Journal of Biomolecular NMR, 10 (1997) 45–52

J-Bio NMR 446

Performance of a neural-network-based determination of amino acid class and secondary structure from ^{1}H-^{15}N NMR data

Kai Huang[a], Michael Andrec[a], Sarah Heald[b], Paul Blake[b] and James H. Prestegard[a,*]

[a]*Department of Chemistry, Yale University, New Haven, CT 06511, U.S.A.*
[b]*Structural Chemistry, Bayer Corporation, Pharmaceutical Division, West Haven, CT 06516, U.S.A.*

Received 30 October 1996
Accepted 13 March 1997

Keywords: ^{15}N-labeled proteins; Protein structure; Automated assignment

Summary

A neural network which can determine both amino acid class and secondary structure using NMR data from ^{15}N-labeled proteins is described. We have included nitrogen chemical shifts, $^{3}J_{H^{N}H^{\alpha}}$ coupling constants, α-proton chemical shifts, and side-chain proton chemical shifts as input to a three-layer feed-forward network. The network was trained with 456 spin systems from several proteins containing various types of secondary structure, and tested on human ubiquitin, which has no sequence homology with any of the proteins in the training set. A very limited set of data, representative of those from a TOCSY-HSQC and HNHA experiment, was used. Nevertheless, in 60% of the spin systems the correct amino acid class was among the top two choices given by the network, while in 96% of the spin systems the secondary structure was correctly identified. The performance of this network clearly shows the potential of the neural network algorithm in the automation of NMR spectral analysis.

Introduction

Current NMR-based methods for the three-dimensional structure determination of macromolecules require that resonances first be assigned to particular sites in a known primary sequence. This assignment step is frequently labor intensive and very time consuming. Hence, considerable effort has been devoted to the development of computer software that can automate this process (see Zimmerman and Montelione (1995) for a review). A few years ago, our laboratory attempted to contribute to this development by designing and training a neural network to take over some of the initial manual decisions in the assignment process (Hare and Prestegard, 1994). We present here a further refinement of this approach.

Traditionally, assignment strategies for proteins (Wüthrich, 1986) have been divided into several phases: association of resonances into spin systems representing individual residues; identification of possible amino acid types for these residues; and sequential connection of spin systems to allow proper placement in the known amino acid sequence. Many of the advances in the automation of assignments have occurred in the last step. In particular, the use of double isotopic labeling (^{13}C,^{15}N) has allowed the sequential connection of spin systems using through-bond scalar couplings. The data obtained tend to be less subject to misinterpretation than previous NOE-based data, and programs to make use of this type of data can be easily written using straightforward deductive logic (Friedrichs et al., 1994; Meadows et al., 1994; Olson and Markley, 1994; Zimmerman et al., 1994; Bartels et al., 1995; Mittard et al., 1995; Neidig et al., 1995). There are a few programs that have been developed to semiautomate the process of identification of amino acid types. In these approaches, a pattern-matching method (Van de Ven, 1990; Oschkinat et al., 1991; Bartels et al., 1995) is usually applied to compare the scalar coupling topologies of spin systems to ideal topologies. The accuracy of classification is often compromised by incomplete coupling connectivities, and resonance degeneracies.

Coupling topologies can be supplemented by chemical shifts and cross-peak intensities which also carry informa-

*To whom correspondence should be addressed.

tion about amino acid type. However, in each case, the values of these parameters not only correlate with a particular amino acid type, but are also influenced by sequence, structure and dynamics in a way that makes identification by deductive logic less straightforward. Moreover, it is not clear that even expert spectroscopists have fully recognized correlations among the various types of information which may help identify a clear path to assignments.

