Reprinted from *J. Mol. Biol.* (1992) **225**, 1049–1063

Hybrid System for Protein Secondary Structure Prediction

Xiru Zhang, Jill P. Mesirov and David L. Waltz

J. Mol. Biol. (1992) 225. 1049-1063

Hybrid System for Protein Secondary Structure Prediction

Xiru Zhang, Jill P. Mesirov and David L. Waltz

Thinking Machines Corporation
245 First Street
Cambridge, MA 02142, U.S.A.

(Received 11 October 1991; accepted 10 February 1992)

We have developed a hybrid system to predict the secondary structures (α-helix, β-sheet and coil) of proteins and achieved 66·4% accuracy, with correlation coefficients of C_{coil} = 0·429, C_α = 0·470 and C_β = 0·387. This system contains three subsystems ("experts"): a neural network module, a statistical module and a memory-based reasoning module. First, the three experts independently learn the mapping between amino acid sequences and secondary structures from the known protein structures, then a Combiner learns to combine automatically the outputs of the experts to make final predictions. The hybrid system was tested with 107 protein structures through k-way cross-validation. Its performance was better than each expert and all previously reported methods with greater than 0·99 statistical significance. It was observed that for 20% of the residues, all three experts produced the same but wrong predictions. This may suggest an upper bound on the accuracy of secondary structure predictions based on local information from the currently available protein structures, and indicate places where non-local interactions may play a dominant role in conformation. For 64% of the residues, at least two experts were the same and correct, which shows that the Combiner performed better than majority vote. For 77% of the residues, at least one expert was correct, thus there may still be room for improvement in this hybrid approach. Rigorous evaluation procedures were used in testing the hybrid system, and statistical significance measures were developed in analyzing the differences among different methods. When measured in terms of the number of secondary structures (rather than the number of residues) that were predicted correctly, the prediction produced by the hybrid system was also better than those of individual experts.

Keywords: protein secondary structure prediction; hybrid system; neural networks; memory-based reasoning; statistical methods

1. Introduction

Determining the mapping between amino acid sequences and secondary structures (α helix, β sheet, etc.) is an important step towards our understanding of how protein sequences specify their overall structures and functions. Currently the main technique to determine protein structures is X-ray crystallography, which is a slow and often difficult process. On the other hand, the database of known protein sequences is growing very rapidly. Thus, it is increasingly important to develop computational approaches to determine automatically (predict) the structures of proteins whose sequences are known. The correct prediction of secondary structures can contribute significantly towards this goal. For example, the knowledge of secondary structures can provide a good starting point and reduce the search space in simulation of protein folding by molecular dynamics (Levitt, 1983) or lattice models (Skolnick & Kolinski, 1990), or can be used in predicting higher order structures (e.g. super secondary structures (Taylor & Thornton, 1984), domains (Lathrop *et al.*, 1987)).

Many algorithms have been developed for protein secondary structure prediction. One of the first efforts was made by Chou & Fasman (1974). Different implementations of their algorithm have all attained about a 50 to 60% level of accuracy in predicting the location of α helices, β strands and "coil" (i.e. anything other than helix or strand) in a protein sequence. Garnier, Osguthorpe & Robson's algorithm (Garnier *et al.*, 1978) is about 58% accurate for this task. More recently, their improved algorithm (Gibrat *et al.*, 1987) is 63% accurate. Qian & Sejnowski (1988) used an artificial neural network algorithm to increase the prediction accuracy to 64%. Similar results have also been achieved by other researchers (e.g. Kneller *et al.*, 1990; Holley & Karplus, 1989). Thus there has been about a 6% improvement of prediction accuracy in

0022-2836/92/121049-15 $03.00/0

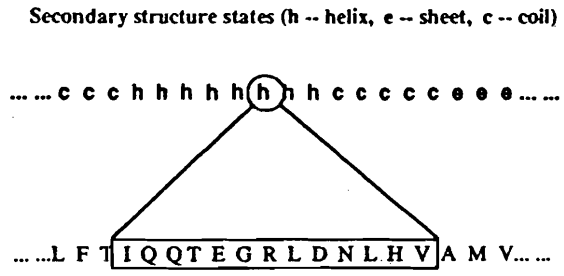

Figure 1. A window is moved along an amino acid sequence to extract correlations between the residues and the secondary structure state of the center residue.

the last 15 to 20 years, which is due to both the improved computational methods and the increase of the known protein structure data. Almost all these algorithms have adopted a "local strategy": moving a "window" (typically covering 7 to 19 residues) along an amino acid sequence and predict the secondary structure state of the center residue in the window according to all the residues inside the window (see Fig. 1). To assess the accuracy of a prediction algorithm for proteins whose structures are not known, it is a common practice to divide the known protein structure database into two separate sets: the "training data set" is used to set the parameters of the algorithm, and the "test data set" is used to test its prediction accuracy. The predictions produced by the existing algorithms, though imperfect, can often show the likelihood or tendency of certain peptide chains to form particular secondary structures. It is also important to know the extent to which the protein structures are determined by "local interactions": interactions among residues adjacent along the polypeptide chain.

Though existing prediction algorithms are all about 60 to 64% accurate for three-state (α-helix, β-sheet, and coil) prediction, they can make incorrect predictions at different places of an amino acid sequence. From the point of view of machine learning (artificial intelligence), secondary structure prediction is an instance of *inductive learning*, generalizing from known examples to solve new problems. Different algorithms may work according to different principles and can generalize in different ways. Therefore, a combination of different algorithms can potentially produce a better prediction than individual ones. Based on this analysis, we developed a hybrid system to predict the secondary structures, which indeed improved the prediction accuracy significantly. Our hybrid system has three different modules ("experts"): a neural network module, a statistical module and a memory-based reasoning module, and a Combiner. The experts were chosen in such a way that they have different mathematical properties. In the training phase, the experts independently learn the mapping between amino acid sequences and secondary structures from the known protein structures; the Combiner learns to combine automatically the outputs of the experts. In the prediction phase, the three experts make predictions separately, then the Combiner takes the predictions from the three experts and makes final predictions. K-way cross-validation was used in evaluating the hybrid system and statistical significance measures were used in comparing different prediction algorithms.

Our experiments showed that (1) the hybrid system had an overall prediction accuracy of 66·4%, which was higher than individual experts and all previously reported algorithms at greater than 0·99 confidence level; (2) the three experts not only had very close overall prediction accuracy, their detailed predictions also agreed with one another much more than with the real structure (i.e. their prediction accuracy); (3) the accuracy of prediction algorithms could change as the test data changes, especially when the test data set was small (e.g. containing 15 protein sequences); (4) for 20% of the residues, all three very different experts produced the same but wrong prediction, suggesting that with the currently available protein structure data, 80% may be the upper bound for the secondary structure prediction accuracy using the local strategy; (5) compared to each expert, the hybrid system also produced better result in terms of the number of secondary structures (rather than the number of residues) that were predicted correctly.

2. Methods and Materials

(a) *The architecture and training of a hybrid system*

Figure 2 shows the overall architecture of our hybrid system. The system contains three "experts", a statistical module, a memory-based reasoning module and a neural network module, and a Combiner. The whole system produces secondary structure predictions as follows: given a set of amino acid sequences (i.e. test data), each expert makes its predictions independently, then the Combiner takes the predictions from the 3 experts and combines them to produce final predictions. At the beginning, the hybrid system learns from the training data set about mappings between amino acid sequences and secondary structures. The training of the whole system involves (1) training the 3 experts and (2) training the Combiner. How each expert is trained and how each makes predictions are discussed in the following sections. In order to train the Combiner, half of the training data is used to train the 3 experts separately, and the outputs of these trained experts on the second half of the training data are recorded. These outputs are then used as inputs to train the Combiner. The reason for dividing the training data set into 2 parts is that the behavior of each expert on training data can be very different from its behavior on the proteins whose structures are unknown: their performance on the data that they are not trained on (the second half of the training set) reflects their behaviors on truly unknown protein structures, which is exactly what the Combiner should know about and be trained on. The training of the experts with half of the training data is done purely for the purpose of training the Combiner. After the training of the Combiner is completed, each

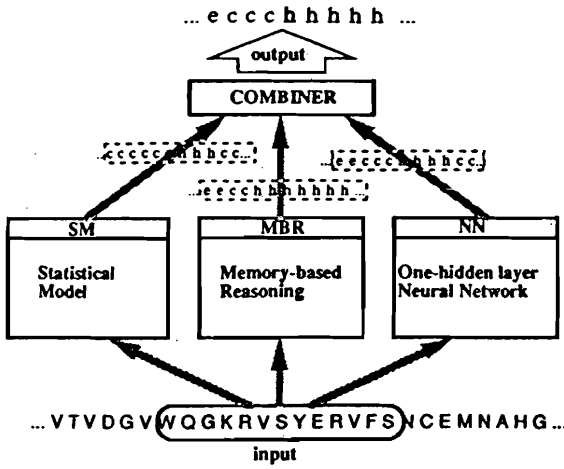

Figure 2. The hybrid system has 3 experts. a statistical module, a memory-based reasoning module and a neural network module. The Combiner combines the outputs of the 3 experts to produce a final output.

expert is trained again with the whole training data set. These trained experts together with the trained Combiner form a trained hybrid system.

