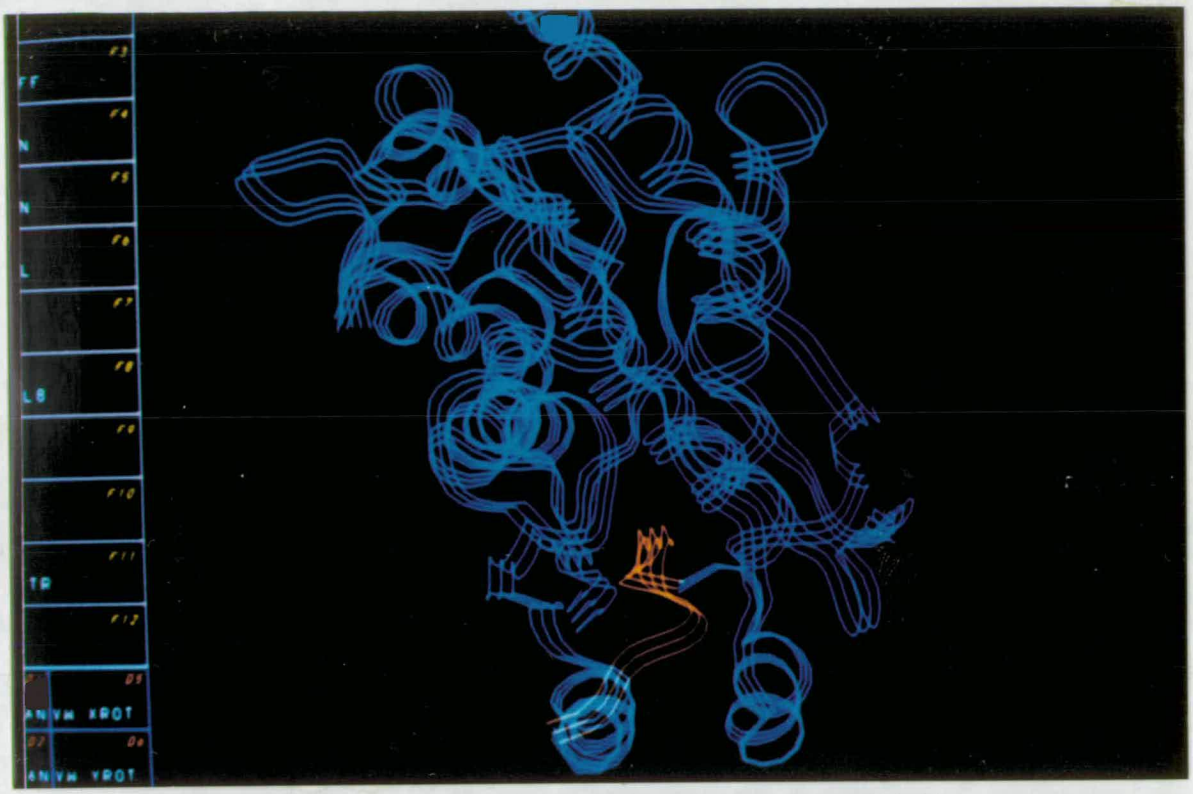
In the paper by Qian and Sejnowski they quote an overall prediction success of 64.3% with correlation coefficients Q_h=0.41, Q_b=0.31, and Q_c=0.41. The values for N, N_h, N_b and N_c are 3520, 848, 750 and 1925 respectively, where N_b is the number of beta-sheet residues and N_c the number of coil residues. Using equation (7.6) one arrives at the result that 92% of helix, 77% of beta-sheet and 37% of coil were predicted correctly. The other roots to equation (7.6) are unacceptable. If one checks whether these percentages give the correct overall percentage of 64.3% one is disappointed to find that it actually gives a value of 59%. This is due to the fact that the correlation coefficients are averages of the correlation coefficients from predictions of the individual test proteins. Nevertheless these percentages will be roughly correct. In support of this, one gets broadly the same result using the results from the Holley and Karplus paper: 93% for helix, 82% for beta-sheet and 35% for coil. What is striking about this result is that prediction is incredibly imbalanced, with prediction overwhelmingly favouring helix. Using these results one gets a correlation coefficient of 0.35 for helix prediction alone. This is very comparable to our result of 0.34, but the values of 59% is well below our value of 73% for the percentage of correct predictions. This can simply be explained by the fact that our prediction on individual residues overpredicted the more abundant non-helix residues and their method overpredicted the less abundant helix residues. Which of these two results is the best is open to debate, but taking the correlation coefficient to be the true measure of prediction success one can conclude that dichotomous and trichotomous prediction do not differ significantly for helix. The overprediction of helix and beta-sheet to the detriment of coil prediction gives results that are more comparable to those from the network with the shifted boundary than the network with the original boundary. It is somewhat surprising that the overprediction was not in favour of coil as the training sets used contained an

overabundance of coil. However, this could possibly be explained by the fact that helix is easier than coil to learn, as a larger percentage of coil sequence is pseudo sequence than helix.

It would have been instructive to have had the exact results from the aforementioned papers to find out whether they also predicted the same regions incorrectly, but they were unavailable. Instead, the GOR prediction method, which is available in the Wisconsin Package, was used. Predictions from GOR were compared to the predictions here on residues that were non-helix, but predicted helix. These include pseudo-helix sequence but due to averaging over the sliding window also include short regions of non-helix between helices that is not pseudo-helix sequence in the sense discussed above. Using the test set only, 56% of non-helix predicted helix are also predicted helix by GOR. In all 81% are predicted either helix or beta-sheet by GOR. For coil alone 59% of coil predicted helix by the network are also predicted helix by GOR. Again this figure rises to 77% for GOR predicting helix or beta-sheet.

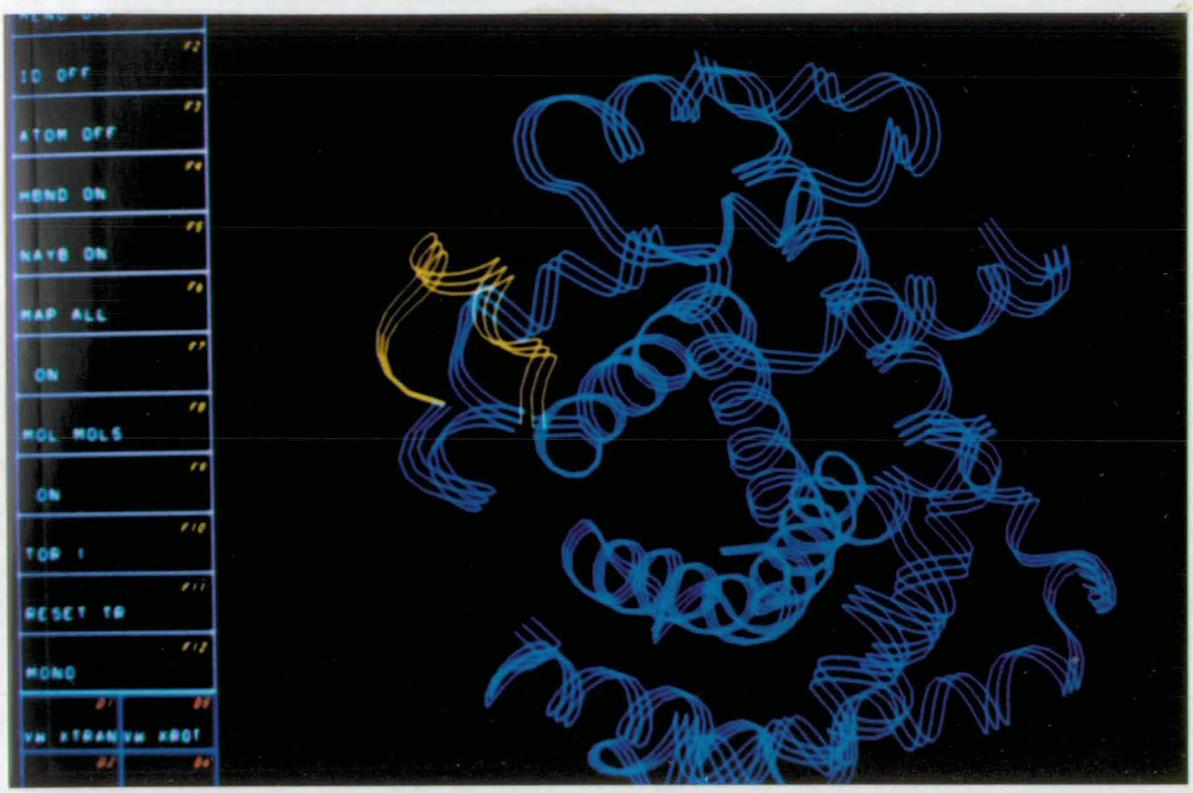
Figure 7.13

Photographs of computer generated ribbons following the backbone of 8 proteins from the training and test sets. Highlighted are regions of coil pseudo-helix sequence, where consecutive prediction of helix occurs as the window (size 10) slides along the sequence. The window does not span any actual helix regions and strong helix prediction occurs towards the centres of the highlighted regions.

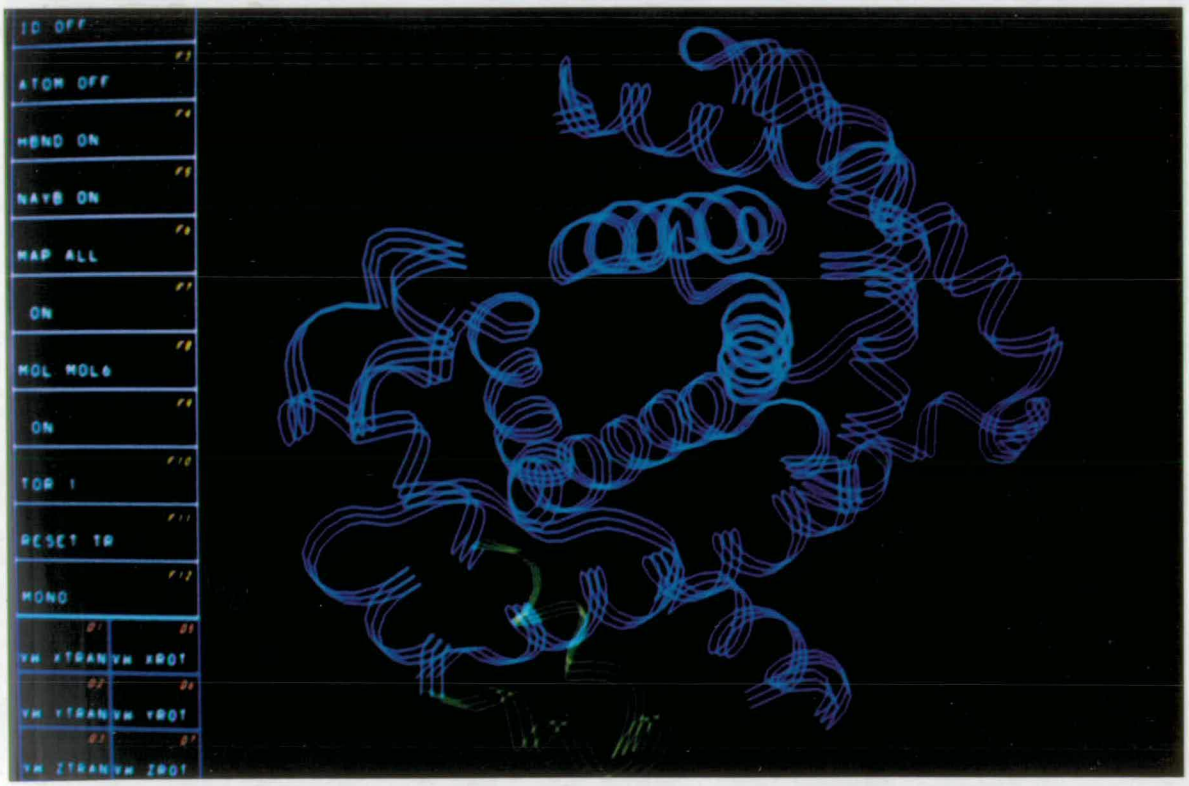


1ABP, L-arabinose-binding protein, residues 56-64
SGAKGFVICT

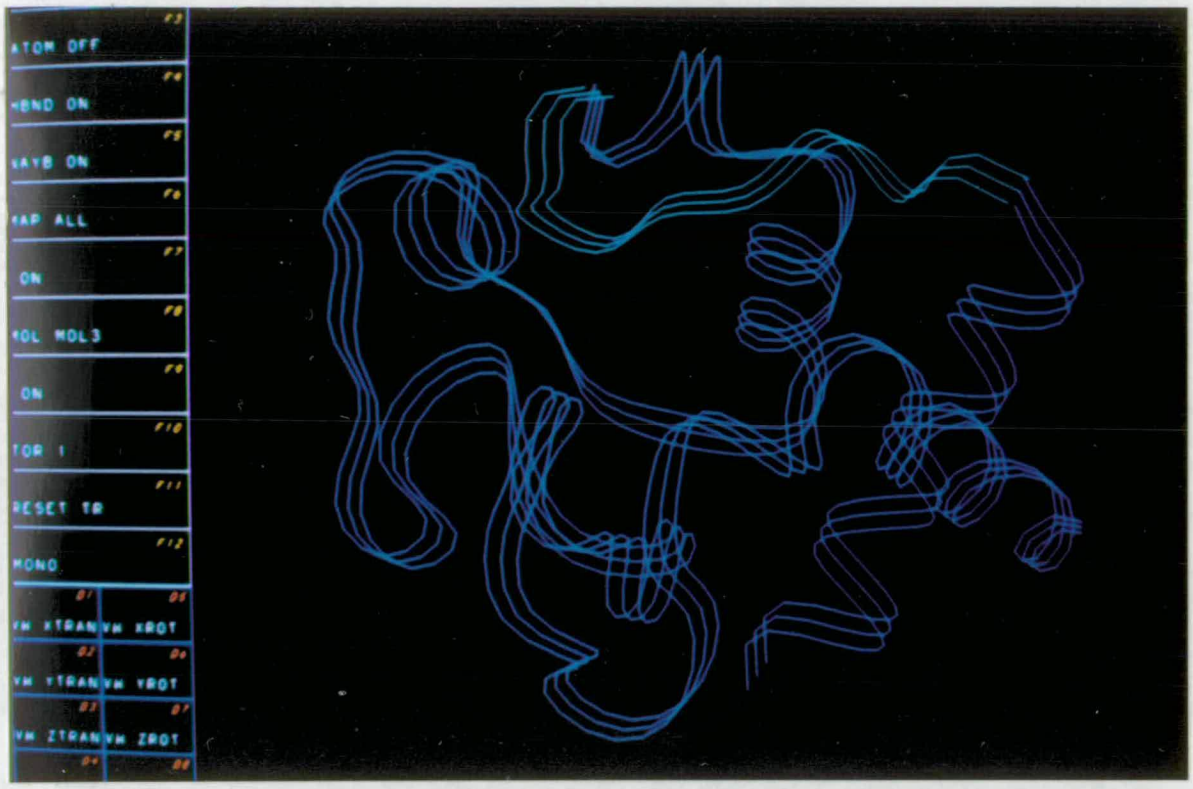
1ABP, L-arabinose-binding protein, residues 82-98
DMKVIAVDDQFVNAKGK