A neural network approach has several advantages. It can construct its own path for correlating input data with output assignments if a sufficient number of correctly assigned examples are given. Thus, misassignments due to inadequate manual programming of the correlations between input and output can be avoided. Unlike many deductive strategies, the network assigns probabilities (or levels of activation) to various choices instead of making a single definitive choice. This is advantageous when data are inadequate for making an assignment with complete certainty. Finally, software packages containing the implementation of standard algorithms and network architectures are readily available, so the major programming effort involves only the design of input and output representations and the transformation of raw experimental data into the desired input form. A primary disadvantage of a neural network approach is the need for a large number of properly assigned examples. We will address this limitation later in our presentation.

In our previous investigation of neural network applications, we had used amide proton–side-chain proton cross peaks in 2D homonuclear TOCSY and 3D heteronuclear TOCSY experiments as the primary input. Total correlation experiments of this type offer a great deal of information about amino acid type in a single data set. The cross peaks which connect a specific backbone amide proton resonance to resonances from its side-chain protons come close to defining a complete amino acid spin system. The number of peaks, their chemical shifts, and relative intensities all carry information about amino acid type. This information was encoded in a 71-unit input layer. The network architecture used contained two additional layers and was described as a three-layer feed-forward network. It was chosen from a commercial package which also contained training algorithms (McClelland and Rumelhart, 1988). The network was trained on data from a highly α-helical protein (acyl carrier protein (ACP) from *E. coli*) and tested on a closely related protein (ACP from spinach).

The application was reasonably successful, and the utility of a neural-network-based approach was demonstrated. However, there were some clear limitations. First, we were not able to fully utilize all of the information in a TOCSY spectrum. Since it is known that α-proton chemical shifts are strongly influenced by secondary structure (Wishart et al., 1991), we feared that the useful information about amino acid type in the α-proton chemical shifts would be inextricably mixed with secondary structure information. The most downfield of the proton cross peaks between 3.0 and 6.0 ppm were, therefore, excluded. For some amino acids (glycine for example), this may leave little or no additional data for input. For others (serine and threonine), α-proton peaks cannot be identified with absolute certainty because of the chemical shift similarity of β-proton peaks, and identification of these amino acid types is compromised.

An alternative to eliminating complex input is to provide sufficient additional information so that a properly trained network is able to decipher the correlations in the input. Since we know that in the case of α-proton peaks the likely source of complexity is variation in secondary structure, it seems logical to add data which more directly reflect secondary structure. Such data are readily available if we recognize that many proteins studied today can be economically prepared with ^{15}N enrichment. Proton resonance information from a ^{15}N-edited TOCSY experiment (TOCSY-HSQC) had been used in our previous work, but ^{15}N had only been used to resolve spin systems. It is now clear, however, that secondary structure is a major factor contributing to the dispersion of ^{15}N chemical shifts (Glushka et al., 1989,1990; de Dios et al., 1993; Braun et al., 1994; Le and Oldfield, 1994). In fact, explicit correlations between ^{15}N chemical shifts and the dihedral angles, ϕ_i and ψ_{i-1}, have been documented. This results in a general tendency for α-helical residues to have upfield ^{15}N shifts in addition to the more widely recognized upfield H^{α} shifts. The inclusion of ^{15}N chemical shift data could, therefore, help in deciphering secondary structure effects. In addition, there is complementary amino acid type information in ^{15}N chemical shifts, e.g. glycine nitrogens have particularly small shift values. Another fairly clear source of secondary structure information is the $^3J_{H^NH^\alpha}$ coupling constant. Through the Karplus relationship, $^3J_{H^NH^\alpha}$ coupling constants can be related directly to the ϕ backbone dihedral angle (Karplus, 1959; Wüthrich, 1986). A consequence is that α-helical residues have small coupling constants while residues in more extended secondary structure have larger coupling constants. By combining α-proton shifts, nitrogen shifts, and $^3J_{H^NH^\alpha}$ values with our previous TOCSY-HSQC data, there should be sufficient information to determine both the amino acid and secondary structure type.