(b) *Memory-based reasoning*

Memory-based reasoning (MBR†) (Stanfill & Waltz, 1986) is one expert in our hybrid system. The essential idea of MBR is to use known examples directly in problem solving. For predicting the protein secondary structures, this involves matching each segment (window) of amino acid sequences in the test data set against all the sequences in the training set, finding its "nearest neighbors", and choosing the secondary structure state of the majority of its neighbors as the prediction. Similar approaches have been referred to as the "nearest neighbor method", "exemplar-based reasoning", etc. Levin *et al.* (1986) and Nishikawa & Ooi (1986) called this approach the "homologous method". The key component in this approach is the distance function or metric used to compute the neighbors. The choice of a metric is especially difficult for elements such as amino acids, because there is no linear ordering among the elements, which are often referred to as having "nominal values". Stanfill & Waltz (1986) proposed several distance functions for nominal values in their work on memory-based reasoning. We improved their functions and applied them to protein secondary structures in this work.

Based on the idea of MBR, one distance matrix is computed for each position of the window using the training data set. At window position i, the distance matrix D_i contains the distance between every pair of amino acids at that position. The distance between 2 segments of amino acid sequences $A = a_1 a_2 \ldots a_n$ and $B = b_1 b_2 \ldots b_n$ is defined as:

$$D(A, B) = \sum_{i=1}^{n} D_i(a_i, b_i),$$

where n is the window size, $D_i(a_i, b_i)$ is the distance between amino acids a_i and b_i at position i. The smaller this distance is, the more similar a_i and b_i are in terms of forming secondary structures, and the less effect it has on secondary structures if one is replaced by the other. The distance matrices D_i can be computed from the training data. Assuming there are m secondary structure states $s_1, s_2, \ldots s_m$ and q different amino acids, $x^1, x^2, \ldots x^q$ ($a_i\, b_i \in \{x^1, \ldots x^q\}$), $D_i(a_1, b_i)$ is computed as:

$$D_i(a_i, b_i) = \frac{1}{m}\sum_{j=1}^{m} |p(s_j|a_i) - p(s_j|b_i)| + \frac{1}{m \cdot n \cdot q}\sum_{j=1}^{m}\sum_{k=1}^{n}\sum_{h=1}^{q} |p(s_j|a_i, x_k^h) - p(s_j|b_i, x_k^h)|. \quad (1)$$

where x_k^h denotes amino acid x^h at window position k; $p(s_j|a_i)$ is the conditional probability of secondary structure state s_j given that a_i has occurred: it represents the influence on secondary structure s_j by the singleton amino acid at position i. $p(s_j|a_i, x_k^h)$ is the conditional probability of s_j given both a_i and x_k^h have occurred: it represents the influence on s_j by a_i together with its neighbor amino acids. Thus when $p(s_j|a_i) \approx p(s_j|b_i)$ and $p(s_j|a_i, x_k^h) \approx p(s_j|b_i, x_k^h)$, a_i and b_i are similar in determining secondary structures, and $D_i(a_i, b_i)$ should be small, which is exactly what equation (1) yields.

(c) *A statistical method*

A statistical module (SM) is the second expert in our hybrid system. It works as follows: for each secondary structure state s_j, if the conditional probability of s_j given a window of n residues $a_1 \ldots a_n$, $p(s_j|a_1 \ldots a_n)$, is known, then the s_j that has the highest value for this conditional probability is chosen as the prediction for $a_1 \ldots a_n$:

$$\text{Prediction} = \left\{ s_j | \max_j p(s_j|a_1, a_2, \ldots a_n) \right\},$$

$$s_j \in \{\alpha\text{-helix}, \beta\text{-sheet}, \text{coil}\}.$$

According to Bayes Theorem:

$$p(s_j|a_1 \ldots a_n) = \frac{p(s_j) \cdot p(a_1 \ldots a_n|s_j)}{p(a_1 \ldots a_n)}. \quad (2)$$

where $p(s_j)$ is the probability of s_j and $p(a_1 \ldots a_n|s_j)$ is the probability of $a_1 \ldots a_n$ in secondary structure state s_j; $p(a_1 \ldots a_n)$ is the probability of $a_1 \ldots a_n$ in all states. Since we only want to find the largest $p(s_j|a_1 \ldots a_n)$, $p(a_1 \ldots a_n)$ need not be computed. Currently there is not enough protein structure data available for us to compute the frequencies of $a_1 \ldots a_n$ in each state s_j in order to estimate $p(a_1 \ldots a_n|s_j)$. They have to be estimated by some simpler terms. We extend and apply the Bahadur-Lazarsfeld expansion (Bahadur, 1961) here (which only deals with binary variables in its original form). Assuming that $y_1, y_2, \ldots y_n$ are random variables with nominal values, then

$$p(y_1, \ldots, y_n) = \prod_i^n p(y_i) \times \left\{ 1 + \sum_{i<k} Z_{ik} + \sum_{i<k<h} Z_{ikh} + \ldots \right\}, \quad (3)$$

where Z_{ik} is the second order correlation between y_i and y_k:

$$Z_{ik} = \frac{p(y_i, y_k)}{p(y_i)p(y_k)} - 1,$$

† Abbreviations used: MBR, memory-based reasoning; IP, input pattern; SM, statistical module.

and Z_{ikh} is the third order correlation among y_i, y_k and y_h:

$$Z_{ikh} = \left(\frac{p(y_i, y_k, y_h)}{p(y_i)p(y_k)p(y_h)} - 1\right) - \left(\frac{p(y_i, y_k)}{p(y_i)p(y_k)} - 1\right) - \left(\frac{p(y_i, y_h)}{p(y_i)p(y_h)} - 1\right) - \left(\frac{p(y_k, y_h)}{p(y_k)p(y_h)} - 1\right).$$

and so on.

In practice, for the secondary structure prediction problem, we can only estimate up to the second order correlations with the currently available protein structure data. The reliability of these estimates depends on the sample size used. Thus, we postulate the following equation:

$$p(a_1, \ldots, a_n|s_j) \approx \prod_i p(a_i|s_j) \cdot \left[1 + C_f \cdot \sum_{i<k} f_{ik} \cdot \left(\frac{p(a_i, a_k|s_j)}{p(a_i|s_j)p(a_k|s_j)} - 1\right)\right] \quad (4)$$

where f_{ik} is proportional to the size of the sample in which ratio

$$\frac{p(a_i, a_k|s_j)}{p(a_i|s_j)p(a_k|s_j)}$$

is computed, to represent its reliability:

$$f_{ik} = \sqrt{\frac{p(a_i|s_j)p(a_k|s_j)}{(1-p(a_i|s_j))(1-p(a_k|s_j))}}.$$

Some observations about equation (4): (1) Compared with equation (3), correlations among 3 or more residues are ignored. This is due to the limited sample size. This truncation may have an overall positive or negative effect on the contribution from higher-order correlations in the approximation, thus coefficient C_f is introduced to compensate for this. C_f can be experimentally determined. (2) When there are no higher-order correlations among the residues in a window (i.e. they are all independent), $p(a_1, \ldots, a_n|s_j)$ is reduced to $\prod_i p(a_i|s_j)$, which is correct. (3) Information of all C_n^2 possible pairs of residues in a window of size n is used here, whereas in a commonly used statistical method, the GOR III method, only $n-1$ pairs are used. (4) If the pairwise correlation terms are small and the approximation $\log(1+x) \approx x$ is used, we get the following equation:

$$p(a_i, \ldots, a_n|s_j) \approx \prod_i p(a_i|s_j) \cdot e^{C_f \cdot \sum_{i<k} f_{ik} \cdot \left(\frac{p(a_i, a_k|s_j)}{p(a_i|s_j)p(a_k|s_j)} - 1\right)}. \quad (5)$$

This is exactly the form in Lazarsfeld's original expansion (Lazarsfeld, 1961), which he derived from a completely different path. One advantage of equation (5) is that it guarantees that the probability approximation is non-negative, which equation (4) does not do. Equation (5) is the final form of the statistical expert used in this work.