1HDS, Haemoglobin (sickle cell), residues 37B-47B
TQRFFQHFGN

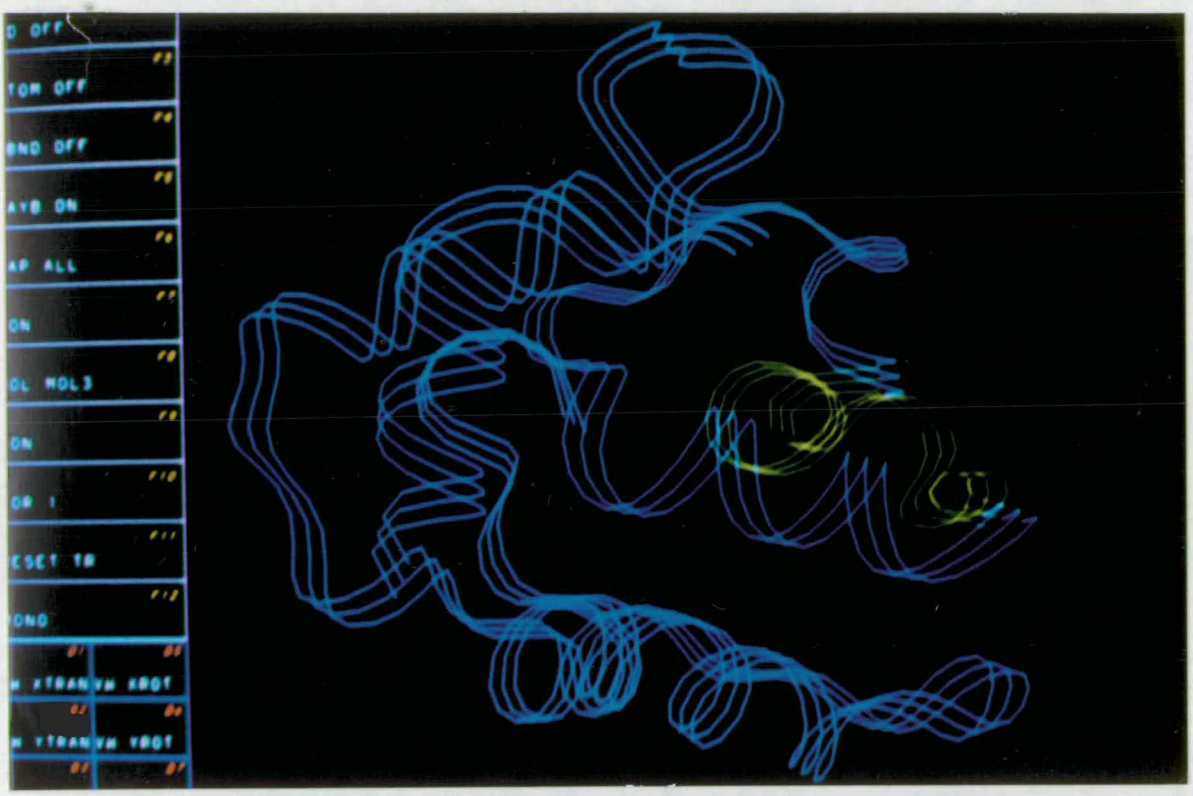


1HDS, Haemoglobin (sickle cell), residues 66B-86B
VLDAFTQGLKHLDDLKGNFAQ

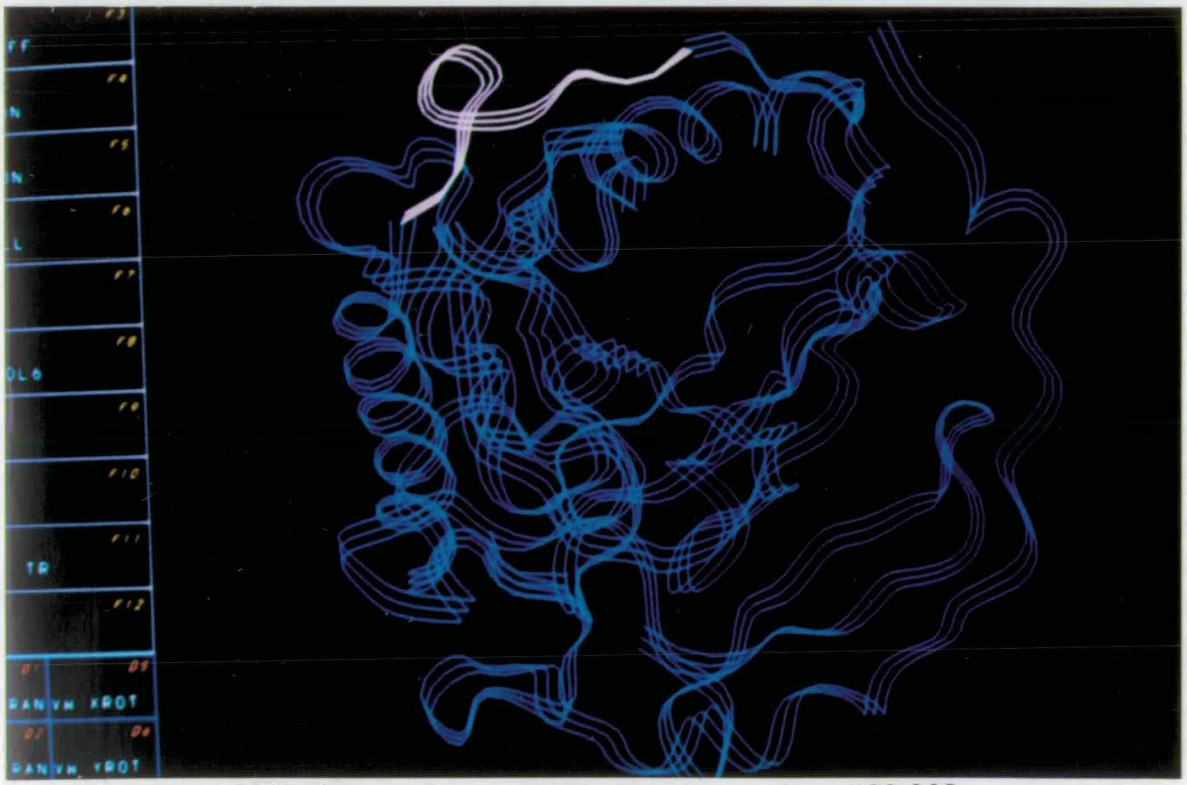


1CYC, Ferrocytochrome c, residues 76-87
PGTKMIFAGIKK

1CY3, Cytochrome c3, residues 69-87
SLEFRDKANAKDIKLVES



1CPV, Calcium binding protein, residues 65-78
LFLQNFKADARALT



1PYP, Inorganic pyrophosphatase, residues 199-209
ENQFAFSGEAK

See Appendix D for a table of all documented runs in this work.

DISCUSSION AND CONCLUSIONS

This work began with the Hopfield model. Initially, results with this model gave some hope of more promising results if the training set were larger. Later however, a difficult choice had to be made whether to continue with this approach and parallelize the program to cope with more data, or to abandon it for a layered network. As a layered network approach seemed to be more suited to this problem the Hopfield model was abandoned. In hindsight it can be confidently stated that the decision to abandon the Hopfield model was the correct one. It would have been impossible to deduce all that has been deduced with the layered networks with a Hopfield network. It is possible that the Hopfield network would have eventually performed as well as the layered networks in terms of prediction success because in essence it should be as capable as a single layered network. But it would have been impossible for us to have directly deduced as much about the input distribution as we have been able to do with layered networks with hidden nodes. In addition to these limitations the Hopfield network has a simple practical disadvantage compared to layered networks in that it has no output node to indicate the class to which the input pattern belongs. In fact one cannot be too confident that the Hopfield network would have performed as well as the layered network as it receives less information about the data than a layered network. During training the layered network receives not only the input pattern but the target output to indicate to which class the input pattern belong. The Hopfield network, however, only receives the input pattern and must find the correct grouping without any help.

Initially work with the layered networks lacked clear direction. At this time there existed no publications in this field and little was known about layered networks trained by the back propagation algorithm. It was natural, therefore, to abandon one approach for another as it was not achieving the desired 100% prediction success. This initial work was often unsuccessful for a variety of reasons, but one underlying thread for failure was that too much information was demanded at the output when too little was being presented at the input. Early work using the pair coding scheme was an example of this.

Initially the pair coding scheme seemed like a good idea for the reason that the translation of the window would be reflected by a proportionate change in the input code, whereas with positional coding scheme a large change would occur even when the window translates only one amino acid along the sequence. More importantly though, the paper by Gibrat *et al.* suggests that pairs are important for secondary structure formation, and so prediction. So what are the reasons for networks using this coding scheme performing significantly worse than networks with the positional coding scheme? Firstly, performance was probably somewhat impaired by the information content of the pair coding scheme used being less than with the positional coding scheme. Even when the 400 nearest neighbour pairs are coded for, this does not necessarily contain all the information needed to specify the sequence. Perhaps more essentially, however, is the fact that if the positional coding scheme is used the problem is basically linearly separable. This means that the positional coding scheme is the natural coding scheme for this problem and not the pair coding scheme. If one maps the input space for the positional coding scheme to the input space for the pair coding scheme it will distort the simple decision plane to some other more complicated surface. This is supported by the fact that a single layer network with the pair coding scheme does significantly worse than those with hidden nodes. In essence, the problem of how to deduce whether a sequence is helix or not is a very simple one. One needs only to look at each of the amino acids in their positions as independent entities to make one's decision. This is what the single layer network with the

positional coding scheme is doing. The pair coding scheme, however, unnecessarily complicates the matter, and a network with hidden nodes is needed to bound the resulting decision surface. In other words, the hidden nodes are being used to map the problem back to the natural code for this problem: the positional code.

Our attempt to predict helices by predicting the presence of helix boundaries was motivated by the belief that boundaries may be inherently more predictable than the structure to which an individual residue belongs. The idea behind this is that there may exist an inertia that encourages additional residues to continue the previously formed structure. This would mean that many residue combinations would occur within a given secondary structure, but much stronger features would be needed to initiate or terminate that structure (Richardson and Richardson, 1988). However, even if this is true this method has a serious drawback. If a mistake is made in predicting a boundary it will be more serious for the overall secondary structure prediction of a protein than if a single residue is falsely predicted. For that reason boundary prediction needs to be particularly reliable. The attempt to predict boundaries, although unsuccessful, highlighted another important point. If the nature of the problem is such that the occurrence of one feature far outweighs that of another (in this case non-boundaries to boundaries), how does one construct the training set? If one includes all non-boundaries then the training set will be so unbalanced that the network will be impossible to train. If the training set is balanced by choosing only as many non-boundary examples as boundary examples, then there may be too few non-boundary examples for the network to be able to establish the difference between a boundary and a non-boundary. To circumvent this problem a method was devised that did not require any non-boundary examples. This method relied on training the network with the positional coding scheme to output a signature sequence as the window translated a boundary. It was hoped that although the network would always output one of the outputs belonging to the signature sequence it would only output that sequence at a boundary. In fact this method was largely unsuccessful. Whether

this was due to a basic flaw in the method or due to a lack of training examples remains an unanswered question.

At this point three important lessons had been learnt. Firstly, one needs to select an approach that requires as little information as possible from the output and allows more information to be presented at the input. Secondly, an approach that does not, unlike the boundary problem, have examples of one feature far outweighing those of the others is probably essential. And thirdly, the positional coding scheme is the natural coding scheme for the prediction of sequence segments with a specific secondary structure. For these three reasons, helix prediction with such segments using the positional coding scheme forms the main aspect of this work. The first problem to be tackled, concerned the problem of balancing the training set. Although the number of non-helix examples outweighs the number of helix examples, it was still possible to train a network with the unbalanced training set. It was found that the non-helix sequences in the test set were over predicted, but the helix examples were under predicted in relation to a balanced training set. In terms of the correlation coefficient, the network trained with the balanced training set does far better than the network trained with the unbalanced training set, despite the fact that the unbalanced training set contains more information. This is something that many researchers in this field have not noticed. A simple method of including the information left out to achieve a balance was devised. This, however, did not improve the situation significantly. This is probably due to the nature of the problem. From figure 7.4 one can see that for a training set of this size (the maximum size in the figure) an increase in the size of the training set will not produce a large increase in prediction success. So although this method was not of great benefit here, it may be useful to problems that have a steeper learning curve.