Methods

Overview on programs

Our classification method requires a peak list from a TOCSY-HSQC spectrum and $^3J_{H^NH^\alpha}$ values. A partially automated interactive peak filtering and sorting program, *mkcol*, which is written in *perl* v. 5.0 supplemented with the graphics package *pgplot*/*pgperl* has been developed to classify TOCSY-HSQC peaks into spin systems. Then a C program, *prenet*, is used to construct neural network

input vectors using information from proton cross peaks, nitrogen chemical shifts, and $^3J_{H^N H^\alpha}$ values. Additional input vectors for training are also generated using *prenet* by randomization of elements in a real input vector. As in our previous work, we choose a feed-forward network with input, hidden, and output layers from a software package developed by McClelland and Rumelhart (1988) and train the network using modules from that package.

Application of a trained network to the evaluation of an unknown data set is carried out using a C program, *pats* (*p*rediction of *a*mino acid *t*ypes and *s*econdary structure). The inputs to *pats* include a peak list, which has been sorted into spin systems containing both nitrogen chemical shifts and aliphatic proton cross-peak chemical shifts using *mkcol*, a list of $^3J_{H^N H^\alpha}$ coupling constants, a configuration file specifying the network architecture and an optimized weight matrix which is the result from network training using this particular network architecture. The chemical shift information, peak volumes together with $^3J_{H^N H^\alpha}$ values are combined, and converted to the desired neural network input format. *Pats* then computes the activation levels of output units from the input vector and the weight matrix which defines the correlation between input and output. The feed-forward algorithm is used in this procedure. The results are output as a list of possible amino acid classes for each test pattern sorted from highest to lowest probability.

Network design

Since we do not anticipate a linear response of the output units to the inputs, the use of at least one hidden layer in the chosen network architecture is necessary. Each unit in this hidden layer is connected with every unit in both the input layer and the output layer by an adjustable weight w_{ij}. The activations of input units are clamped by the externally supplied patterns representing input NMR parameters. In the feed-forward procedure, the activations of hidden units and output units are calculated from the activations of the units in the previous layer and the weights connecting them as follows:

$$\text{activation}_i = \frac{1}{1 + e^{-(\text{net})_i}}$$

$$(\text{net})_i = \sum a_j * w_{ij} + (\text{bias})_i$$

The bias_i term is set to zero in our case, but could be used to scale relative sensitivity to various types of data. A back-propagation learning procedure is used in the training process. An error function is defined as the sum of the squares of the differences between the target outputs and computed outputs, and a gradient descent algorithm is employed to make a change in the weight proportional to the negative of the derivative of the error function with respect to each weight. After each adjustment, the feed-forward procedure is repeated to recompute the outputs. The back-propagation and feed-forward procedures are iterated until the error between the computed output and target output is minimized. The optimized weights are then stored into a weight matrix.

Finding a proper presentation for input data is necessary for a network to perform well. In our networks, we retain the representation for proton cross peaks developed in our original work (Hare and Prestegard, 1994). The first 71 units are used to represent proton cross peaks from –1.0 to 6.0 ppm with a grid size of 0.1 ppm. The activation of an input unit is set to the sum of the volumes of the peaks whose chemical shifts happen to fall within that cell of the grid. The volumes used in this step have been normalized to make the sum of the volumes of all the proton cross peaks of a spin system equal to 1. The size of the grid cells is chosen so that it is fine enough to distinguish most cross peaks from the side-chain protons, and it is coarse enough to reflect the degree of scatter in proton chemical shifts due to factors other than secondary structure and residue type. Another input unit has been added to store $^3J_{H^N H^\alpha}$ information. To ensure that the activation is between 1 and 0, we define the input as the ratio of the measured $^3J_{H^N H^\alpha}$ and 14.0 Hz, since no $^3J_{H^N H^\alpha}$ is larger than 14.0 Hz. An HNHA experiment (Vuister and Bax, 1993; Kuboniawa et al., 1994) is chosen as the source of $^3J_{H^N H^\alpha}$ coupling constants because the spectrum is easy to analyze and it provides good accuracy. Similar to the representation for proton chemical shifts, nitrogen chemical shifts are binned into cells distributed over the 90.0–135.0 ppm region. The activation of each unit is either set to 1 if the nitrogen chemical shift value falls in that cell, or to 0 if not. In our tests, we varied the size of the cell from 7.0 to 3.0 ppm in an attempt to match the grid cell size to the statistical variation in the data.