(d) *Artificial neural network*

Artificial neural networks have been used widely in many applications (McClelland & Rumelhart, 1986), including protein secondary structure prediction (Qian & Sejnowski, 1988; Kneller *et al.*, 1990). An artificial neural network usually consists of a large number of simple processing units connected by weighted links. Each unit computes its output by applying an "activation function" to its inputs. The training algorithm used in this work, the Back-propagation algorithm (Rumelhart *et al.*, 1986), works on a particular kind of artificial neural network, a layered, feed-forward network (see Fig. 3), where the processing units are arranged in layers: there is an input layer, an output layer, and one or more "hidden layers" (layers between the input and output layer). A feed-forward network computes its output in the following fashion: first, the input layer is set according to an input pattern; then one layer at a time, from the input to hidden to output layer, the units compute their outputs by applying an activation function to the weighted sum of the outputs from the units at the lower layer. The weights come from the links between the units. The "sigmoid function" is often used in feedforward networks as the unit's activation function:

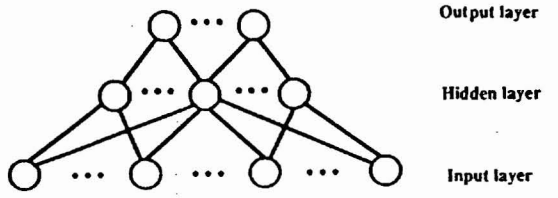

Figure 3. A one-hidden-layer feedforward artificial neural network. The network computes its output based on the values of the units at the input layer.

$$O_{i,j} = \frac{1}{1+e^{-x}}$$

Where O_{ij} is the output of unit j at layer i, and x is the weighted sum of outputs from units at one layer below:

$$x = \sum_k u_{i-1,k}^{i,j} O_{i-1,k}$$

$u_{k,i-1}^{i,j}$ is the weight of the link from unit k at layer $i-1$ to unit j at layer i. This can also be seen as a projection of the network input to a certain direction specified by the weights. Thus, each hidden unit represents a different projection from the multiple dimensional input space to a new space whose dimensionality is determined by the number of hidden units in the network.

The Back-propagation algorithm "trains" a layered network by adjusting the link weights of the net using a set of "training examples". Each training example consists of an input pattern and an ideal output pattern that the user wants the network to produce for that input. The weights are adjusted based on the difference between the ideal output and the actual output of the network. This can be seen as a gradient descent process in the weight space. An "epoch training cycle" consists of presenting all training examples once to the network, and then adjusting the weights on the basis of the accumulated errors at the output layer. A number of epoch cycles may be required before the output errors are reduced to an accepted level. After the training is completed, the network can be applied to inputs that are not in the set of training examples. For a new input pattern IP, the trained network tends to produce an output similar to the training example whose input is similar to IP. This can be used for interpolation, approximation, or generalization from examples depending on the goal of the user.

Table 1
Protein structures used in this work

Protein	Code	Subunit	Length	No. H	No. E
Cytochrome c550	155C		134	35	5
Cytochrome B562 (*E. coli*, oxidized)	156B		110	67	0
L-Arabinose-binding protein	1ABP		306	106	20
Actinoxanthin	1ACX		107	0	47
Phospholipase A2	1BP2		123	54	8
Cytochrome c5 (oxidized)	1CC5		83	39	0
Cytochrome c	1CCR		111	44	0
Calcium-binding parvalbumin B	1CPV		108	52	6
Crambin	1CRN		46	20	4
Subtilisin carlsberg (inhibitor)	1CSE		63	11	22
17/112 Ribosomal protein (C-terminal domain)	1CTF		68	35	18
Cytochrome c3	1CY3		118	16	0
Hemoglobin (erythrocruorin, deoxy)	1ECD		136	97	0
Elongation factor tu (domain i)	1ETU		196	78	36
Immunoglobin FAB	1FB4	(H L)	445	11	208
FC fragment (IGG1 class)	1FC1	(A)	206	15	95
Immunoglobin fc and fragment B of protein a complex	1FC2	(C)	43	21	0
Ferrodoxin	1FDX		54	5	4
Flavodoxin	1FX1		147	43	32
Ferredoxin	1FXB		81	10	0
Glucagon (pH 6–pH 7 form)	1GCN		29	14	0
γ Crystallin	1GCR		174	5	77
Glyceraldehyde-3-phosphate dehydrogenase	1GD1	(O)	336	73	95
Glutathione peroxidase	1GP1	(A)	184	43	29
Oxidized high potential iron protein (HIPIP)	1HIP		85	10	9
Hemerythrin (MET)	1HMQ	(A)	113	73	0
Insulin	1INS	(A D)	51	22	3
Leghemoglobin (Acetate, MET)	1LH1		153	107	0
Lysozyme	1LZ1		130	39	10
Myoglobin (DEOXY, pH 8·4)	1MBD		153	113	0
Immunoglobulin FAB fragment (MC/PC603)	1MCP	(H L)	442	8	211
Melittin	1MLT	(A)	26	22	0
Neurotoxin B	1NXB		62	0	26
Pseudoazurin	1PAZ		120	17	44
Plastocyanin	1PCY		99	4	35
Hydroxybenzoate hydroxylase	1PHH		394	119	96
Calcium-free phospholipase A2	1PP2	(L)	133	48	8
Avian pancreatic polypeptide	1PPT		36	18	0
Rhodanese	1RHD		293	81	32
Ribonuclease A	1RN3		124	22	48
Ribonuclease T1 isozyme	1RNT		104	17	28
Subtilisin BPN	1SBT		275	83	49
Trypsin (SGT)	1SGT		240	21	77
Scorpion neurotoxin (variant 3)	1SN3		65	8	12
Trypsinogen complex with porcine pancreatic secretory	1TGS	(I)	57	9	11
Triose phosphate isomerase	1TIM	(A)	248	106	42
Tonin	1TON		238	10	71
Ubiquitin	1UBQ		76	12	24
α-Bungarotoxin	2ABX	(A)	74	0	4
Actinidin (sulfhydryl proteinase)	2ACT		218	56	40
Acid proteinase, penicillopepsin	2APP		323	30	147
Acid proteinase (rhizopuspepsin)	2APR		325	26	146
Azurin (oxidized)	2AZA	(B)	129	13	41
Cytochrome B5 (oxidized)	2B5C		85	21	21
Carbonic anhydrase form B (carbonate dehydratase)	2CAB		256	17	79
Cytochrome c′	2CCY	(A)	127	90	0
Cytochrome c3	2CDV		107	27	10
Chymotrypsinogen A	2CGA	(A)	245	18	79
Chymotrypsin inhibitor 2 (CI-2)	2CI2		65	11	14
Concanavalin A	2CNA		237	4	103
Cytochrome P450CAM (camphor monooxygenase)	2CPP		405	180	41
Citrate synthase	2CTS		437	257	6
Cytochrome c peroxidase	2CYP		293	134	16
Gene 5 DNA binding protein	2GN5		87	0	4
Hemoglobin (deoxy)	2HHB	(A B)	287	197	0
Hemoglobin V (CYANO.MET)	2LHB		149	100	0
Lysozyme	2LZM		164	109	14
Cytoplasmic malate dehydrogenase	2MDH	(A B)	649	213	110