The most significant result of the previous section comes from the experiment to distinguish correctly recognised helix sequence from pseudo-helix sequence. It failed, giving a correlation coefficient of 0. One is tempted to conclude that the

points representing the pseudo-helix sequences are randomly mixed amongst the space occupied by the points representing the successful helix sequences. Why is this a natural conclusion to reach? Consider the situation where the points in the input space from the two classes, pseudo-helix and successful helix, are at least partially separated. In training the network should be able to form a boundary between the two regions from examples in the training set. The more examples the training set contains, the more accurate this boundary will be to the true boundary that separates the whole two regions. As the test set examples are also from these two regions one would expect, given the training set contains sufficient examples for an approximation to the true boundary to be established, a positive correlation coefficient on testing. In the case where the points from the two classes are randomly mixed, after training to 100%, each point in the training will be surrounded by a region (the islands in figure 7.8) within which all points will be recognised as belonging to the same class as that point. In our case, where a balanced training set was used, there was an equal number of points representing each of the two classes and consequently one would expect there to be regions of equal area representing the two classes. A test point from the same random set is therefore as likely to fall within a region that is not thought to belong to its class as to fall within a region that does. From this we can conclude that our null result arises either because there were insufficient examples in the training set (this is discussed below), or because the points from the two classes were randomly mixed.

In the case of the pseudo-non-helix sequences a positive correlation coefficient was found in an analogous experiment, suggesting that successful non-helix sequences and pseudo-non-helix sequences are distinguishable. One can, however, select a group of non-helix sequences (those with higher outputs) for which the correlation coefficient is reduced. One can also select a group (those with lower outputs) for which the correlation coefficient remains about the same. Again, one is tempted to conclude that the pseudo-non-helix sequences are randomly distributed only in a certain region of the space of successful non-helix sequences.

Figure 7.1 can now be explained if the input space for helices and non-helices is as depicted in figure 7.8 and by the way the network bounds the helix and non-helix regions during training. First, the network establishes a simple decision plane represented by the continuous line in figure 7.8 during the fast learning phase. This must be the case as the successfully learnt sequences can be learnt to 100% by a single layer network. After the decision plane has been established, pseudo-helix sequence, which is non-helix sequence, and pseudo-non-helix sequence, which is helix sequence, are learnt during the slow learning phase. Bounding the regions of pseudo-helix sequence causes islands of non-helix prediction in the majority helix region. These islands of non-helix prediction are *effectively* randomly distributed and are therefore more likely to coincide with actual helix sequence than pseudo-helix sequence in the test set, contributing to a decrease in helix prediction success. The islands of helix prediction established in the majority non-helix region will on the whole not coincide with the regions of pseudo-non-helix sequence in the test set and will effect little or no contribution to helix prediction success. Thus overall there will be a decrease in helix prediction during the slow learning phase. An analogous argument can be given for non-helix prediction. Here, there are fewer islands of helix prediction established in the majority non-helix region and the decrease in non-helix prediction success is consequently less pronounced (see figure 7.2). Figure 7.3 can be simply explained by the fact that the single layer network is unable to learn the pseudo sequences and consequently the error remains at the value at which the networks with hidden nodes switch from rapid to slow learning. As a further consequence the prediction success does not significantly decrease with further training.

Figure 7.4 also has a simple explanation. The networks with hidden nodes do not improve upon the single layer network as they also establish the same decision plane as the single layer network. This gives maximum prediction success for the reason explained above. The form of this curve is also what one would expect for the establishment of a decision plane. With very little data the chances of the

plane being the correct one are very small and therefore prediction success is small. As the amount of data increases the position of the plane becomes less and less ambiguous until even large amounts of extra data do not effect a large change in its position. Thus prediction success plateaus. The fact that the correlation coefficient is well below 1 is largely then due to pseudo sequence in the test set.

Although figure 7.1 is explained by the above reasoning it could also be explained if the training set and test set examples occupy different regions of the total set's space. After the simple decision plane is established, learning further examples in the training set could move the boundary away from the test set resulting in a decrease in prediction success. It is, however, hard to see how all the results of the previous section can be explained by such a mismatch between the training and test sets. The inability of a network to distinguish pseudo-helix sequence from successful helix sequence would require a further unlikely difference between the training and test sets. As already mentioned, our null result could possibly be explained by the network being trained on an insufficient number of examples for it to be able to find the true, non-trivial distribution. In fact, according to Baum and Haussler (Baum and Haussler, 1989), a fully connected network of with W weights and one hidden layer, trained to 100%, on fewer than W/ϵ training patterns, where $0<\epsilon\leq 1/8$, will fail on a finite number of occasions to correctly classify more than a $1-\epsilon$ fraction of test patterns. If a prediction success of 90% is required, then roughly 10 times as many training patterns as weights will be needed. The network used here had 1005 weights and was trained on only 500 patterns, so, in order to determine whether pseudo-helix sequences are actually randomly distributed amongst the majority of helix sequences, or a non-trivial distribution exists, a far greater number of training examples will be needed. So at the present moment in time we can say only that the distribution, to the network, is effectively random. Although 10,000 patterns (which is 20 times the present figure of 500), giving an expected prediction success of 90% may be unnecessarily many to deduce whether the distribution is non-trivial or random,

it is clear that given the rate at which new structures appear in the Brookhaven data bank, this approach will not be able to address this problem properly in the foreseeable future. Here we have glossed over an important point. Is it really feasible that the two sets are even approximately randomly mixed? Consecutive windows are often predicted as pseudo-sequence. So, when the window translates in the left to right direction one amino acid along the sequence, the 1's in the codes for each of the amino acids within the window, except for the first one will shift one position to the left. So there will indeed be some underlying structure to the distribution. However, that structure may be too complicated and non-deterministic for any pattern recognition method to be of any use. If, for example, the distribution of the pseudo-helix sequences amongst the successful helix sequences has a very branched or fractal like structure, it will probably be too complicated for any pattern recognition method to learn to generalise it. In other words, the two sets do not need to be randomly mixed for pattern recognition methods to fail in prediction. From a physical point of view, our result can be plausibly explained if pseudo-helix sequences are potential helix sequences that have not been able to realise that potential due to global constraints.

Are there other methods we could use to try to understand this distribution? In fact there existed before the emergence of neural network techniques a number of methods for analysing multivariate data. Many problems that have been analysed with these techniques have also been analysed with neural network models and often with better results. One of these methods is Principle Component Analysis, PCA (Wold, *et al.*, 1983). This is a method by which points in a high dimensional space can by analysed by means of pictures and statistical parameters in a space of lower dimension (2 or 3), whose axes are determined by the natural form of the point distribution. If there is a lower dimensional grouping then this method should find it. With this method it may be possible to reduce the dimensions of the space from 200 to 2 and then analyse the resulting distribution of points in this 2-dimensional space for any emerging pattern. If we

expect the distribution for the two point sets to be a random mix, then the resultant 2-dimensional space could be divided up into squares and the expected normal distribution for the difference in the number of points from the two sets could be calculated and compared to the actual difference using the Chi-Squared test. However, this method will require a large number of points. In terms of pattern recognition, the network would have also found any low dimensional division of the points. It is likely, therefore, that the only real use of this method will be to give some pictorial impression of the distribution. Another method that may be of some use in the analysis of the distribution is the K Nearest Neighbour method (Wold, *et al.*, 1983). This method classifies a new object in the class of the majority of its K nearest neighbours, where K is usually chosen to be 1 or 3. If the two sets are randomly mixed one would not expect to find any correlation between the prediction as to which class the point belongs and its actual class. Will this method be able to improve on the neural network approach? As already mentioned the neural network is able to successfully bound every point in the training set. Any test set point that falls within one of these bounded regions will be predicted to belong to the same class as the training set points successfully bounded in that region. These training set points are also extremely likely to be the nearest neighbours to the test set point and one can conclude therefore that the K Nearest Neighbour method is unlikely to improve on the neural network approach. These methods are usually worse at prediction than neural networks and in the end, the real point is that pseudo-helix sequences are indistinguishable from the majority of helix sequences using the best pattern recognition method available and given the size of the present structural data bank. There exists, therefore, at the present time at least, a limit on the success of secondary structure prediction using this approach, and probably any other approach.

As already mentioned a plausible explanation for pseudo-helix sequence is that it is potential helix sequence that has been unable to realise that potential due to global constraints during protein folding. This does seem more plausible than the idea that the pseudo-helix sequence is subtly different from real helix sequence.

The same argument can be put for pseudo-non-helix sequence. However, as we can only deduce that the distribution is effectively random one cannot rule out the possibility that, for example, some sequences or specific combinations of residues always form a helix. However, from these results, it does seem unlikely that all sequences found as helix in known protein structures will necessarily be found as helix in new structures. In section 7.10.2 we compared the pseudo-helix sequences with some real examples of helix sequences and pseudo-non-helix sequences with real examples of non-helix sequences. One can see from their residue composition why these sequences were falsely predicted. If one visually compares a typical example of a pseudo-helix sequence with a typical example of a pseudo-non-helix sequence, one would naturally say that the pseudo-helix sequence should be a helix with so many pro-helix residues, and the pseudo-non-helix sequence should not be in the helix conformation with so many anti-helix residues, including glycines or prolines. So in some sense the network is matching our intuitive ideas about what sequences should form helices or non-helices. To verify these ideas, one could look for helices of length 10 or longer whose sequences were identical to those in other structures. In 1984 Kabsch and Sander (Kabsch and Sander, 1984) published a paper on their discovery of 6 pentapeptides that occurred in both the helix and beta-sheet structures in unrelated proteins. In the database they used, which contained 62 proteins, they calculated that it was too small for them to expect to see any hexapeptides. The database is likely to remain too small for us to expect to see identical sequences of 10 or more peptides in different structures.

The fact that this problem is basically solvable by a single layer network means the weights are more easily interpretable than in the case of a network with hidden nodes, where the meaning of the weights is often obscure. If a weight is negative, the residue concerned does not favour helix at that position; if it is positive it favours helix at that position. Proline and glycine are, as expected, strongly anti-helix. Serine is also somewhat anti-helix. With the exception of histidine, all residues are roughly either pro-helix or anti-helix in virtually all

window positions. It should be pointed out that the beginnings and ends of the windows do not correspond directly to the beginnings and ends of helices and non-helices. However, the beginning of the window does correspond to the window position that is nearest the N-terminus of the helix when the window is within a helix or the N-terminus of a non-helix when the window is within a non-helix region. Similarly, the end of a window corresponds to the window position that is nearest the C-terminus of the helix when the window is within a helix and the C-terminus of a non-helix, when the window is in a non-helix region. If a residue, such as histidine, has weights that are anti-helix or more explicitly pro-non-helix at the beginning of its window this is due to its preference for a position in a non-helix region directly adjacent to the C-terminus of a helix. If, again like histidine, a residue has weights that are pro-helix at the end of its window, this is due to its preference for a position near the C-terminus of a helix, but within the helix this time. So one can conclude that histidine likes to be near, and on either side of, the C-terminus of a helix. Consider the case of a residue that favours the N-terminus of a helix in the same way that a histidine favours the C-terminus. The plot for this residue will slope in the other direction to that for histidine. Now consider a residue that simply favours the termini of helices. Its plot will show weight values of near zero in all window positions and one will not be able to determine whether this is because it simply is a residue with no preference for either structure in all window positions, or because it favours the termini of helices. The fact that a single layer network is unable to differentiate between these sorts of situations, and performs as well as networks with hidden nodes, which would be able to differentiate, suggests that, either such situations do not arise, or that they are unimportant for secondary structure prediction. Generally, it is the ratio of a residue's occurrence in the helix structure to that of its occurrence in the non-helix structure that determines the weights in a single layer network. As a single layer network performs as well a those with hidden nodes, it must be this ratio that is important in determining whether a helix forms, and not the absolute values that make up the ratio. In the case of histidine one has a possible physical explanation for its weight values. The histidine ring with an

excess of positive charge forms an electrostatic bond with the C-terminus of the helix which has an excess of negative charge due to the helix dipole (Šali *et al.*, 1988). It will similarly avoid the N-terminus of a helix as it will also have a positive charge. As for the other charged residues, glutamic acid does show perhaps a weak preference for the N-terminus of a helix rather than its C-terminus, but aspartic acid does not shown any clear preference for either terminus. The case of lysine is strange as one would expect it to prefer the C-terminus of a helix like histidine. In fact it seems to prefer the N-terminus. In accordance with the expectation that helices are usually partially buried and therefore favour hydrophobic residues, all the hydrophobic residues, without polar groups are pro-helix, apart from proline.

It was shown in section 7.9 that window size does improve prediction success when training and testing on whole structures. It was also shown that this is not due to boundary regions upsetting prediction and that the difference in going from window size 7 to window size 10 and including 3 extra residues gave a greater improvement than including a further 3 residues in going from window size 10 to 13. This analysis is telling us that there are more pseudo-helix sequences and pseudo-non-helix sequences the shorter our window is. This is what one would expect. If one considers the extreme case of a window containing a single residue, then there will be many instances of a single residue appearing in both the helix and non-helix structures. With a longer window there are fewer pseudo sequences and this must be because the longer the sequence of pro-helix or pro-non-helix residues the more likely it is that part of that sequence will form the favoured structure. The fact that prediction on whole proteins is the same for all three window sizes used does not mean that their predictions on the individual residues are the same. In fact they are not, but it is probably not coincidence that the overall percentages of helix and non-helix are similar. This is probably due to the problem being solvable by a single layer network and our technique of simple averaging to predict the structure of an individual residue.

By looking at figure 7.9 one can see that from the point of view of real helices there seems to be a clear area of preference to the right of the original decision boundary in figure 7.8. This suggests that the boundary is real and not an artefact of the neural network. There are relatively few real helices that encroach across the boundary into the majority non-helix region (the white islands in the shaded region in figure 7.8). What is more, it is only a simple rule (weighted sum of the inputs) that needs to be followed in order to decide whether a sequence is a potential helix or not. The non-helix sequences are much less well behaved and encroach far across the decision boundary into the majority helix region (the shaded islands in the white region), making it somewhat less certain whether the sequence predicted helix is really in the helix configuration. Figure 7.8, which was deduced purely from the results of experiments with the neural network, has a very plausible physical explanation if one accepts the pseudo-helix sequences as potential helix forming sequences and the pseudo-non-helix sequences as potential non-helix forming sequences. This physical argument also explains why there are more pseudo-helix sequences than pseudo-non-helix sequences. At the extreme right of this figure there are no helices. The sequences here are those with large numbers of anti-helix residues such as glycines, prolines and serines. No possible arrangement of the whole protein could possibly stabilise these sequences in the helix structure. As one moves to the right in figure 7.8 one finds more and more sequences, the pseudo-non-helix sequences, that, under the right circumstances, can form helices, although the majority do not. As one moves further right still and crosses the decision boundary, one reaches those sequences that favour the formation of helices, but some of these sequences, the pseudo-helix sequences, are from disrupted helices in non-helix conformations. Even sequences that are strongly pro-helix can be found in non-helix conformations. The asymmetry in figure 7.8 is probably due to an entropy factor, as there are many non-helix structures that a helix can be disrupted to form, while a helix is a single structure requiring special circumstances for natural non-helix sequences to form.

So one can explain the main diagram, figure 7.8, by accepting pseudo-helix

sequence as potential helix forming sequence that is unable to do so, probably because of long range constraints in the formation of the folded protein. There is evidence that some polypeptides can form helices in an aqueous environment (Marqusee and Baldwin, 1987) and it is feasible therefore that in protein folding (Dill, 1990), some secondary structures form first and are then disrupted in the final folding phase to form tertiary structure. Our result in section 6.9 certainly supports one aspect of this physical interpretation of figure 7.8. In section 6.9 it was proposed that inaccessible helices may be less fussy about their sequence, due to the constraint of forming hydrogen bonds to satisfy their main chain atoms, than accessible helices, which have the choice of making bonds with water molecules. This argument was supported by the result that around 90% of accessible helices could be correctly predicted compared to around 75% of inaccessible helix sequences. In figure 7.8 this means that more of the inaccessible helices lie to the left of the decision boundary than accessible helices. In other words, more inaccessible helices have sequences that have been falsely predicted (pseudo-non-helix sequences), than accessible helices. This does indeed support the physical interpretation of figure 7.8 because it suggests the mechanism of hydrogen bond formation for the stabilisation of these unusual sequences in the helix conformation. Perhaps more convincing in supporting the idea that the pseudo-helix sequence is indeed potential helix-forming sequence, which is unable to form helix because of global constraints, are the photographs showing coil regions of pseudo-helix sequence in figure 7.13. Many of these pseudo-helix sequence regions are indeed helical like in structure. Probably the most convincing independent evidence for pseudo-helix sequence being potential helix sequence is the fact that they occur with a high degree of significance at the C- and N-termini of actual helices. This suggests that potentially longer helices have been disrupted by global constraints during folding. That more pseudo-helix sequence occurs at the C-termini than the N-termini, although not found to be statistically significant, can perhaps be explained by the following argument. The amino acid chain is produced N-terminus first on the ribosome. It is thought that chain folding happens at a faster rate than chain synthesis and it likely therefore

that the chain already formed will fold, unless some outside agent is preventing it, whilst synthesis is taking place. It will fold then in the N-terminus to C-terminus direction. This means that any potential helix will also form from its N-terminus and as it will be preceded by coil that is flexible, it will be free to position itself relative to the rest of the partly formed protein such that helix formation is favoured. As more potential helix-forming sequence is produced a point will be reached when the helix begins to extend beyond the bulk of the protein and will have to be disrupted in order to satisfy the requirement that the folding protein is globular. Although this argument can explain C-terminus pseudo-helix sequence it excludes the possibility of N-terminus pseudo-helix sequence altogether which is clearly not the case. It also demands that protein structures are dependent on the direction of folding, which is clearly not in agreement with the results of the protein denaturing experiments mentioned in the Introduction. However, perhaps these successful renaturing experiments were done on proteins whose structures do not depend on the direction of folding. As for the presence of N-terminus pseudo-helix sequence, this could be explained by further structural rearrangements during folding.

That beta-sheet has a distribution between that of helix and coil may be explained physically as follows. In the majority helix region it is, like coil, a structure that potential helix sequences can be disrupted to. It is, however, more likely that helix is disrupted to form beta-sheet than coil. This is probably because the main chain hydrogen bonds can be satisfied by the beta-sheet structure if a helix is unable to form. The reason for the preference of the beta-sheet structure to that of helix could be explained by the need of placing hydrophilic residues on the protein surface. This was discussed in the section on protein folding in the Introduction (see page 11). In the majority coil region, beta-sheet like helix will need special physical conditions for it to form. The proportion of beta-sheet sequences located between the two boundaries is greater than for both helix and coil, however. This could explain the moderate amount of success some researchers have had in predicting beta-sheet.

But what could be the possible explanation for proteins having evolved sequences that under different environmental conditions would form helices but cannot because of global constraints? It often appears to be the case that coil forming pseudo-helix sequence is found at the protein surface. Here water molecules will form hydrogen bonds with the main chain atoms and so the main energy loss will arise from hydrophobic side chains in contact with water. One could argue that this is energetically unfavourable and that proteins would have evolved structures that would be less costly in energetic terms. But as long as the overall energy equation balances there is no reason why helices cannot be disrupted. One could even speculate that these regions of pseudo-helix sequence have functional properties, perhaps taking on the helix structure when undergoing a biologically important conformational change. The interaction with other molecular species may exclude water molecules from satisfying the main chain hydrogen bonds which could then be satisfied by the formation of a helix with some functional role. Perhaps these regions have evolved from proteins that did indeed contain the whole undisrupted helices. If this is true it will be significant in the field of protein sequence alignment.

One may ask why other secondary structure prediction methods have not resulted in the conclusion that regions of false helix prediction are potential helix regions that have not been able to realise that potential due to global constraints. It is difficult to see how rule-based methods could have achieved this. Any rule based system that predicts regions of beta-sheet or coil as helices would be either considered wrong or insufficient. The alternative deduction that non-local effects are responsible and the rules are basically correct is too bold. This highlights the advantage of using neural networks. They do not invent rules, but make their predictions based solely on the information they are presented with. Conclusions, therefore, can be drawn about the information itself. In a rule-based system one is forced to make deductions about the rules, rather than the information. It is surprising, however, that other workers using neural networks have failed to home in on these special regions. In the papers by Holley and Karplus (Holley and

Karplus, 1989), and by Qian and Sejnowski (Qian and Sejnowski, 1988), the fact that a single layered network did as well as those with hidden nodes was also demonstrated, but the fact that both these papers quoted the results from networks with hidden nodes suggests the full significance was not realised. Only in the paper by Holley and Karplus is there mention of a similar finding to the main result of this work. To quote:

'A 20-hidden-unit network involves over 7000 weights and biases. Since there are only 8315 residues in the training set, the free variables in the network are sufficient for it to learn to reproduce most of the "idiosyncratic" aspects of the training set. In the process, however, the network loses some of its ability to generalise, and prediction accuracy goes down. This potential for a loss of correlation between training accuracy and predictive accuracy is a fundamental limitation of neural networks.'

This is the same as saying that there is a difference in the training and test sets such that learning the unusual data in the training set upsets test set prediction. Such a case has already been discussed. What is referred to here as idiosyncratic data are in fact the pseudo sequences. This idea of idiosyncratic data causing a decrease in prediction success is akin to the idea of over-learning, whereby it is the over-specialisation on the training data that causes the loss in the network's ability to generalise. To quote from the Ph.D by Richards (Richards, 1990):

'A way of picturing this is if learning is viewed as curve fitting. The entire data set can be thought of as a set of points and the mapping function produced by the network as a curve that is to be fitted through these points. The training set is a subset of these points and at the start of learning the function (or curve) will be just as poor for both the training and test set. As learning progresses the curve will become a closer fit and will tend to approach both subsets but if learning continues too far the curve may well become such a good fit to the training set that it will move away from the test set points.'

In this work the feature of over-learning affecting test set prediction was also assumed to be the cause of the peak in figure 7.1 and this hindered progress. In fact a peculiar effect was seen in figure 6.3 for a single layer network. Here the training set was slightly unlearnt resulting in a noticeable drop in prediction

success. This could possibly be explained by the Runge effect, whereby an over-fitting occurs, which is not the same as over-learning and is a direct consequence of the back propagation algorithm which minimises the squared error without limiting the maximum error (Chauvin, 1990). However, the overall situation here cannot be likened to over-learning or over-fitting as it is the learning of around 15% of the training set that causes the decrease in prediction success. The term idiosyncratic cannot be applied to 15% of the data either. The most probable explanation, however, that the other workers in this field have not found these regions of pseudo sequence is that their networks are not forced to learn them. One can deduce this from both the papers by Qian and Sejnowski, and Holley and Karplus. In the case of Qian and Sejnowski the network is trained only to 75%. In the case of Holley and Karplus one of their networks does achieve 90% on the training set, giving a lower test set prediction success than those trained to a lesser extent. This led them to the conclusion quoted above.

As already mentioned in the Introduction, reporting only the overall percentage of correct predictions along with the correlation coefficients for the individual secondary structures can be misleading. It turns out that it is possible to calculate from these values the percentages of correct predictions for the structures individually. It was very surprising to find that the percentage of correct predictions on coil was at most 40% in the work by Qian and Sejnowski, and by Holley and Karplus. Helix prediction, however, was in the low 90's and beta-sheet in the high 70's. This result is more like our result with the shifted boundary where 40% of coil was predicted correctly and 90% of helix. That helix prediction exceeds coil prediction is somewhat surprising given our result that if care is not taken to balance the training set, then coil prediction will be higher than helix as the number of coil examples outweighs the number of helix examples. However, in training with a balanced set, helix is always learnt more easily than non-helix as there are fewer pseudo sequences in the helix set. Perhaps this effect more than compensates for the unbalance in the training sets. These figures highlight a still unsatisfactorily answered question: what is the best measure of prediction

success? Ideally, one would like a single figure so that different methods can be easily compared. One figure can say as much as the overall percentage and the individual correlation coefficients if the prediction method gives the same percentage of correctly predicted residues for all two or three structures being predicted. With a neural network this should be achievable by shifting the boundaries around, or more simply by altering the decision thresholds. This was nearly achieved in this work where 67% of helix residues and 69.5% of non-helix residues were correctly predicted. Even if this is not possible it should be required for researchers to quote the percentages of correctly predicted residues for the individual structures. In addition different networks could be used to suit different problems. If, for example, a high degree of confidence is required for coil prediction, then a network with a shifted boundary like the one in this work could be used. Non-helix prediction accuracy was 92%, but this was at the cost of helix prediction accuracy which fell to around 32%. The non-helix predictions occurred on about one third of cases for whole proteins but in many cases these predictions were quite obvious in that they included sequences with high proportions of the anti-helix residues proline and glycine. So there were few coil predictions, but one could be fairly sure that residues predicted to be in the coil conformation were in that conformation. So it must be borne in mind that a price must always be paid: the more accurate the prediction required, the fewer predictions there will be.

As already mentioned in the section on secondary structure prediction Kneller *et al.* achieved 79% prediction success on all α proteins, 70% on all β proteins and 64% on α/β proteins. It is not surprising that prediction success on all α proteins is highest, as prediction success for helix is higher than for non-helix due to there being less pseudo-non-helix sequence than pseudo-helix sequence. The result for α/β proteins is lowest probably because a high proportion of pseudo-helix sequence is beta-sheet sequence.

One of the other major differences in this approach to that of most other neural network approaches is our use of structurally homogenous segments in training and testing. Most other methods use all possible windows from the proteins irrespective of whether they contained boundaries between structures or not. The aim here is to predict the structure of the central residue only. In the work by Bohr *et al.* (Bohr *et al.*, 1988), which is perhaps for comparison purposes more appropriate, as they also try to predict helix only, they achieve a prediction success of 73% with a correlation coefficient of 0.38. The best result achieved here was 73% with a correlation coefficient of 0.34. As they make no mention of testing for homology between the training and test sets, their slightly higher correlation coefficient may in fact be a consequence of homology. Bohr *et al.* used a window 51 residues in length, and it is significant that using all this extra information did not produce better results than this work, where a window size 10 was used and boundaries were not included. In fact the paper by Bohr *et al.* highlights what could be a major drawback in using a long window containing boundaries between structures. A neural network needs to be presented with a large number of examples for it to be able to generalise. With a short window the ratio of the number of examples to conformations will be greater than with a longer window, but on the other hand it may not be long enough to include all residues that affect conformation. With a longer window one may have included all the residues that affect conformation, but the aforementioned ratio will be too small for effective generalisation. So, in using a neural network one must be aware of this limitation and that there is a balance to be achieved. With a large window size such as 51, one has the situation where the number of possible conformations that the 51 residues can adopt is much larger than the number of examples. This may explain why the results of Bohr *et al.* are not significantly better than the results here. If, as here, the window contains no boundaries but always contains either all helix or non-helix sequence then the situation is different. For a long window within a helix the number of possible structural conformations within the window is one and so one would expect generalisation to be better than the case where the window spans more than one structure. Indeed the results of this work support

this. For sequences known to be either wholly helix or non-helix, a prediction success of 80% with a correlation coefficient of 0.52 was achieved. There is, however, another possible explanation for this. Prediction of the structure of the whole sequence can also be regarded as a prediction for the central residue only. But now the central residue will always reside within at least half a window's length of the termini of that structure. It has been shown that the pseudo-helix sequence is often located at helix non-helix boundaries, so by avoiding these boundaries one may achieve higher prediction success.