Table 1 *(continued)*

Protein	Code	Subunit	Length	No. H	No. E
CD. ZN Metallothionein (isoform II)	2MT2		61	0	0
Ovomucoid third domain	2OVO		56	10	9
Prealbumin (human plasma)	2PAB	(A)	114	8	59
Proteinase K	2PRK		279	66	60
Staphylococcal nuclease complex	2SNS		141	26	28
CU.ZN Superoxide dismutase	2SOD	(B)	151	0	54
Streptomyces subtilisin inhibitor	2SSI		107	17	26
Satellite tobacco necrosis virus	2STV		184	18	82
Tomato bushy stunt virus	2TBV	(C)	321	4	112
Cytochrome c551 (oxidized)	351C		82	38	0
Adenylate kinase	3ADK		194	106	25
Bacteriochlorophyll	3BCL		356	57	170
Cytochrome c2 (reduced)	3C2C		112	44	0
Native elastase	3EST		251	13	82
Ferredoxin	3FXC		98	7	15
Catabolite gene activator protein-cyclic AMP complex	3GAP	(A)	208	64	21
Glutathione reductase, oxidized form (E)	3GRS		461	132	111
Calcium-binding protein	3ICB		75	43	0
Phosphoglycerate kinase complex with ATP	3PGK		415	143	46
Phosphoglycerate mutase DE-phospho enzyme	3PGM		230	69	15
Rat mast cell protease II	3RP2	(A)	237	12	83
Rubredoxin	3RXN		52	0	8
Wheat germ agglutinin (isolectin 2)	3WGA	(B)	171	16	16
TRP aporepressor	3WRP		101	77	0
APO-liver alcohol dehydrogenase	4ADH		374	79	77
Aspartate carbamoyltransferase	4ATC	(A B)	463	133	65
Carboxypeptidase Aα (COX) complex	4CPA	(I)	37	0	6
Dihydrofolate reductase complex	4DFR	(B)	159	29	56
Ferredoxin	4FD1		106	18	14
Flavodoxin (semiquinone form)	4FXN		138	47	29
Lactate dehydrogenase APO enzyme M4	4LDH		333	111	37
Trypsin inhibitor	4PTI		58	8	14
β Trypsin, diisopropylphosphoryl inhibited	4PTP		234	16	72
Southern bean mosaic virus coat protein	4SBV	(C)	222	32	72
Thermolysin complex	4TLN		316	117	54
Troponin C	4TNC		160	101	6
Carboxypeptidase Aα (COX)	5CPA		307	111	50
Catalase	7CAT	(A)	498	137	71
Papain CYS-25 oxidized	9PAP		212	49	36
Total		113	19.861	5324	4098

If there is more than 1 subunit in a protein, column Subunit indicates which subunit(s) was used. Length indicates the number of residues in the protein sequences used. No. H indicates the number of residues in α-helix; No. E indicates the number of residues in β-sheet. There are 107 proteins in this Table, with 113 subunits, 19,861 residues.

(e) *Database*

A database of 107 proteins was selected from Brookhaven Protein Data Bank. It contains 19,861 residues, 113 subunits. All sequences (subunits) are less than 50% homologous with one another. The DSSP program (Kabsch & Sander, 1983*a*) was used to assign the secondary structure state of each residue. The DSSP program assigned 7 states, B, E, G, H, S, T and "the rest" to the residues in our database. For the purpose of this work, H was considered α helix, E was considered β sheet, and the rest were considered coil. Table 1 lists names of all the proteins in our database.

(f) *Prediction accuracy measurements*

In this work, we adopted the commonly used definition of prediction accuracy, which is the percentage of correctly predicted residues for the 3 types of secondary structures:

$$Q_3 = \frac{q_\alpha + q_\beta + q_{\text{coil}}}{N},$$

where N is the total number of residues in the test data sets, q_s is the number of residues of secondary structure type s that are predicted correctly, $s \in \{\alpha\text{-helix}, \beta\text{-sheet}, \text{coil}\}$. To measure the "quality" of the prediction on each type of secondary structure, Matthews' correlation coefficient was also used. For secondary structure type s,

$$C_s = \frac{(p_s \cdot n_s) - (u_s \cdot o_s)}{\sqrt{(n_s + u_s) \cdot (n_s + o_s) \cdot (p_s + u_s) \cdot (p_s + o_s)}},$$

where p_s is the number of positive cases that were correctly predicted; n_s is the number of negative cases that were correctly rejected; o_s is the number of overpredicted cases, and u_s is the number of underpredicted cases. These coefficients thus measure the differences of predictions for different types of structures.

Table 2

Number of residues, helix and sheet contents and names of protein sequences in each test group

Group	No. residue	Helix (%)	Sheet (%)	Proteins and their maximum homology with other proteins (%)	Average of maximum (%)
1	2417	29·6	21·2	1FC2-C(39·5), 2MT2(32·8), 1FXB(34·6), 2B5C(32·9), 3C2C(43·7), 2SNS(31·2), 2LHB(29·5), 1MBD(29·4), 1ETU(24·0), 2ACT(45·0), 1MCP-H(42·8), 1SBT(36·0), 2MDH-B(37·5), 3BCL(19·4)	34·2
2	2465	28·1	19·6	1GCN(44·8), 1PPT(38·9), 4PTI(34·5), 1UBQ(38·2), 3WRP(31·7), 156B(30·9), 2PAB-A(31·6), 1CY3(30·5), 4FXN(31·2), 2STV(26·6), 3RP2-A(31·2), 1TON(37·4), 2CAB(24·2), 4LDH(21·0), 2CTS(19·2)	31·5
3	2550	27·5	22·7	1INS-D(40·0), 2CI2-I(36·9), 1SN3(32·3), 1PCY(31·3), 1HMQ-A(29·2), 2AZA-B(30·2), 2HHB-B(41·1), 1LH1(30·1), 3GAP-A(24·5), 1FB4-H(41·5), 4PTP(41·9), 2CYP(21·8), 2APR(39·1), 3GRS(18·4)	32·7
4	2450	25·1	20·7	1MLT-A(46·2), 2OVO(33·9), 351C(35·4), 3FXC(30·6), 1CCR(34·2), 1PAZ(32·5), 1ECD(28·7), 2LZM(26·2), 3WGA-B(24·6), 1MCP-L(42·3), 2CGA-A(40·0), 1ABP(22·5), 2TBV-C(22·7), 1PHH(21·1)	31·5
5	2492	26·0	20·1	1CRN(37·0), 1NXB(37·1), 1CTF(39·7), 1RNT(29·8), 2CDV(29·9), 1RN3(27·4), 1PP2-L(38·3), 4ATC-B(30·1), 1GP1-A(25·5), 4SBV-C(27·0), 3EST(35·5), 5CPA(21·8), 4TLN(21·8), 3PGK(20·5)	30·1
6	2476	23·7	20·4	3RXN(38·5), 1FDX(37·0), 3ICB(36·0), 2GN5(32·2), 1CPV(35·2), 1LZ1(26·9), 2HHB-A(42·6), 2SOD-B(28·5), 1FC1-A(25·2), 9PAP(46·2), 1SGT(32·1), 4ATC-A(22·3), 1GD1-O(22·3), 4ADH(20·3)	31·8
7	2507	27·2	21·0	4CPA-I(35·1), 1CSE-I(34·9), 1CC5(33·7), 4FD1(31·1), 2SSI(32·7), 1BP2(41·5), 1FX1(27·9), 4DFR-B(27·0), 3ADK(28·9), 3PGM(26·1), 2CNA(24·9), 1RHD(23·5), 2APP(39·3), 2CPP(19·5)	30·4
8	2504	27·4	19·4	1INS-A(47·6), 1TGS-I(35·1), 2ABX-A(36·5), 1HIP(32·9), 1ACX(3·6), 2CCY-A(34·6), 155C(36·6), 4TNC(28·1), 1GCR(24·7), 1FB4-L(43·1), 1TIM-A(24·2), 2PRK(35·5), 2MDH-A(37·7), 7CAT-A(17·1)	33·4

The number of residues, helix and sheet contents, and the names of the protein sequences (subunits) in each test group. (1FC2-C, subunit C of 1FC2.) The number in the parenthesis after each protein name is the maximum homology between that sequence and all sequences in other groups (i.e. the training data set for that test group). The last column is the average of the maximum homology of each group.

(g) *A measure of statistical significance*

When comparing different prediction algorithms, we need to know whether the differences in prediction accuracy among them are statistically significant. Statistics theory gives us a method to compute the "significance interval" for the difference between 2 population proportions (Daniel, 1987). In the case of secondary structure prediction, the "proportion" is the percentage of residues in a set of test data whose secondary structure state has been correctly predicted. Assume the prediction accuracy of 2 algorithms are p_1 and p_2 for 2 test data sets of r_1 and r_2 residues, respectively, and the test data are randomly selected, then we say that we are $a \times 100\%$ confident that the accuracies of the 2 algorithms are really different if

$$|p_1 - p_2| > I,$$

where:

$$I = z(1+a/2) \cdot \sqrt{\frac{p_1(1-p_1)}{r_1} + \frac{p_2(1-p_2)}{r_2}}, \qquad (6)$$

z is the inverse cumulative normal distribution. For example, when $a = 0{\cdot}95$, $z(1+a/2) = 1{\cdot}96$; if $r_1 = r_2 = 20{,}000$, the significance interval is $I \approx 0{\cdot}9\%$; if $r_1 = r_2 = 4000$, $I \approx 2{\cdot}1\%$. If we choose $a = 0{\cdot}99$, $r_1 = r_2 = 20{,}000$, then $I \approx 1{\cdot}2\%$. Thus, the bigger the difference between 2 prediction accuracies, the more significant it is. For the same difference, the more test data used, the more significant it is (and the more confident we are). Equation (6) is used in this paper to determine whether the difference in the accuracies of 2 different predictions is statistically significant.