In order to deduce whether pseudo-helix sequences are really indistinguishable from real helix sequences, then, following the argument above, one may need a data bank as much as 20 times larger than it is at present. If the conclusion is that pseudo-helix sequences really are indistinguishable from real helix sequences, then the only plausible explanation is that they are potential helix forming sequences and would have formed helices under different circumstances, e.g. being buried within the protein, rather than being at its surface. In this work we have shown that such sequences exist up to a length of 13 residues and it is likely that longer examples exist - we were constrained by worsening statistics not to go beyond a window size 13. So, assuming then that these sequences are really indistinguishable from real helix sequences, a much longer window than 13 will be needed in order to determine, from the context of these potential helix sequences, whether the longer range influences are in favour of helix formation or not. As one can see from the photographs in figure 7.13 the coil regions of pseudo-helix sequence appear to be determined by very long range and possibly even global constraints. In the worse possible case one may need, then, to take the whole protein sequence into account. Even if this is not necessary, it is certain that a very much a longer window than those used at present in secondary structure prediction will be required.

To summarise our conclusions. A much larger structural data bank will be needed

in order to deduce whether the pseudo-helix sequences of the windows sizes used here are indeed distinguishable from real helix sequences. For the time present there exists, therefore, an unavoidable limit on helix prediction success. If it turns out that the pseudo-helix sequences are indistinguishable from helix sequences, then given the limitations discussed above concerning long windows, this may mean that secondary structure prediction with this method, or any other method based on information in the structural data bank, is impossible.

BIBLIOGRAPHY

Amit D.J., Gutfreund H. and Sompolinsky H., *Spin glass models of neural networks*, Phys. Rev., **A32**, 1007 (1985).

Anfinsen C.B., *Principles that govern the folding of protein chains*, Science, **181**, 223-230 (1973).

Bacon, D.J. and Anderson W.F., *Multiple sequence alignment*, J. Mol. Biol., **191**, 153-161 (1986).

Baum E. and Haussler D., *What size net gives valid generalization?*, Neural Computation, **1**, 151-160 (1989).

Bernstein F.C., Koetzle T.F., Williams G.J.B., Meyer E.F. Jr, Brice M.D., Rodgers J.R., Kennard O., Shimanouchi T. and Tasumi M., *The protein data bank: A computer-based archival file for macromolecular structures*, J. Mol. Biol., **112**, 535-542 (1977).

Bohr H., Bohr J., Brunak S., Cotterill R.M.J., Lautrup B., Nørskov L., Olsen O.H. and Petersen S.B., *Protein secondary structure and homology by neural networks (The α-helices in rhodopsin)*, FEBS Lett., **241 number 1,2**, 223-228 (1988).

Carpenter G.A. and Grossberg S., *Neural dynamics of category learning and recognition: Attention, memory consolidation, and amnesia*, in *Brain structure, learning and memory* (Davis J., Newburgh R. and Wegman E., editors), AAAS Symposium Series (1986).

Chauvin Y., *Generalization performance of overtrained back-propagation networks*, Lecture Notes in Computer Science, **412**, 46-55 (1990).

Chothia C., *The nature of accessible surface area in proteins*, J. Mol. Biol., **105**, 1-14 (1976)

Chou P.Y. and Fasman G.D., *Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins*, Biochemistry, **13**, 211-222 (1974a).

Chou P.Y. and Fasman G.D., *Prediction of protein conformation*, Biochemistry, **13**, 222-245 (1974b).

Cohen F.E. and Kuntz I.D., *Tertiary structure prediction*, in *Prediction of protein structure and the principles of protein conformation* (G.D.Fasman, editor), Plenum Press, 647-706 (1989).

Davies D.R., *A correlation between amino acid composition and protein structure*, J. Mol. Biol., **9**, 605-609 (1964).

Dayhoff M.O., Schwartz R.M. and Orcuttt B.C., in *Atlas of protein sequence and structure*, **Vol 5, Supplement 3**, (Dayhoff M.O., editor), NBRF, 345 (1978).

Devereux J., Haeberli P. and Smithies O., *A comprehensive set of sequence analysis programs for the VAX*, Nuc. Acid. Res., **12(1)**, 387-395 (1984).

Dill K.A., *Dominant forces in protein folding*, Biochemistry, **Vol 29, No 31**, 7133-7155 (1990).

Ellis R.J., *The molecular chaperone concept*, seminars in Cell Biology, **1**, 1-9 (1990).

Finkelstein A.V. and Ptitsyn O.B., *Why do globualr proteins fit the limited set of folding pattems?*, Prog. Biophys. Molec. Biol., **50**, 171-190 (1987).

Fisher R.A., *Statistical methods for research workers*, 13th **Edition**, Oliver and Boyd (1958).

Gardner E., Stroud N. and Wallace D.J, *Training with noise, and the storage of correlated patterns in a neural network model*, Edinburgh Preprint **87/394** (1987).

Garnier J., Osguthorpe D.J. and Robson B., *Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins*, J. Mol. Biol., **120**, 97-120 (1978).

Gibrat J.F., Garnier J. and Robson B., *Further developments of protein secondary structure prediction using information theory*, J. Mol. Biol., **198**, 425-443 (1987).

Gō N. and Noguti T., *Structural basis of hierarchical multiple substates of a protein*, Chemica Scripta, **29A**, 151-164 (1988).

Guzzo A.V., *The influence of amino acid sequence on protein structure*, Biophys. J., **5**, 809-822 (1965).

Hebb D.O., *The organization of behavior*, John Wilely & Sons, New York (1949).

Hinton G.E., McClelland J.L. and Rumelhart D.E., *Distributed representations*, in *Parallel distributed processing: Explorations in the microstructure of cognition, Vol.1: Foundations* (Rumelhart D.E. and McClelland J.L., editors), MIT Press, 318-362 (1986).

Hinton G.E. and Sejnowski T.J., *Analyzing cooperative computation*, Proceedings of the Fifth Annual Conference of the Cognitive Science Society, (1983).

Holley L.H. and Karplus M., *Protein secondary structure prediction with a neural network*, Proc. Natl. Acad. Sci. *U.S.A.*, **86**, 152-156 (1989).

Hopfield J.J., *Neural networks and physical systems with emergent collective computational abilities*, Proc. Natl. Acad. Sci. *U.S.A.*, **79**, 2554-2558 (1982).

Jones T.A. in *Computational Crystallography*, (Sayne D., editor), Oxford University Press, London and New York, 303-317 (1982).

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

Kabsch W. and Sander C., *How good are predictions of protein secondary structure?*, FEBS Lett., **155**, 179-182 (1983b).

Kabsch W. and Sander C., *On the use of sequence homologies to predict protein structure: Identical pentapeptides can have completely different conformations*, Proc. Natl. Acad. Sci. *U.S.A.*, **81**, 1075-1078 (1984).

Kneller D.G., Cohen F.E. and Langridge R., *Improvements in protein secondary structure prediction by an enhanced neural network*, J. Mol. Biol., **214**, 171-182 (1990).

Lesk A.M., Boswell D.R., Lesk V.I., Lesk V.E. and Bairoch A., *A cross-reference table between the protein data bank of macromolecular structures and the national biomedical research foundations-Protein identification resource amino acid sequence data bank*, Protein Seq Data Anal, **2**, 295-308 (1989).

Lewis P.N., Gō N., Gō M., Kotelchuck D. and Scheraga H.A., *Helix probability profiles of denatured proteins and their correlation with native structures*, Proc. Natl. Acad. Sci. *U.S.A.*, **65**, 810-815 (1970).

Lim V.I., *Structural principles of the globular organization of protein chains: A stereochemical theory of globular protein secondary structure*, J. Mol. Biol., **88**, 857-872 (1974a).

Lim V.I., *Algorithms for prediction of α-helices and β-structural regions in globular proteins*, J. Mol. Biol., **88**, 873-894 (1974b).

Marqusee S. and Baldwin R.L., *Helix stabilization by $Glu^{-}...Lys^{+}$ salt bridges in short peptides of de novo design*, Proc. Natl. Acad. Sci. *U.S.A.*, **84**, 8898-8902 (1987).

McGregor M.J., Flores T.P. and Sternberg M.J.E., *Prediction of β-turns in proteins using neural networks*, Protein Engineering, **2 no.7**, 521-526 (1989).

Minsky M. and Papert S., *Perceptrons: An introdution to computational geometry*, MIT Press (1969).

Pauling L. and Cory R.B., *Atomic coordinates and structure factors for two helical configurations of polypeptide chains*, Proc. Natl. Acad. Sci. *U.S.A.*, **37**, 235-240 (1951).

Ptitsyn O.B. and Finkelstein A.V., *Connection between the secondary and primary structures of globular proteins*, Biofisika, **15**, 757-768 (1970).

Qian N. and Sejnowski T.J., *Predicting the secondary structure of globular proteins using neural network models*, J. Mol. Biol., **202**, 865-884 (1988).

Ramachandran G.N. and Sasiekharan V., *Conformation of polypeptides and proteins*, Adv. Prot. Chem., **23**, 283-437 (1968).

Rees A.R. and Sternberg M.J.E., *From Cells to Atoms. An illustrated introduction to Molecular Biology*, Blackwell Scientific Publications (1984).

Richards G.D., *Implementation and Capabilities of layered feed-forward Networks*, Ph.D thesis, Edinburgh University, (1990).

Richards G.D. and Tollenaere T., *Documentation for Rhwydwaith Version 2.1*, ECS Note ECSP-UG-7 (1989).

Richardson J.S. and Richardson D.C., *Amino acid preferences for specific locations at the ends of α helices*, Science, **240**, 1648-1652 (1988).

Richardson J.S. and Richardson D.C., *Principles and patterns of protein conformation*, in *Prediction of protein structure and the principles of protein conformation* (Fasman G.D., editor), Plenum Press, 1-98 (1989).

Rumelhart D.E., Hinton G.E. and Williams R.J., *Learning internal representations by error propagation*, in *Parallel distributed processing: Explorations in the microstructure of cognition, Vol.1: Foundations* (Rumelhart D.E. and McClelland J.L., editors), MIT Press, 318-362 (1986).

Rumelhart D.E. and Zipser D., *Feature discovery by competitive learning*, Cognitive Science, **9**, 75-112 (1985).

Šali D., Bycroft M. and Fersht A.R., *Stabilization of protein structure by interaction of α-helix dipole with a charged side chain*, Nature, **335**, 740-743 (1988).

Schiffer M. and Edmundson A.B., *Use of helical wheels to represent the structures of proteins and to identify segments with helical potential*, Biophys. J., **7**, 121 (1961).

Schulz G.E. and Schirmer R.H., *Principles of protein structure*, Springer-Verlag (1978).

Sternberg M.J.E. and Islam S.A., *A relational database of protein structure*, Biochemical Society Transactions, **Vol 17, Issue 5**, 845-847 (1989).

Wold S., Albano C., Dunn III W.J., Edlund U., Esbensen K., Geladi P., Hellberg S., Johansson E., Lindberg W. and Sjöström, *Multivariate data analysis in chemistry*, in *Chemometrics: Mathematics and statistics in chemistry*, Mathematical and Physical Sciences, **138, Series C**, (Kowalski B.R., editor), NATO ASI Series, 17-95 (1983).

Zimm B.H. and Bragg J.K., *Theory of phase transition between the helix and random chain in polypeptide chains*, J. Chem. Phys., **31**, 526-535 (1959).

APPENDIX A

Listed below are the proteins from the Brookhaven data bank used for training and testing. Those in the test set are underlined. (After Qian and Sejnowski, 1988)

Code	Protein name	N	n_c	h	e	-
1mbs	Myoglobin (met)	1	All	111	0	42
1mlt	Melittin	2	1	22	0	4
1nxb	Neurotoxin 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
1ru3	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 $b5$ (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 $c3$	1	All	27	10	70
2cyp	Cytochrome c peroxidase	1	All	134	16	143
2dhb	Haemoglobin (horse, deoxy)	2	All	172	0	116
2fdl	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 c2 (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 c551 (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
5ate	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_c, subunit numbers used in this study; h, α-helix; e, β-sheet; -, coil.

Code	Protein name	N	n_c	h	c	
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
1cnc	Carbonic anhydrase form c	1	All	18	68	170
1cc5	Cytochrome c5 (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	a-Cobratoxin	1	All	4	16	51
1cy3	Cytochrome c3	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
1ger	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
1gpl	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

APPENDIX B

Shown is primary sequence, the Kabsch and Sander's secondary structure assignments and helix prediction with the neural network for the proteins in the test set (Brookhaven codes are in bold and the bold vertical bars indicate the start of a new subunit. See Appendix A for protein names). The overall prediction success is 73% with a correlation coefficient of 0.34.