3. Experiments and Results

(a) *K-way cross-validation*

To evaluate the hybrid system, all the proteins in our database were randomly divided into eight groups. In each test, one group of proteins was used as the test data set and the rest as the training data set. The whole experiment consisted of eight such tests, i.e. eight independent runs of the hybrid system, each time on a different test data set. This way, there was no overlap between training data and test data, and every protein was used as test data once. This is the so-called "k-way cross-validation" testing procedure. Table 2 lists the proteins and the number of residues in each group, the α helix and β sheet contents in the group, as well as the degree of homology between proteins in different groups.

(b) *Window size and other choices*

Throughout this work, a window size of 13 residues was used. Each expert looked at 13 residues at a time and predicted the secondary structure state of the center residue in the window. The Combiner looked at the predictions of 13 residues from each expert and made a final prediction for the center residue. For each amino acid sequence in the test data set, the window was moved over the whole

sequence, and a prediction was made for every residue.

There were other choices that had to be made before starting our k-way cross-validation experiment with the hybrid system. They included (1) the number of hidden units and the number of training cycles for neural networks; (2) the threshold for "nearest neighbors" in MBR module; and (3) the coefficient C_f in the SM. If these choices were made according to the system's performance on the test data set, then they might be fine-tuned to fit the particular data set and make the system's accuracy appear higher than it really is. To avoid this, prior to the k-way validation experiment, a "pilot set" of 20 proteins was randomly chosen from the database, and the above choices were made based on the system's performance on this pilot set. (The pilot set consisted of: 1INS-A, 3RXN, 2MT2, 1CTF, 351C, 2CDV, 1HMQ-A, 1RN3, 1PP2-L, 4FXN, 2SOD-B, 1MBD, 1GP1-A, 1FB4-L, 4PTP, 1TON, 2PRK, 4ATC-A, 4LDH, 1PHH.)

(c) *MBR and SM: training and prediction*

In Memory-Based Reasoning module, first the distance matrices were computed using the training data set. There was one distance matrix for each position of the window, see equation (1) for details. Then for each segment (window) of the amino acid sequences in the test data set, $b_1, b_2, \ldots b_n$, the top 25 instances in the training data set that had the shortest distance to it were considered its neighbors. The strength of prediction (score) for each secondary structure state was the percentage of neighbors in that state weighted by the inverse of their distances. The structure state that had the highest score was taken as the prediction by MBR.

In the statistical module, the frequencies of singletons and pairs of amino acids within a window $a_1, \ldots, a_n$ were calculated for each structure state s_j in the training data set, to approximate the conditional probabilities $p(a_i|s_j)$s and $p(a_i, a_k|s_j)$s. Then for each segment of amino acid sequences in the test data set, $b_1, b_2, \ldots b_n$, these probability values were used to estimate the probability $p(s_j|b_1, b_2, \ldots b_n)$ according to equation (5) ($C_f = 1{\cdot}5$ was used in this work), where s_j is one of the secondary states (α-helix, β-strand and coil). The value of $p(s_j|b_1, b_2, \ldots b_n)$ was taken as the score of prediction for structural state s_j, and the state that had the highest score was taken as the prediction by SM.

(d) *Training neural networks*

One important issue in training neural networks by the Back-propagation algorithm is deciding when to stop training. If a network is trained through too many cycles, the network tends to memorize the training examples but generalizes poorly on the inputs that it has not been trained on (i.e. test data). One practice is to monitor the performance of the network being trained on the test data, and to stop training when the performance peaks. This strategy cannot be used in real situations where the true answer is truly unknown. We used the following techniques to solve this problem: (1) limiting the number of training cycles; (2) limiting the number of hidden units, thus the number of free variables (the "memory capacity") in the network; (3) when available, using a separate control data set to control when to stop training the network, that is, to monitor the performance of the network being trained on the control data set and stop training when the performance peaks.

A one-hidden-layer neural network was used as one of the three experts. This network is referred to as EXPERT-NN in the following discussion. A total of 21 input units was used to encode one residue, one unit for each of the 20 amino acid types plus one end marker. With a window size of 13 residues, there were $21 \times 13 = 273$ input units total. EXPERT-NN had three output units, one for each of the three secondary structure states (α-helix, β-sheet and coil). The network had only two hidden units. EXPERT-NN was trained up to 200 epoch cycles on the training data set, and the network weights that gave the best performance on the *training set* during training were saved as the final result of training. The activation of the output units were used as the score of prediction for the corresponding secondary structure.

The Combiner of our hybrid system was also a one-hidden-layer neural network. The Combiner took the outputs of the three experts as inputs and made final predictions based on these outputs. For every residue, each expert generated three numbers representing the prediction score for α-helix, β-sheet and coil, respectively. The Combiner took the predictions of 13 residues from each of the three experts as its input, thus it had $13 \times 3 \times 3 = 117$ input units. It also had three output units, one for each of the secondary structure states. As discussed in Methods and Materials, in order to train the Combiner, the training data set was divided into two halves, which will be referred to as $\{H_1\}$ and $\{H_2\}$ in the following discussion. The three experts were first trained on the first half of the data set $\{H_1\}$. Then they were applied on the second half $\{H_2\}$. Their outputs on $\{H_2\}$, $\{\text{Output}(H_2)\}$, were then used as input patterns for training the Combiner. Similarly, the three experts were also trained on $\{H_2\}$ and their outputs on $\{H_1\}$, $\{\text{Output}(H_1)\}$, were recorded. Finally, the Combiner was trained up to 200 epoch cycles, using $\{\text{Output}(H_2)\}$ as training data and $\{\text{Output}(H_1)\}$ as control data. The weights that gave the best performance on both $\{\text{Output}(H_1)\}$ and $\{\text{Output}(H_2)\}$ during training were saved as the result of training the Combiner. A total of 30 hidden units was used in the Combiner. Since there was a control data set here, the number of hidden units was less crucial here than in EXPERT-NN.

(e) *The hybrid system improved prediction accuracy*

Table 3 shows the results for the eight test data sets in our k-way cross-validation experiment. Table

Table 3
Prediction accuracy on test data sets

Group	No. sequence	No. residue	EXPERT-NN (%)	SM (%)	MBR (%)	Hybrid (%)
1	14	2417	60·7	62·8	64·4	65·3
2	15	2465	63·8	63·3	63·9	66·3
3	14	2550	62·2	63·6	64·7	66·2
4	14	2450	62·3	62·9	64·0	66·2
5	14	2492	63·2	62·4	64·4	66·6
6	14	2476	65·2	64·1	65·8	68·1
7	14	2507	62·3	63·8	63·1	65·1
8	14	2504	64·9	65·5	65·5	67·5
Total	113	19.861	63·1	63·5	64·5	66·4

The prediction accuracy on each test data set by the 3 experts and by the hybrid system. No. sequence is the number of sequences (subunits) in each group: No. residue is the number of residues in each group.

4 shows the accuracy for each sequence. Overall, for the prediction of secondary structures α-helix, β-sheet and coil, EXPERT-NN was 63·1% accurate, MBR was 64·5% and SM was 63·5%. The hybrid system was 66·4% accurate. The total number of residues used in the experiment was 19,861. According to the statistical significance measures described in equation (6), the improvement of the hybrid system over each expert was statistically significant (with higher than 0·99 confidence level). Thus we are highly confident that our hybrid system really improved the prediction accuracy.

The Matthews' correlation coefficients for each expert and for the hybrid system are shown in Table 5. All three experts had similar coefficients and produced better prediction on α helix and coil than on β strand. One reason for this might be that a single β strand can hardly be stable; more than one strand get stabilized when they interact with one another to form a β sheet; this interaction is often not local along the sequence and thus cannot be captured very well by the local approach. Thus, no matter what algorithm is used, β strand would still be the most difficult state to predict. The hybrid system improved the prediction for all the structure states.

(f) *A single small test data set is dangerous*

From Table 3 we computed the average difference in prediction accuracy among the three different experts for the same sets of test data, which was 0·9%. This shows that the overall accuracies of the three experts were very close. We also computed the average difference for the same expert on the eight different test data sets, which was 1·3%. Thus, if each test data set is observed independently, the difference in prediction accuracy caused by the different test data sets were at least as large as the difference brought about by the different experts. This observation argues strongly against using a single small test data set: (1) "statistical noise" can

Table 4
The accuracy on each protein sequence (subunit) by the three experts and the hybrid system

Protein	SM (%)	MBR (%)	EXPERT-NN (%)	Hybrid (%)
155C	64·9	74·6	64·9	70·9
156B	62·7	61·8	70·0	64·5
1ABP	59·8	53·3	57·8	57·5
1ACX	70·1	63·6	60·7	61·7
1BP2	52·0	55·3	52·0	52·8
1CC5	75·9	72·3	69·9	77·1
1CYR	67·6	70·3	70·3	73·0
1CPV	63·9	55·6	60·2	66·7
1CRN	50·0	56·5	50·0	52·2
1CSE-I	65·1	68·3	63·5	71·4
1CTF	55·9	57·4	50·0	54·4
1CY3	68·6	70·3	75·4	74·6
1ECD	44·9	43·4	36·8	45·6
1ETU	68·9	71·4	70·9	77·0
1FB4-H	71·6	75·5	65·9	71·2
1FB4-L	66·2	70·8	61·1	68·1
1FC1-A	56·8	60·2	60·2	58·3
1FC2-C	74·4	76·7	60·5	79·1
1FDX	72·2	79·6	70·4	72·2
1FX1	52·4	57·8	57·1	57·8
1FXB	82·7	75·3	70·4	77·8
1GCN	55·2	41·4	48·3	48·3
1GCR	45·4	52·9	56·9	52·9
1GD1-O	57·1	60·1	64·3	63·1
1GP1-A	64·7	60·3	65·8	65·2
1HIP	64·7	67·1	60·0	60·0
1HMQ-A	60·9	56·6	58·4	63·7
1INS-A	47·6	38·1	47·6	42·9
1INS-D	76·7	73·3	76·7	83·3
1LH1	70·6	61·4	68·0	72·5
1LZ1	73·8	66·9	70·0	68·5
1MBD	67·3	67·3	60·1	68·0
1MCP-H	64·4	75·7	61·3	81·5
1MCP-L	65·0	70·0	61·4	69·1
1MLT-A	42·3	50·0	50·0	46·2
1NXB	67·7	61·3	66·1	62·9
1PAZ	59·2	65·8	63·3	66·7
1PCY	58·6	64·6	65·7	67·7
1PHH	60·4	58·1	66·0	64·2
1PP2-L	65·4	70·7	66·9	69·2
1PPT	77·8	83·3	75·0	88·9
1RHD	64·2	66·2	64·8	66·2
1RN3	54·8	62·9	58·1	66·9
1RNT	64·4	61·5	68·3	67·3

Table 4 *(continued)*

Protein	SM (%)	MBR (%)	EXPERT-NN (%)	Hybrid (%)
1SBT	63·3	68·4	66·5	67·3
1SGT	66·7	75·4	65·0	78·3
1SN3	73·8	76·9	70·8	72·3
1TGS-I	63·2	57·9	66·7	56·1
1TIM-A	70·6	66·1	69·0	73·8
1TON	70·2	78·2	69·7	75·6
1UBQ	61·8	55·3	65·8	63·2
2ABX-A	89·2	78·4	81·1	81·1
2ACT	68·3	72·0	67·0	73·4
2APP	61·9	57·9	55·7	59·4
2APR	69·2	68·9	66·8	68·3
2AZA-B	45·7	51·2	48·1	48·1
2B5C	65·9	67·1	63·5	55·3
2CAB	64·1	69·5	71·1	69·5
2CCY-A	75·6	63·8	79·5	83·5
2CDV	72·9	73·8	71·0	76·6
2CGA-A	64·9	74·7	58·4	72·2
2CI2-I	60·0	70·8	64·6	70·8
2CNA	57·8	58·6	60·3	59·1
2CPP	69·6	61·5	61·5	65·9
2CTS	68·0	62·5	64·1	69·1
2CYP	63·8	63·5	60·1	64·8
2GN5	69·0	65·5	62·1	70·1
2HHB-A	72·3	70·2	66·0	78·0
2HHB-B	61·0	59·6	47·9	61·6
2LHB	68·5	65·8	60·4	67·1
2LZM	60·4	64·6	61·6	61·6
2MDH-A	55·2	57·4	56·8	61·7
2MDH-B	48·0	53·2	47·1	52·0
2MT2	95·1	91·8	95·1	96·7
2OVO	62·5	64·3	60·7	66·1
2PAB-A	50·0	50·9	59·6	58·8
2PRK	63·8	69·9	63·8	71·3
2SNS	61·0	60·3	61·7	63·8
2SOD-B	66·2	67·5	72·2	71·5
2SSI	74·8	69·2	72·0	78·5
2STV	51·1	53·8	51·1	53·8
2TBV-C	59·5	62·0	60·1	64·5
351C	81·7	76·8	79·3	86·6
3ADK	64·4	68·0	61·9	69·6
3BCL	49·7	40·7	50·0	44·7
3C2C	71·4	82·1	59·8	68·7
3EST	70·1	77·7	64·9	79·3
3FXC	72·4	75·5	65·3	77·6
3GAP-A	50·5	60·1	54·8	56·7
3GRS	58·8	55·7	62·7	63·3
3ICB	85·3	89·3	86·7	90·7
3PGK	66·0	66·0	67·5	67·7
3PGM	63·9	67·8	67·4	68·3
3RP2-A	54·9	66·7	55·3	62·0
3RXN	82·7	84·6	84·6	84·6
3WGA-B	80·1	77·2	80·7	80·7
3WRP	75·2	70·3	66·3	73·3
4ADH	57·2	54·3	59·4	57·5
4ATC-A	58·4	61·3	61·0	63·5
4ATC-B	62·1	61·4	64·1	62·7
4CPA-I	78·4	67·6	73·0	70·3
4DFR-B	59·1	58·5	59·7	62·9
4FD1	67·9	73·6	75·5	72·6
4FXN	68·1	63·8	64·5	68·8
4LDH	59·8	58·3	59·2	61·3
4PTI	70·7	60·3	70·7	62·1
4PTP	71·4	82·5	68·8	77·8
4SBV-C	53·2	54·1	57·2	55·9
4TLN	57·9	62·3	58·2	65·8
4TNC	83·7	78·1	77·5	79·4
5CPA	60·9	63·8	63·8	66·4
7CAT-A	65·7	64·5	65·5	64·9
9PAP	70·3	80·7	70·3	76·4
Total	63·5	64·5	63·1	66·4

Table 5

The Matthews' correlation coefficients for each expert and the hybrid system on each structural state

Method	C_{coil}	C_α	C_β
SM	0·390	0·418	0·350
MBR	0·396	0·416	0·357
EXPERT-NN	0·395	0·383	0·333
Hybrid	0·429	0·470	0·387

Table 6

The percentage of total residues for which two experts produced the same secondary structure prediction

EXPERT-NN	MBR	SM	Hybrid
EXPERT-NN	76·6%	84·3%	82·9%
MBR		77·7%	82·0%
SM			83·6%

Table 7

Percentage accuracy

One correct	Two correct	Three correct	Three incorrect
76·6%	64·0%	50·6%	19·4%

make the same algorithm have different accuracies on different test data sets if the sets are small; (2) the difference among different algorithms, even if it truly exists, can be easily "buried" by such statistical noise. Thus, large or multiple test data sets should be used whenever possible.