Row 2	Row 3
H=Alpha-Helix	H=Alpha-Helix Prediction
E=Beta-Sheet	N=Non-Alpha-Helix Prediction
T=Reverse-Turn	
S=Bend	
B=Bridge	
G=3/10 Helix	

```
1ENLKLGFLVKQPEEPWFQTEWKFADKAGKDLGFEVIKIAVPDGEKTLNAI
A
B       EEE    SSTTHHHHHHHHHHHHHSSS    EEE   SHHHHHHHH
PHHHHHNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHHHHNNNNNNNNNHHHH


DSLAASGAKGFVICTPDPKLGSAIVAKARGYDMKVIAVDDQFVNAKGKPM

HHHHHT        B   S SS TTHHHHHHHHH      B SS    STT
NNNNNNNNNNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHHHNNNNNNNNN


DTVPLVMMAATKIGERQGQELYKEMQKRGWDVKESAVMAITANELDTARR

 SS     B SHHHHHHHHHHHHHHHHHHT  STT  EEE    SSGGGHH
NNHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH


RTTGSMDALKAAGFPEKQIYQVPTKSNDIPGAFDAANSMLVQHPEVKHWL

HHHHHHHHHHHHS  SSS EE   SSSSSHHHHHHHHHHHHH  S  S E
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHHNNNNHHHHH


IVGMNDSTVLGGVRATEGQGFKAADIIGIGINGVDAVSELSKAQATGFYG

EE SSSTTTHHHHHHHSSS  STT    EESSTTTHHHHSSSS  S
NNNNNNNNNNNNNNNNNNNNNHHHHNNNNNNNNNNNNNNNNNNNNNNNN
```

SLLPSPDVHGYKSSEMLYNWVAKDVEPPKFTEVTDVVLITRDNFKEELEK

EE TTTTHHHHHHHHHHHHT S B B SSSTTTGGGT
NNNNNNNNNNNNNNHHHHNNNNNNNNNNNNNHHHHHHHHHHHHHHHHHH

KGLGGK1APAFSVSPASGASDGQSVSVSVAAAGETYYIAQaAPVGGQDAaN
A
TT TT C EEEEE SS SS EEEEEEES SEEEEEEE EETTEE
HNNNNNXNNNNNNNNNNNNNNNNNNNNNHHHHHHHNNNNNNNNNNNNN

PATATSFTTDASGAASFSFTVRKSYAGQTPSGTPVGSVDbATDAbNLGAG

TTT EEE SS EEEEE SEEEEE TTS EEEEEETTTS EEEEE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NSGLNLGHVALTFG1GFPIPDPYCWDISFRTFYTIVDDEHKTLFNGILLLS
H
SS M SS GGG S HHHHHHHHHHHHHHHHHH
NNNNNNNNHHHHHHQNNNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHHH

QADNADHLNELRRCTGKHFLNEQQLMQASQYAGYAEHKKAHDDFIHKLDT

H SHHHHHHHHHHHHHHHHHHHHHHHHTT TTHHHHHHHHHHHHHHHHT
HHHHHHHHHHHHHHNNNNNHHHHHHHHHHHHHHHHHHHHHHHHHHHNN

WDGDVTYAKNWLVNHIKTIDFKYRGKI1RDFTPPTVKILQSSaDGGGHFPP
I
S HHHHHHHHHHHHHHTTGGGTTT G B EEEEE SGGGT SSS
NNNNNNHHHHHHHHHHHHHHHHHHHNNENNNNNNNNNNNNNNNNNNN

TIQLLbLVSGYTPGTINITWLEDGQVMDVDLSTASTTQEGELASTQSELT

EEEEEEEEEEEBSS EEEEESSSSB SEEE EE TTS EEEEEEEE
NNNNNNNNNNNNNNNNNNNHHHHHHHHHHNNNNNNNNNNNNNNNNNNN

LSQKHWLSDRTYTbQVTYQGHTFEDSTKKcADSNPRGVSAYLSRPSPFDL

EEHHHHHTT EEEEE TTSTTS EEE SS EEEEE TTTT
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

FIRKSPTITdLVVDLAPSKGTVNLTWSRASGKPVNHSTRKEEKQRNGTLT

STTS EEEEEEES SSS B EEETTB STT EEEETTEEE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

```
VTSTLPVGTRDWIEGETYQdRVTHPHLPRALMRSTTKTSGPRAAPEVYAF

EEEEEE  HHHHHHT  EEEEEB TTSSS EEEEE   SS  B  EEEEE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


ATPEWPGSRDKRTLAeLIQNFMPEDISVQWLHNEVQLPDARHSTTQPRKT

   SGGGGSSEEEEEEEEESBBSS  EEEEESSS  TTTTS   B  EE
NNNNNNNNNNNHHHHHHHHHHHHHNHHHHHHNNNNNNNNNNNNNNNNNN


KGSGFFVFSRLEVTRAEWEQKDEFIeRAVHEAASPSQTVQRAVSVNPGK1R
                                                    N
TTS EE EEEEEEEHHHHHTT   EEEEE  SSSSSS EEEE        X
NNNNNHHHHHHHHHHHHHHHHHHHHHHHHHHNNNNNNNNNNNNNNNNNBN


IaFNQHSSQPQTTKTbSPGESSaYHKQWSDFRGTIIERGbGcPTVKPGIK

EEE  STT S  EEE  TT   EEEEEEEETTEEEEEEES     SS
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


LScdESEVdNN1IPEYVDWRQKGAVTPVKNQGSaGSCWAFSAVVTIEGIIK
           P
EEEE STTTT P  S EEGGGGT       B  SSB HHHHHHHHHHHHHHH
NNNNNNNNNNND NNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHHHHHHHHH


IRTGNLNQYSEQELLDbDRRSYGaNGGYPWSALQLVAQYGIHYRNTYPYE

HHHS     B HHHHHHH TTS TTB   HHHHHHHHHHT B BTTTS
NNNNNNNNNNNHHHHNNNNNNNNNNNNNNNNNNNHHHHHHNNNNNNNNNN


GVQRYbRSREKGPYAAKTDGVRQVQPYNQGALLYSIANQPVSVVLQAAGK

SS      TTTT S SB  SEEEE  SS HHHHHHHHHHS EEEEE   SH
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHHHH


DFQLYRGGIFVGPcGNKVDHAVAAVGYGPNYILIKNSWGTGWGENGYIRI

HHHT  SSEE  S  S   EEEEEEEEETTEEEEE SB TTSTBTTEEEE
HHHHNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


KRGTGNSYGVcGLYTSSFYPVKN1TYTTRQIGAKNTLEYKVYIEKDGKPVS
                        P
E  S  TT GGGTTS  EEEE  Y   EEE S SSSS   EEEB SSSB
NNNNNNNNNNNNNNNNNNNNNNNPNNNNNNNNNHHHHHHHHHHNNNNNNNNN
```

```
AFHDIPLYADKEDNIFNMVVEIPRWTNAKLEITKEETLNPIIQNTKGKLR

TTTS   SS STTTT    B      SS  B      SSSS         SSSS
NNNNNNNNNNNNHHHHHHHHHNNNNNNNNHHHHHHHHNNNNNNNNNNHHHHH


FVRNCFPHHGYIHNYGAFPQTWEDPNVSHPETKAVGDNNPIDVLQIGETI

B     SSS S   S      TT     SS    TT TT    SS    EEE   SS
HHNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHH


AYTGQVKEVKALGIMALLDEGETDWKVIAIDINDPLAPKLNDIEDVEKYF

  TT   B  EEEEEEEEE SS EEEEEEEE TTSSSTTT SSSSGGGTTS
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHNNNNNNNNNNNNNNNNNNNN


PGLLRATDEWFRIYKIPDGKPENQFAFSGEAKNKKYALDIIKETHNSWKQ

TTSHHHHHTTTHHHHHHHH       BSGGG  B HHHHHHHHHHHHHHHHH
NNNHHHHHHHHHNNNNNNNNNNNNNNNNNNNNNHHHHHHHHHHHHHHHHH


LIAGKSSDSKGIDLTNVTLPDTPTYSKAASDAIPPASPKADAPIDKSIDK

HHTSSSSS    S     TT SSSTT  TTTTTTS                TTSS
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNH


WFFI2LPSYVDWRSAGAVVDIKSQGEaGGCWAFSAIATVEGINKITSGSLI
    A
 S  C  S EEGGGGT        B TTS   HHHHHHHHHHHHHHHHHHS
HHHHTNNNNNNNNNNNNNNNNNNNNNNNNNNHHHHHNNNNNNNNNNNNNN


SLSEQELIDbGRTQNTRGaDGGYITDGFQFIINDGGINTEENYPYTAQDG

 B HHHHHHH   BTTB GGG    HHHHHHHHHHHT B BTTTS    SS
NNNNHHHNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


DbDVALQDQKYVTIDTYENVPYNNEWALQTAVTYQPVSVALDAAGDAFKQ

   HHHHH    B    EEEE   TT HHHHHHHHHHS EEEEE    SHHHHH
NNNNHHHHHHHHHHHNNNNNNNNNNNNHHHHHHHNNNNNNNNNNNNNHNNN


YASGIFTGPcGTAVDHAIVIVGYGTEGGVDYWIVKNSWDTTWGEEGYMRI

  SSEE      SS  EEEEEEEEEEETTEEEEEEE SB TTSTBTTEEEE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

```
LRNVGGAGTcGIATMPSYPVKY2ANIVGGIEYSINNASLaSVGFSVTRGAT
                       A
E     TT GGGTTSS EEEE  L EEEET EEEETTTEEEE  EEEEETTE
NNNNNNNNNNNNNNNNNNNNNNPNNNNNNNNNNNNNNNNNNNNNNNNNN


KGFVTAGHaGTVNATARIGGAVVGTFAARVFPGNDRAWVSLTSAQTLLPR

EEEEE GGG  TT EEEETTEEEEEEEEEE SBS EEEEEE TT EEEEE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


VANGSSFVTVRGSTEAAVGAAVbRSGRTTGYQbGTITAKNVTANYAEGAV

EEETTEEEE  B      TT EEEEEETTTEEEEEEEEEEEEEEETTEEE
NNNNNNNNNNNNNNNNNNNHHNNNNNNNNNNNNNNNNNNHHHHHHHHNNN


RGLTQGNAcMGRGDSGGSWITSAGQAQGVMSGGNVQSNGNNcGIPASQRS

EEEEEE    BTT TT EEE TTSBEEEEEEEE   TTSBSTTS GGG
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


SLFERLQPILSQYGLSLVTG2APKAPADGLKMDKTKQPVVFNHSTHKAVKC
                    C
EEEEEHHHHHHHHT EE   D       S EEE SSSS EEE SGGGTTS H
NNNNNNNNNNNNNNNNNNNNVNNNNNNNNNNNNNNNNNNNNNNNNNNNN


GDCHHPVNGKENYQKCATAGCHDNMDKKDKSAKGYYHAMHDKGTKFKSCV

HHHS EETTEE  S TTSTTSS B  TT  STTBHHHHHH SS SS  HH
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHHHHHHHNNNNNNNNNNNN


GCHLETAGADAAKKKELTGCKGSKCHS2VASYDYLVIGGGSGGLASARRAA
                           G
HHHHHHHTT HHHHHHHH SSSSSS  R    S  EES BTTHHHHHHHHH
NNNNNNHHHHHHHHHHHNNNNNNNNNNSNNNNNNNNNNNNNNNHHHHHH


ELGARAAVVESHKLGGTaVNVGaVPKKVMWNTAVHSEFMHDHAVYGFPSC

HTT  EEEEESS TBHHHHHHSHHHHHHHHHHHHHHHHSS GGGS
HHHHHHHHHHNNNNNNNNNNNNNNNNHHHHHHHHHHHHHHHNNNNNNNNN


EGFFNWRVIKEKRDAYVSRLNAIYQNNLTKSHIEIIRGHAAFTSDPKPTI

     HHHHHHHHHHHHHHHHHHHHHHHHTTEEEEES   B SSSSS B
NNNNHHHHHHHHHHHHHHHHHHHNNNNNHHHHHHHHHHHHHNNNNNNNNN
```

```
EVSGKKYTAPHILIATGGMPSTPHESQIPGASLGITSDGFFQLEELPGRS

 STT  B  SSEEE     EE     STTSSS TT EEHHHHTT SS  SSE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


VIVGAGYIAVEMAGILSALGSKTSLMIRHDNVLRSFDSMISTNCTEELEN

EEE  SHHHHHHHHHHHHTT   EEE SSSSS TTS HHHHHHHHHHHTT
NNNNNNHHHHHHHHHHNNNNHNNHHHHHHHHHHHHNNNNNNNNNNNHHH


AGVEVLKFSQVKEVKKTLSGLEVSMVTAVPGRLPVMTMIPDVDCLLWAIG

SS  EETTEEEEEEETTSSS EEEEEE SSSS  EEEE  SS EEEE S
HHHHHHHHHHHHHHHHHHHHHHHHNNNNNNNNNNNNNNNNHHHHHHHHN


RVPNTKDLSLNKLGIQTDDKGHIIVDEFQNTNVKGIYAVGDVCGKALLTP

EE S TTS TTTTT  B TTS B   SSSB SSSSEEE STTSSS   HH
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHHHHHH


VAIAAGRKLAHRLFEYKEDSKLDYNNIPTVVFSHPPIGTVGLTEDEAIHK

HHHHHHHHHHHHHHS  TT    SSS  EEE  SS EEEEE  HHHHHHH
HHHHHHHHHHHHHHHHHNNNNNNNNNNNNNNNNNNNNNNNNNHHHHHHH


YGIENVKTYSTSFTPMYHAVTKRKTKCVMKMVCANKEEKVVGIHMQGLGC

S SSSEEEEEEEE  GGGSS SS   EEEEEEEETTTTEEEEEEEESTTH
NNNNNNNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHHHHHHHNNNNN


DEMLQGFAVAVKMGATKADFDNTVAIHPTSSEELVTLR2PIVDTGSVAPLS
                                        L
HHHHHHHHHHHTTT BHHHHHTS   SSSSGGGGSS  H   SSS
NNHHHHHHHHHHHHHHHHHHNNNNNNNNNNNNNNNNNBNNNNNNNNNNN


AAEKTKIRSAWAPVYSTYETSGVDILVKFFTSTPAAQEFFPKFKGLTTAD

HHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHH GGGGGG GGGTT  SHH
NHHHHHHHHNNNNNNNNNNNNNHHHHHHNNNNNNNNNNNNNHHHHHH


ELKKSADVRWHAERIINAVDDAVASMDDTEKMSMKLRNLSGKHAKSFQVD

HHHT HHHHHHHHHHHHHHHHHHTTT HHHHHHHHHHHHHHHHHTS
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHNNNNNNNN
```

```
PEYFKVLAAVIADTVAAGDAGFEKLMSMICILLRSAY2AQSVPYGVSQIKA
                                          S
GGGHHHHHHHHHHHHHTT HHHHHHHHHHHHHHHTT B     HHHHTTS
NNNHHHHHHHHHHHHHNNNNHHHHHHHHHHHHHHHHHHTNNNNNNNNNNNN

PALHSQGYTGSNVKVAVIDSGIDSSHPDLKVAGGASMVPSET|PNFQDDNS

TTHHHHT SS S EEEEEES   SS TT    EE   SSS   |  S SS
NNNNNNNNNNNNNHHHHHNNNNNNNNNNNNNNNNNNNNNNN|NNNNNNNN

HGTHVAGTVAALNNSIGVLGVAPSSALYAVKVLGDAGSGQYSWIINGIEW

HHHHHHHHH    STTS   S TTSEEEEEE S SSS   HHHHHHHTTT
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNHHH

AIANNMDVINMSLGGPSGSAALKAAVDKAVASGVVVVAAAGNEGSTGSSS

TTTTT SEEEE    SS  SHHHHHHHHHHHTS EEEE   SS   TTS
HHHHHHHHHHNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHNNNNNNNNNNN

TVGYPGKYPSVIAVGAVDSSNQRASFSSVGPELDVMAPGVSIQSTLPGNK

   TTTSSSSSEE  EE SSS   TT   STT    EE SSEEEEETTTE
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

YGAYNGTSMASPHVAGAAALILSKHPNWTNTQVRSSLQNTTTKLGDSFYY

EEEE SSTTTHHHHHHHHHHHHHH TT  HHHHHHHHHHT    S TTTT
NNNNNNNNNNNNNNNHHHHHHNNNNNNNNNNNNNNNNNNNNNNNNNNN

GKGLINVQAAAQ3GKVKVGVDGFGRIGRLVTRAAFNSGKVDIVAINDPFID
          G
TT    TTTT  P  S EEEE  STHHHHHHHHHHHTT  S   EE SSS
NNNNNHHHHHHHDNNNNNNNNNNNNHHHHHHHHHHNNNNNNNNNNNNNHH

LHYMVYMFQYDSTHGKFHGTVKAEDGKLVIDGKAITIFQERDPENIKWGD

HHHHHHHHHEETTTEE  S EE  SSEEEETTEEEEEE  SSGGGS  TT
HHHHHHHHNNNNNNNNNNNNNHNHHHHHHHHHHHHHHNNNNNNNNNNNN

AGTAYVVESTGVFTTMEKAGAHLKGGAKRIVISAPSADAPMFVMGVNHFK

TT SEEEE  SS  SHHHHHHHHHS SEEEEES  SSS  B TTT GGG
NNNNNNNNNNNNHHHHHHHHHNHHHHHHHHHHHHNNNNNNNNNNNHNHHHH
```

```
YANSLKIISNASCTTNCLAPLAKVIHDHFGIVEGLMTTVHAITATQKTVD

  TT SEEEE  TTHHHHHHHHHHHHHHH  EEEEEEEEE   SS  SSB
HHHHHNNNNNNNNNNNNNNNHHHHHHHHHHHHHHHHHHHHHHHHNNNNNN


SPSGKLWRGGRGAAQNLIPASTGAAKAVGKVIPELDGKLTGMAFRVPTAN

    TT  TTB SSS  EEEE  HHHHHHHHSSSSSSSEEEEEEE S  S
NNNNNNNNNNNNNNNNNNNNNNNNNNHHNNNNNNNNNNNHHHHHHNNNNN


VSVLDLTCRLEKPAKYDDIKKVVKEASEGPLKGILGYTEDEVVSDDFNGS

  EEEEEEE SS   HHHHHHHHHHHHTTTSSS      S   TT  SS
NNNNNNNNNNNNNNNNHHHHHHHHHHNNNNNNNNNNNNNNNNNNNNNNN


NHSSIFDAGAGIELNDTFVKLVSWYDNEFGYSERVVDLMAHMASKE6HPTF
                                                  A
S SS   SSTT EEETTEEEEE    HHHHHHHHHHHHHHHHHHH P TTH
NNNNNNNNNNNNNHHHHHHHHHHHNNNNNNNNHHHHHHHHHHHHHHHHHINNNN


NKITPNLAEFAFSLYRQLAHQSNSTNIFFSPVSIATAFAMLSLGTKADTH

HHHHHHHHHHHHHHHHHHHHHH SS EEE HHHHHHHHHHHHTT  HHHH
NNNNNNHHHHHHHHHHHHNNNNNNNNNNNNNNNHHHHHHHHHHHNNNHHHH


DEILEGLNFNLTEIPEAQIHEGFQELLRTLNQPDSQLQLTTGNGLFLSEG

HHHHHHTT  TTTS HHHHHHHHHHHHHHTS  TTSEEEEEEEEEETT
HNHHHNNNNHHHHNNHHHHHHHHHHHHNNNNNNNNNNNNNNNNNNNNH


LKLVDKFLEDVKKLYHSEAFTVNFGDTEEAKKQINDYVEKGTQGKIVDLV

    HHHHHHHHHHS  EEEEE TT HHHHHHHHHHHHHHHTTTS   S
HHHHHHHHHHHHHHHHHHHNNNNNNNNNNNNHHHHNNNNNNNNNHHHHHH


KELDRDTVFALVNYIFFKGKWERPFEVKDTEEEDFHVDQVTTVKVPMMKR

S   TT  EEEEEEEEEE  BSS   GGG EEEEEE SSS EEEEEEEEE
HHHHHHHHHHHHHHHHHHHNNNNNNNNNNHHHHHHHHNNHHHHHHHHHHH


LGMFNIQHCKKLSSWVLLMKYLGNATAIFFLPDEGKLQHLENELTHDIIT

EEEE EEEETTTTEEEEEEEBSSSEEEEEEEE TT HHHHHHH  HHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHNNNNNNNNNHHHHHHHHHHH
```

```
KFLENEDRRSASLHLPKLSITGTYDLKSVLGQLGITKVFSNGADLSGVTE

HHHH     EEEEEEEE EEEEEEEE HHHHHHHH  STTSTT    TTT S
HHHHHHHHHNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN


EAPLKLSKAVHKAVLTIDEKGTEAAGAMFLEAIPM|SIPPEVKFNKPFVFL

SS    EEEEEEEEEEEE SS EEEEEEEEEEE    |  S EEE  S EEEE
NNNNNHHHHHHHHHHHHHHHHHHHHHHHHHHHHH|NNNNNNHHHHHHHHH


MIEQNTKSPLFMGKVVNPTQ

EEETTTTEEEEEEES S
HHHHHNNNNNNNNNNNNNN
```

APPENDIX C

Shown below are pseudo-non-helix sequences (helix sequences predicted non-helix) denoted by H, aligned with its best match successful non-helix sequence, denoted by N. The numbers are the scores using a 100 PAM table (see Table?).

```
   H RdcQTHDNeY     H FIgNeDRNAA     H AHYYAGVTYD     H FAYPDTHRHR
21 N EECRDKASPY  20 N FAGGSSNNSG  20 N TGYDASIGYG  34 N YSYTDANKSK

   H dcQTHDNeYK     H IRFYDNLQQY     H HYYAGVTYDY     H TKIGERQGQE
27 N QCHTVENGGK  22 N QRFFDSFGNL  22 N YSFNSVVLDY  21 N IAKGERQSPV

   H cQTHDNeYKQ     H YDNLQQYLNV     H YYAGVTYDYY     H RRRTTGSMDA
30 N CHTVENGGKH  28 N YEGVQRYbRS  17 N SFNSVVLDYV  26 N QREATgTSEV

   H QTHDNeYKQA     H FSQVCTHLDT     H YAGVTYDYYK     H TTGSMDALKA
26 N DTHTAKYDPS  32 N FTQGLKHLDD  19 N FNSVVLDYVP  32 N HSGSVTALNA

   H AFIgNeDRNA     H GDLFSLGGVT     H GVTYDYYKNV     H TGSMDALKAA
21 N ASIGYGDGSA  23 N GNLVGFAGAQ  20 N GLAQDYVKAG  39 N SGSVTALNAT

   H FIgNeDRNAA     H DLFSLGGVTA     H NEAISDIFGT     H GYKSSEMLYN
20 N FAGGSSNNSG  21 N NGADLSGVTE  33 N DEVVSDDFNG  28 N GFPSCEGFFN

   H VARSNFNVcR     H FSLGGVTAVQ     H IPKEQARIKT     H YKSSEMLYNW
21 N VSRLGdNVTR  28 N LAFSSINTVQ  25 N SPKADAPIDK  29 N YRLIQFHFHW

   H CIDCHALKKK     H SLGGVTAVQE     H PRYTCQREFA     H TIVDDEHKTL
41 N CAQCHTVDKG  26 N AFSSINTVQG  27 N AIYYcARNYY  28 N APYNKEHKNL

   H AAWGATLDTF     H YNMINTVKSD     H TRTRLSFQTS     H HLNELRRCTG
26 N SYWGSTVKNS  34 N FSSINTVQGS  33 N QRMFLSFPTT  24 N QLPDARHSTT

   H AWGATLDTFF     H IASANAIRNY     H RTRLSFQTSM     H RCTGKHFLNE
31 N CWDISFRTFY  30 N ITAKNVTANY  37 N RMFLSFPTTK  19 N RTPGSRNLdN

   H GATLDTFFGM     H IVAALPTIKY     H KSIVDFVKNH     H CTGKHFLNEQ
37 N GPNLNGLFGR  23 N GVGTVPMTDY  40 N ASVVbFLNNF  14 N YYGKGLINVQ

   H TVRDYTQMND     H GQQEAARAGE     H IPTAQETWLG     H TDGFQFIIND
28 N DVRQYVQGaG  42 N GHQENAKNEE  28 N GATADSTYLG  23 N NDQMQFNTNA

   H PQNFRLLGNV     H APAVDAHYYA     H TNWAIGLSVA     H SRLNAIYQNN
25 N PDDSRVIAHT  28 N APSADAPMFV  20 N EHWHKDFPIA  32 N ETLNPIIQNT

   H dcETHDNeYR     H AVDAHYYAGV     H LQGRLFAYPD     H NAIYQNNLTK
16 N QCHTVENGGK  22 N TADSTYLGAI  17 N GQAEGYSYTD  28 N NPIIQNTKGK

   H ETHDNeYRDA     H VDAHYYAGVT     H GRLFAYPDTH     H SPHVAGAAAL
23 N DTHTAKYDPS  23 N VEPPKFTEVT  22 N GNLVGFAGAQ  26 N GTHSGSVTAL

   H AFIgNeDRNA     H DAHYYAGVTY     H LFAYPDTHRH     H TPNLAEFAFS
21 N ASIGYGDGSA  20 N ATGYDASIGY  31 N LVVYPWTQRF  22 N VGNLVGFAGA
```

Shown below are pseudo-helix sequences (non-helix sequences predicted helix) denoted by N, aligned with its best match successful helix sequence, denoted by H.