(g) *Different algorithms made similar predictions*

The three experts used in our experiments did not only have similar overall prediction accuracies, but also made similar predictions for each sequence. Table 6 shows the percentage of the residues in the test data sets for which different experts produced the same predictions. On average, each pair of experts agreed with each other on about 80% of the total 19,861 residues. All three experts produced the same prediction on about 70% of the total residues (not shown in the Table). Table 7 shows the percentage of residues for which at least one expert was correct, at least two experts were correct, all three experts were correct and all three experts gave the same but wrong predictions. For about 20% of the residues, all three experts produced the same but wrong predictions. This, together with the information from Table 6 indicates that the "local rules" (the rules mapping short segments of amino acid sequences to secondary structures) obtained by the three very different experts were actually quite similar, but they did not apply quite as well to the test data. This may suggest an upper bound on the secondary structure prediction accuracy based on local information from the currently available data. In places where all algorithms were the same but

Table 8
Accuracy of predictions

Method	α Helix				β Sheet			
	Correct	Over	Under	Coef.	Correct	Over	Under	Coef.
SM	345	195	171	0·392	449	299	375	0·283
MBR	309	182	207	0·341	404	281	420	0·244
EXPERT-NN	314	181	202	0·351	421	316	403	0·238
Hybrid	353	162	163	0·445	450	234	374	0·335

The number of correct predictions (Correct), underpredictions (Under), overpredictions (Over) for α helix and β sheet by each expert and the hybrid system. Coef. is the Matthews' correlation coefficient (see Methods and Materials).

incorrect, the structures might be determined by non-local interactions. Among the residues where all three experts did agree with one another, they were correct for 71% of the residues. Thus, if we only consider the cases where all three experts agreed, we have a much higher prediction accuracy.

(h) *Homology between training and test data set*

It is known that if the training data and the test data are identical or highly homologous, then the prediction accuracy could be misleadingly high. However, when the degree of homology between training and test data was below 50%, we did not find strong positive correlations between the prediction accuracy and the degree of homology. For example, the degree of homology between 1GCN, 1MLT-A and 1INS-A and their training data were 44·8%, 46·2% and 47·6% respectively, and their prediction accuracies were quite low (see Table 4); whereas 2CTS, 3GRS and 7CAT-A had very low homology with their training data (19·2%, 18·4% and 17·1%, respectively), but their prediction accuracies were much higher.

(i) *Secondary structures as individual units*

Often it is more important to predict correctly the occurrence or absence of a secondary structure (α helix or β strand) as a whole rather than just to predict the states of individual residues. Thus the following criteria were also used in this work to evaluate the predictions of different methods: we took an α helix or β strand as an individual unit, and checked how many of these secondary structures were correctly predicted (positive cases), how many of them were not predicted at all (underpredicted), how many were predicted which do not exist in the real structures (overpredicted). Then a Matthews' correlation coefficient is calculated for each method. We found that the hybrid system had the most positive cases and the fewest overpredictions and underpredictions. (Note that this is in terms of number of secondary structures, not residues.)

Specifically, in this work an α helix is said to have been predicted if at least four continuous residues in a sequence are predicted to be in H state; a β strand is said to have been predicted if at least two continuous residues were predicted to be in E state. If the overlapping region between a real secondary structure and a predicted secondary structure of the same type is greater than half of the length of the real structure or the predicted structure, then the real secondary structure is considered to have been correctly predicted. If more than one predicted secondary structure overlaps with one real secondary structure, only one of the predicted secondary structures is considered as a correct prediction, and the rest are counted as overpredictions. If one predicted secondary structure overlaps with more than one real secondary structure, only one of the real secondary structures is considered as correctly predicted, and the rest are counted as underpredictions. Table 8 lists the correct predictions, overpredictions, underpredictions and Matthews' coefficient for α helix and β strand by each expert and the hybrid system according to these criteria. (In calculating Matthews' coefficients, the residues between 2 helices (sheets) are considered to form 1 non-helix (non-sheet).) The hybrid system produced the best result by this criteria as well.

No doubt the above criteria are not perfect. And the details such as the numbers 2 for β strand and 4 for α helix are to some extent arbitrary. However, we need some criteria to capture the intuitive notion of "how many secondary structures are predicted correctly". We believe the above criteria serves as an unbiased, first-order approximation to that. It provides a new perspective to evaluate different prediction methods. For example, SM is better than MBR and EXPERT-NN by this criteria, whereas that is not the case if we count the number of correctly predicted residue states (see Table 4).

(j) *An example*

Figure 4 shows the prediction for protein 1PAZ by each expert and the hybrid system. It illustrates the points discussed in previous sections. Note that the inputs from each expert to the Combiner in our hybrid system are the three prediction scores for each of the three states (α helix, β sheet and coil), not just the predicted states themselves; and the Combiner looks at the prediction scores of 13 posi-

```
          0         10        20        30        40        50        60
Protein:  ENIEVEMLNKGAEGAMVFEPAYIKANPGDTVTFIPVDKGHNVESIKDMIPEGAEKFKSKINENYVLTVTQ
Structure:-eeeeeeeeee--eeeeee--eeee----eeeeee-------ee------------------eeee---

SM:       ----ehhhh-----hhhh-hh-ee------eeeee-----------hh-h--hhhhhhhh----eeee--
MBR:      ---eehhh-h-----ee-----------eee--------------h--h--hhhhh-h----------
Expert-NN:---hhhhh------hehh----------eee--------------------hhh------eeeee--

Hybrid:   ----eee--------ee----------eeee------------------hhhhhh-----eee--

          70        80        90        100       110
Protein:  PGAYLVKCTPHYAMGMIALIAVGDSPANLDQIVSAKKPKIVQERLEKVIA
Structure:-eeeeee---------eeeeee------hhhhhh----hhhhhhhhhhh-

SM:       ----eee-----hhhhhhhee-------hhhhhh----hhhhhhhh----
MBR:      ----ee----------eeeee-------hhhhh-----hh-hhhhhhh--
Expert-NN:---eee------hhhhheeee---------hhh-------hhhhhhhh--

Hybrid:   ----ee------hhheeeeee--------hhhh-----hhhhhhhhhh--
```

Figure 4. The secondary structure prediction generated by SM, MBR, EXPERT-NN and the hybrid system. Structure indicates the secondary structure assignment by the DSSP program.

tions at a time. That is why we can see that in certain cases the Combiner can override the majority of the three experts, such as between residue 0 and 10 of 1PAZ. In some places, all three experts made the same but wrong predictions. For example, there is a short β strand between residue 40 and 50 that none of the experts predicted; and they all predicted a helix between residue 50 and 60 that does not exist in the real structure. In both cases the Combiner made the same mistake also. None of the experts could always make better predictions than others. For example, SM is the only one that predicted the sheet between residue 20 and 30. MBR is the only one that did not give the false prediction of a helix between residue 80 and 90, and EXPERT-NN made fewest mistakes between residue 50 and 70.

(k) *Comparison with other methods*

Qian & Sejnowski (1988) used a "cascaded neural network" system in secondary structure prediction and achieved 64·3% accuracy on a test set of 15 proteins (containing 3520 residues). Their system contained two networks: the first network took amino acid sequences as inputs and produced the initial prediction; the second network "cleaned up" this initial prediction to produce final predictions. This system could also be seen as a hybrid system but with only one expert. We applied their method to our eight test data sets. Table 9 shows the results. This was done not only to compare the final results, but also to see whether adding two more experts could really help. The overall prediction accuracy of the cascaded system on our test data sets was 64·0%, which, on a much larger scale (19,861 *versus* 3520 residues), confirmed Qian & Sejnowski's results. However, the improvement of the cascaded network over a single network was only 0·5%, not 1·5% as reported in their paper. According to our statistical significance measure (equation (6)), both 0·5% for 19,861 residues and 1·5% for 3520 residues were not statistically significant differences at confidence level 0·95. We also noticed that there was

Table 9

The accuracy on the eight test data sets by Cascaded networks of Qian & Sejnowski (1988)

Group	No. sequence	No. residue	Single network (%)	Cascaded network (%)
1	14	2417	61·9	62·5
2	15	2465	64·3	64·3
3	14	2550	62·5	63·2
4	14	2450	62·7	62·9
5	14	2492	63·3	64·3
6	14	2476	65·5	66·6
7	14	2507	62·6	62·9
8	14	2504	65·3	65·5
Total	113	19,861	63·5	64·0

some difference in prediction accuracy (0·4%) between their single network and our EXPERT-NN, even though they were both trained and tested on the same data sets. The reason was that according to Qian & Sejnowski's method, the performance of their network on the test data set was monitored during training. The network weights that performed the best on the test set were saved and used. Whereas in our work, the EXPERT-NN never saw the test data set during training (see Methods and Materials).

The GOR III algorithm by Gibrat *et al.* (1987) was reported to have achieved 63% prediction accuracy by using correlations between certain pairs of amino acids and secondary structures. Biou *et al.* (1988) further improved the GOR III algorithm by combining its result with that of two other algorithms, the Homologue method and the bit pattern method, achieving a reported accuracy of 65·5% (we refer to this combined algorithm as GOR-Combined in the following discussion). We ran the GOR-Combined program on protein sequences in our database. Since their program contained the statistics calculated using their database, i.e. their training data, we divided our database into two groups. Group A contained sequences that were identical or more than 50% homologous to their training data. Group B contained the rest of the sequences. There were 64 sequences in group A and 49 sequences in group B. Apparently group B should be used as the test data to compare the GOR-Combined against other algorithms, because a prediction algorithm could easily have a very high prediction accuracy on protein sequences that are either identical or highly homologous to its training data, which cannot be used as an objective assessment of the algorithm's prediction accuracy. For group B, the GOR-Combined was 62·4% accurate. This is 3% lower than their reported result. One reason for this might be that GOR-combined algorithm used certain rules to combine the outputs of different methods, and those rules did not work quite as well for proteins not in its database. We used the 64 protein sequences in group A to train our hybrid system and applied it to the 49 protein sequences in group B. It was 65·3% accurate. This is about 1% lower than the average accuracy of the hybrid system in the k-way cross-validation experiment. We believe this was due to the smaller training set used here, which had only 64 protein sequences.

Table 10

Accuracies of different algorithms for three states (helix, sheet, coil) prediction

Method	Accuracy (%)
Lim (1974)	59
Chou & Fasman (1974)	50
Levin *et al.* (1986)	62·2
GOR III	63
Qian & Sejnowski (1988)	64·3
Holley & Karplus (1989)	63·2
Hybrid	66·4

Table 10 lists the results of several other algorithms. The results were obtained from each author's original report except those by Lim (1974) and Chou & Fasman (1974), because in their original reports they used the same data set for both training and testing. Kabsch & Sander (1983*b*) assessed the accuracies of these two algorithms with separate test data, and the results were included in the Table instead. Among these, our hybrid system was tested with the largest set of protein data and it gave the highest prediction accuracy.

4. Discussion

The idea of combining the strength of different methods is not entirely new in either machine learning research (Wolpert, 1990) or protein secondary structure prediction. For example, Biou *et al.* (1988) used certain rules to combine three methods. However, the authors did not explain how their rules were generated in the first place. Thus it is difficult for us to justify the use of those rules. In our hybrid system, the Combiner learns how to combine the outputs of different experts automatically from the training data. A novel procedure has to be developed to train the Combiner because different experts can have very different behaviors. For example, after training, some experts can be 100% correct on the training data set while others may be only 70% correct on the training data, even though they have very similar prediction accuracies for proteins not in the training set. Our training procedure for the Combiner can cope with experts that have such different characteristics.

This work showed that although different algorithms may have very similar overall secondary structure prediction accuracies, their detailed predictions can be different. No single algorithm always gives a better prediction than others. A combination of them can produce a statistically significant improvement over each individual method. We developed a way to train a Combiner, which learned to combine the outputs of different experts automatically. A neural network was used

as the Combiner in this work. But it is not the only choice. A MBR system, for example, can also be used as a Combiner. This paper is the first place where the SM algorithm and the particular MBR distance function have been introduced. Their accuracy were as good as or even better than any other single algorithm reported to date for secondary structure prediction. They deserve a more detailed discussion, which is beyond the scope of this paper and is done elsewhere (X. Zhang, unpublished results). The techniques we used to control the training of artificial neural networks were not only objective but also effective. For a single one-hidden-layer network, the accuracy was 63·1% with our techniques (to control training purely based on the training data). Whereas the other approach, to monitor the performance of the network on the test data during training, was 63·5%. The difference between them was only 0·4%. Thus our techniques produced near-optimal training.

One of the reviewers of this paper raised the issue of whether residues assigned to state G by the DSSP program (Kabsch & Sander, 1983*a*) should be considered as in helix, especially when they are adjacent to state H. In our original experiments, we wanted to make our result directly comparable with results obtained by other researchers, such as Qian & Sejnowski (1988), since the main point of this paper is that for the same secondary structure assignment, the hybrid system gives better prediction than other algorithms. Thus we used the same assignment as Qian & Sejnowski (1988), i.e. only considering H for α helix and E for β strand. After we received the reviewer's comments, we did the following experiment: we assigned G states to be helix if they are adjacent to H, otherwise assigning them to be coil. This way, among the 19,861 residues in our database, 162 residues (0·8% of the total residues) were assigned differently, i.e. to helix instead of coil. Then we compared the original prediction of our hybrid system with this new assignment. It is 66·1% accurate. This is very close to the original accuracy of 66·4%. The change in accuracy (0·3%) is much smaller than the change in the assignment (0·8%). This means that even though the hybrid system was trained with a different assignment, it can still predict correctly most of the new assignment. This is in accordance with observations by other researchers (e.g. Richardson & Richardson, 1988) that there are certain ambiguities on secondary structure boundaries assigned by DSSP.

Good criteria for evaluating and comparing different prediction algorithms are crucial for the progress of this research field. In this work, we made use of the significance interval measure from statistics, which could tell us whether the differences observed are significant or not, and what factors can influence that. We emphasize the importance of the fact that in our tests, the hybrid system never looked at the test data during training, thus making the performance of the system on the test data as objective as possible. The k-way cross-validation allowed us to test our hybrid system with as many data as we have, and yet still avoided overlapping between the test data and training data. Some researchers have used one protein in each test group, thus maximizing the training data size. However, the extremely large amount of computation in our work prevented us from doing that (i.e. $k = 113$, the total number of protein sequences of our database). We choose $k = 8$, which did not reduce the size of each training data set very much, and yet cut the amount of computation dramatically. Even so, a large amount of computation was still needed to carry out our experiment. This involved (1) computing many statistics for SM and distance matrices for MBR; (2) pattern matching and sorting through the whole database to find neighbors in MBR; and (3) training many neural networks with large numbers of input/output examples. The experiment was done on a massively parallel computer Connection Machine CM-2. The particular machine we used had 4096 processors. In general, CM-2 can have up to 65,536 processors.

There are many important issues in protein secondary structure prediction, such as: (1) is "the percentage of correctly predicted residues" the best measure for success? (2) What is the best way to assign the secondary structures to a protein once its three-dimensional co-ordinates are known? (3) What is the right criteria for homology in selecting test/training data? A comprehensive discussion of these issues is beyond the scope of this paper. The emphasis here is to demonstrate that our hybrid system gives significantly better performance than individual algorithms and all previous methods, using the same criteria in selecting data and the same accuracy measure as used by other researchers.

We are grateful to Eric Lander and Tau-Mu Yi for valuable comments and suggestions on several drafts of this paper. We thank Christian Sander for providing us with the DSSP program and Anand V. Bodapati for helpful discussions. We also thank the anonymous reviewers who gave us insightful comments.

References

Bahadur, R. R. (1961). On classification based on responses to n dichotomous items. In *Studies in Item Analysis and Prediction*, chapt. 10, Stanford University Press.

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

Chou, P. Y. & Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry*, **13**, 222–244.

Daniel, W. W. (1987). *Biostatistics: A Foundation for Analysis in the Health Sciences*. 4th edit., John Wiley & Sons.

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

Gibrat, J.-F., Garnier, J. & Robson, B. (1987). Further

developments of protein secondary structure prediction using information theory. *J. Mol. Biol.* **198**, 425–443.

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

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

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

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

Lathrop, R. H., Webster, T. A. & Smith, T. F. (1987). nishiARIADNE: pattern-directed inference and hierarchical abstraction in protein structure recognition. *Commun. A.C.M.* **30**, 909–921.

Lazarsfeld, P. F. (1961). The algebra of dichotomous systems. In *Studies in Item Analysis and Prediction*, chapt. 8. Stanford University Press.

Levin, J. M., Robson, B. & Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS Letters*, **205**, 303–308.

Levitt, M. (1983). Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.* **170**, 723–764.

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

McClelland, J. L. & Rumelhart, D. E. (eds) (1986). *Parallel Distributed Processing*. MIT Press.

Nishikawa, K. & Ooi, T. (1986). Amino acid sequence homology applied to the prediction of protein secondary structures, and joint prediction with existing methods. *Biochim. Biophys. Acta*, **871**, 45–54.

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

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

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing*, MIT Press.

Skolnick, J. & Kolinski, A. (1990). Simulations of the folding of a globular protein. *Science*, **250**, 1121–1125.

Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Commun. A.C.M.* **29**, 1213–1228.

Taylor, W. R. & Thornton, J. M. (1984). Recognition of super-secondary structure in proteins. *J. Mol. Biol.* **173**, 487–514.

Wolpert, D. H. (1990). *Stacked Generalization*. Tech. Report LA-UR-90-3460, LANL.

Edited by F. Cohen