```
   N LAVLGIFLKV     N QKLKIAKVFK     N GELFLARSVT     N AAVGNLVGFA
27 H PNVQALFQKV  24 H DMLRDAMVAK  18 H EELNSAWTIA  26 H VEVKSIVDFV

   N AVLGIFLKVG     N AWTLVGIVSW     N VSFEATFAFL     N NLVGFAGAQD
28 H NVQALFQKVV  27 H EWALQTAVTY  43 H VSIATAFAML  23 H AfEAFIgNeD

   N VSLTbLVKGF     N HEHAEVVFTA     N QKIKVEKQII     N LVGFAGAQDA
30 H IALAKNVAAF  32 H RKHGNTVLTA  21 H KKVKAAKKAA  25 H YLSFAAAMNG

   N FFLYSKLTVD     N QVRIVSHKLH     N HCELSTELAV     N AGAQDAALGG
33 H FSLYRQLAHQ  25 H QAQEVHEKLR  18 H HCLLVTLAAH  31 H ADTHDEILEG

   N PDVLKaLKAP     N VRIVSHKLHV     N KYAWVAIRYT     N DAALGGFVIA
41 H PDALKAQAAA  28 H FKLLSHSLLV  23 H KYLSIVKEYA  40 H DEMLQGFAVA

   N LSTGKWSIAY     N RIVSHKLHVR     N AWVAIRYTYL     N AALGGFVIAb
32 H ELNSAWTIAY  29 H KLLSHSLLVT  19 H VWFAKKFTEN  27 H EMLQGFAVAV

   N VTAHGQAVQA     N IVSHKLHVRG     N RLYASYTIRL     N LGGFVIAbTS
37 H VAGHGQDILI  27 H IVGWAHDVRG  30 H KVYNAIALKL  24 H LNKFLANDST

   N TAHGQAVQAA     N QTFRFIWFRD     N HCELSTELAV     N WSIYSAIFEI
31 H KAHGKKVLGA  20 H AEFAFSLYRQ  18 H HCLLVTLAAH  21 H WTIAYDELAI

   N AHGQAVQAAA     N FTILKDVTLN     N KYAWVAIRYT     N SIYSAIFEII
36 H AHGQKVANAL  39 H FPVVKEAILK  23 H KYLSIVKEYA  30 H AINEAISDIF

   N GNVFTDSVTV     N QYSFNSVVLD     N AWVAIRYTYL     N IYSAIFEIIT
24 H GNVLVTVLAI  28 H EFGYSERVVD  19 H VWFAKKFTEN  30 H IYEDCMDLIA

   N VTAHGQAVQA     N AVGELFLARS     N RLYASYTIRL     N YSAIFEIITA
37 H VAGHGQDILI  22 H NVLALVVARN  30 H KVYNAIALKL  30 H YTVLFGVSRA

   N TAHGQAVQAA     N VGELFLARSV     N QGKVIQPVFV     N SAIFEIITAL
31 H KAHGKKVLGA  18 H VLALVVARNF  25 H AGKVFKLVYE  30 H EAISDIFGTL

   N AHGQAVQAAQ     N GELFLARSVT     N GKVIQPVFVV     N AIFEIITALG
38 H AHGQKVANAL  18 H EELNSAWTIA  27 H GKVFKLVYEA  32 H VLFGVSRALG

   N HGQAVQAAQQ     N ELFLARSVTL     N LLNTNIDAGE     N IFEIITALGN
29 H HGQKVANALT  26 H NLVLATAAKL  32 H LFNQDVDAAV  27 H VFQETKAIAD

   N GQAVQAAQQI     N AVGELFLARS     N PDEAAVGNLV     N EIITALGNAE
30 H AAAVQSATDL  22 H NVLALVVARN  25 H PHVAGAAALI  36 H KVLTSLGDAI

   N LAVIGVLMKV     N VGELFLARSV     N EAAVGNLVGF     N DDSRVIAHTK
20 H VVTIEGIIKI  18 H VLALVVARNF  31 H ETHANRIVGF  23 H AKNWLVNHIK

   N SRVIAHTKLI     N LEADGGAVKK     N IYAIDADSCI     N AQQSETISHQ
29 H PKVKAHGKKV  28 H LQAHAGKVFK  20 H LFNQDVDAAV  29 H QQTSEAVNMQ

   N RVIAHTKLIG     N LEFRDKANAK     N LDWGAMNAKV     N QQSETISHQK
27 H QVKAHGKKVA  27 H KKFKKKGAAK  19 H NESGAINEAI  26 H ERADLISYLK

   N VIAHTKLIGS     N EFRDKANAKD     N NQFGHQENAK     N VRQQEGESRL
25 H VKAHGKKVAD  32 H KFKKKGAAKA  26 H NEFGYSERVV  19 H LRKSEAQAKK

   N GEGEPEELMV     N FRDKANAKDI     N SDDATALMTD     N QEGESRLNLL
19 H VAGHGQDILI  30 H FKKKGAAKAK  26 H SERVVDLMAH  23 H QEAARAGELL

   N EELMVDNWRP     N DANKSKGIVW     N PKFITWSPVC     N EGESRLNLLQ
17 H SAEFLEGWKA  37 H QSKRSADILW  26 H PKKVMWNTAV  24 H EGEWQLVLHV

   N ELMVDNWRPA     N AKFTQFAGKD     N VLDAFTQGLK     N GESRLNLLQR
```

```
19 H QLVLHVWAKV  31 H WKFADKAGKD  31 H QRNGFIQSLK  25 H DYTQMNDLQR

   N LFLQNFKADA     N AAHaVDRELT     N LDAFTQGLKH     N ESRLNLLQRN
26 H LFNQDVDAAV  37 H TAETIARQLA  35 H RNGFIQSLKD  24 H EAIIHVLHSK

   N FLQNFKADAR     N TDDVAAGYDI     N DAFTQGLKHL     N NLLQRNANVF
29 H FLEGWKALAT  33 H ADDVKKAFAI  29 H DKFLEDVKKL  27 H ELQAHAGKVF

   N LQNFKADARA     N SVTLNSYVQL     N AFTQGLKHLD     N LLQRNANVFI
27 H LELFRNDIAA  22 H GFALNLYVKH  34 H VFSQVCTHLD  32 H ALAKNVAAFI

   N QNFKADARAL     N LNSYVQLGVL     N LKHLDDLKGA     N LQRNANVFIF
24 H ENFKLLGNVL  24 H KGTFAQLSEL  22 H LKNLGSVHVS  37 H LAKNVAAFII

   N NFKADARALT     N YVQLGVLPRA     N HLDDLKGAFA     N QRNANVFIFI
23 H IFKCQGAALN  31 H YTQMNDLQRR  38 H QLNNFRAGFV  30 H AKNVAAFIIL

   N HTKHATVECV     N VLPRAGTILA     N LDDLKGAFAQ     N YEVIKLKGYE
16 H QLRAAAVQSA  28 H FLASVSTVLT  38 H PEELKGIFEK  26 H YNAIALKLRE

   N TKHATVECVQ     N LTRTNGQLAQ     N PEEQHCADCQ     N EVIKLKGYEN
14 H KKAAKAVCKH  36 H VSRALGVLAQ  26 H AFHTQCIDCH  26 H EVAQLKNSAD

   N HATVECVQCH     N SQSANLLAEA     N EEQHCADCQF     N IKLKGYENWI
31 H AFHTQCIDCH  31 H TESTKLAAAA  23 H ELRRCTGKHF  20 H IIPTAQETWL

   N ATVECVQCHH     N NIIQGSIYAI     N IAHVAQQSET     N AdHLScSALL
31 H FHTQCIDCHA  31 H KMLQGRLFAY  22 H VAQLKNSADT  17 H TQCIDCHALK

   N ECVQCHHTLE     N IIQGSIYAID     N VAQQSETISH     N bQNRDVRQYV
31 H QCIDCHALKK  27 H IINKAAYLIS  30 H VAESAETVMK  32 H EAKKQINDYV

   N QNRDVRQYVQ     N GEDNINVVEG     N VIAVDDQFVN     N NFKEELEKKG
31 H AKKQINDYVE  34 H GEKTLNAIDS  27 H VLTIMEHTVN  19 H ELKAAIGKMS

   N NRDVRQYVQG     N EDNINVVEGN     N DDQFVNAKGK     N KEELEKKGLG
28 H KKQINDYVEK  32 H ADQISTVQAS  31 H DDELKAAIGK  34 H AEALERMFLG

   N DEAAVNLAKS     N ASLNSRVASI     N MDTVPLVMMA     N ELEKKGLGGK
33 H DDEAQTLAKW  34 H ASLDKFLASV  22 H RDDLNLVLAT  28 H ELFRKDIAAK

   N LdNIPcSALL     N TIQLLbLVSG     N DTVPLVMMAA     N NLGHVALTFG
19 H IAKLPCVAAK  24 H TQTLLDLFTI  30 H DDLNLVLATA  16 H SLGELIHTLD

   N KGTDVQAWIR     N AVVGTFAARV     N GQGFKAADII     N RQKGAVTPVK
17 H QAQEVHEKLR  25 H ADVNTFVASH  37 H GQSKRSADIL  27 H VAKSAVAALK

   N TDVQAWIRGa     N QLTTGNGLFL     N KFTEVTDVVL     N EGVQRYbRSR
29 H QDVDAAVRGI  23 H HVAGAAALIL  25 H KFLASVSTVL  22 H NAIRNYAISK

   N DVQAWIRGaR     N FALVNYIFFK     N FTEVTDVVLI     N VQRYbRSREK
35 H DIVGWAHDVR  21 H FPVVKEAILK  28 H FPVVKEAILK  18 H IQTANIALEK

   N KNLDSfKFLV     N TVKVPMMKRL     N VTDVVLITRD     N EKDGKPVSAF
25 H KTLNAIDSLA  24 H DAQAAMKKAL  23 H VKEAILKTIK  20 H ETHANRIVGF

   N NLDSfKFLVD     N VKVPMMKRLG     N DVVLITRDNF     N VSAFHDIPLY
34 H ITDGFQFIIN  26 H VKAHGKKVLG  25 H IGRLVTRAAF  16 H IGERQGQELY

   N LDSfKFLVDN     N LSKAVHKAVL     N VVLITRDNFK     N ADKEDNIFNM
30 H QNGFKFLEPI  33 H LKAAVDKAVA  32 H GRLVTRAAFN  27 H KDEAEKLFNQ

   N DKASPYEVML     N SKAVHKAVLT     N VLITRDNFKE     N DKEDNIFNMV
19 H DFSKAFEKLL  34 H WKEVHKMVVE  24 H VLATAAKLKA  28 H ATLDTFFGMI

   N PAIFKATLNR     N KAVHKAVLTI     N LITRDNFKEE     N KEDNIFNMVV
21 H QELYKEMQKR  34 H KEVHKMVVES  22 H IVARSNFNVc  26 H NVQALFQKVV

   N IFKATLNRSL     N TEAAGAMFLE     N ITRDNFKEEL     N EDNIFNMVVE
24 H ALKAAVDKAV  36 H AEALERMFLS  24 H EQRNGFIQSL  33 H ISDIFGTLVE

   N FMDFLTENGF     N DMKVIAVDDQ     N TRDNFKEELE     N DNIFNMVVEI
```

31 H YLEFISEAII 28 H ERIINAVDDA 33 H AHDDFIHKLD 42 H DDLYNMINTV

N LVKKSDGaKY N MKVIAVDDQF N RDNFKEELEK N NIFNMVVEIP
36 H LFRKDIAAKY 26 H MWNTAVHSEF 28 H RNGFIQSLKD 37 H EVHKMVVESA

N DGaKYGbLKL N KVIAVDDQFV N DNFKEELEKK N IFNMVVEIPR
23 H QGAMNKALEL 20 H RIINAVDDAV 29 H DKFLEDVKKL 26 H VFKLVYEAAI

N FNMVVEIPRW N KLNDIEDVEK N AALNNSIGVL N GKLRFVRNCF
17 H YTQMNDLQRR 24 H TLNAIDSLAA 31 H AQLKNSADTL 16 H VNFKLLSHCL

N WTNAKLEITK N LNDIEDVEKY N ALNNSIGVLG N KLRFVRNCFP
22 H WKALATESTK 33 H DDEIENVIAY 31 H GVSRALGVLA 15 H NFKLLSHCLL

N TNAKLEITKE N NDIEDVEKYF N IEWAIANNMD N IAYTGQVKEV
28 H KKYALDIIKE 42 H DEIENVIAYL 29 H TEWKFADKAG 23 H FGYSERVVDL

N NAKLEITKEE N EDVEKYFPGL N LKGILGYTED N PKLNDIEDVE
26 H KYALDIIKET 40 H EDVQKFRHEL 26 H VKSIVDFVKN 24 H KTLNAIDSLA

N AKLEITKEET N KGIDLTNVTL N QGKIVDLVKE N QTDDKGHIIV
25 H TAKELAEEET 34 H RAIQTANIAL 30 H VKSIVDFVKN 20 H NTEKMAELIA

N LEITKEETLN N IDKSIDKWFF N GKIVDLVKEL N VDEFQNTNVK
26 H FPVVKEAILK 24 H ITQATGVWFA 30 H ERVVDLMAHM 20 H VAESAETVMK

N TKEETLNPII N RSAGAVVDIK N KIVDLVKELD N VGDVCGKALL
28 H VKEAILKTIK 28 H RTTGSMDALK 31 H RVVDLMAHMA 32 H VGMACAISIL

N KEETLNPIIQ N NMDKKDKSAK N IVDLVKELDR N EDSKLDYNNI
33 H KEAILKTIKE 32 H SREKMNETAK 27 H RaELARTLKR 26 H EDFQKVYNAI

N QNTKGKLRFV N MDKKDKSAKG N VDLVKELDRD
32 H DVAKGKKTFV 28 H REKMNETAKE 22 H LDIIKETHNS

N NTKGKLRFVR N IRHDNVLRSF N DLVKELDRDT
32 H VAKGKKTFVQ 26 H AHGKKVLHSF 22 H ETVLDMLRDA

N TKGKLRFVRN N KDLSLNKLGI N KWERPFEVKD
38 H AKGKKTFVQK 22 H KALELFRKDI 37 H KWMRDFEERM

N KGKLRFVRNC N LNKLGIQTDD N SGVTEEAPLK
15 H SARLRNVMAI 22 H LDLFTIQQTE 22 H FPVVKEAILK

APPENDIX D

Table showing all documented network runs in each section and where appropriate the number of hidden nodes used.

SECTION	NUMBER OF DOCUMENTED RUNS	NUMBER OF HIDDEN NODES
5	3	-
6.1.1	5	50
6.1.2	1	40
6.2	8	25,50,100
6.4	15	0,10,20,40,40+8
6.5	1	20
6.6	22	0,5,10,15,20,30,40
6.7	6	0,3,10
6.9	26	5,7,15
7.1	15	0,5
7.2	54	0,5,20
7.3	1	0
7.4.1	5	5
7.4.2	3	5
7.4.3	1	5
7.6	1	5
7.7	4	5
7.8.2	13	20,40,50,100
7.9	63	0,5,20
7.9.2	1	5
7.10.1	3	5,10