One might anticipate the need for at least 21 output units for the 20 amino acid types and a single secondary structure indicator. However, many amino acids have almost indistinguishable proton coupling patterns and nitrogen chemical shifts. A large amount of training data would be needed for the network to learn to make these subtle distinctions. Thus, we have reduced the number of output units to 13 by combining glutamine with glutamic acid, lysine with arginine, asparagine with aspartic acid, and cysteine with all the aromatic amino acids. This yields 12 amino acid classes plus a secondary structure indicator. During the training, the activation for the unit representing the target amino acid class is set to 1 while others are set to 0. The activation for the output unit representing the secondary structure is set to 1 when the secondary structure is an α-helix, and to 0 otherwise.

Choosing an appropriate number of hidden units is extremely important. Using too few will deprive the network of the resources it needs to solve the problem (Mas-

ters, 1993). Using too many will result in overfitting or an inability to find an adequate number of training examples. The minimum number of hidden units can be estimated as follows: if we assume the activation level of the hidden units is binary (either 0 or 1), then the number of classes the network is able to recognize will be 2^n, where n is the number of hidden units. In our case, we can further assume that the activation level for output units is also binary and each class is represented by a different neuron. Thus, n hidden units are needed for 2^n output units, e.g. a total of 13 output units would require the use of at least four hidden units. An upper limit on the number of hidden units to be used is often practically set by the size of the training set. At a given training set size, too many hidden units would cause the network to learn the insignificant aspects of the training set which may not be representative of the general population. The optimum number of hidden units is therefore best determined empirically. In the network we tested, we varied the number of hidden units between five and eight.

Preparation of input data

The network design dictates that the input be in the form of a vector uniquely assigned to a residue, with the elements of this vector representing its proton cross peaks in a TOCSY-HSQC data set, its amide ^{15}N chemical shift and its $^3J_{H^NH^\alpha}$ coupling constant. To prepare these vector components, the intensity maximum of each cross peak must be located and peaks partitioned into sets such that each peak in a given set belongs to one and only one residue in the protein. Although manual separation is in principle quite straightforward, in practice this is time consuming and it can be difficult due to low sensitivity, spectral artifacts, overlap of resonances, and lack of digital resolution. The program *mkcol* is designed to make use of a peak list obtained from an HSQC data set, which can be acquired with high sensitivity, high resolution and few spectral artifacts, for resolving the ambiguities inherent in TOCSY-HSQC data sets.

We begin by generating a list of cross-peak positions using a standard peak-picking routine. For the 3D TOCSY-HSQC data, we have found the *peakpick* v. 1.0 routine of Chylla and Markley (University of Wisconsin, Madison, WI, U.S.A.) to be reliable, robust, and easy to use. The 2D HSQC data were picked interactively using the 2D peak-pick module of FELIX v. 2.3 (Biosym Technologies, San Diego, CA, U.S.A.). Both sets of peak information were read by the program *mkcol*. The TOCSY-HSQC peaks are classified by *mkcol* based on the proximity of their amide proton–nitrogen shift projections to a 2D HSQC peak. This information can be used both to eliminate spurious picks, as well as to provide initial guesses at the 'spin-system' partition. The TOCSY peak data are displayed graphically as a 2D projection of the 3D data. Information about the distance of a TOCSY cross peak from the nearest HSQC peak is conveyed by color coding of the peak marker, while its shape provides information about the spectral region of its indirect proton chemical shift (the dimension along which the projection is performed). The program then allows the user to interactively repartition any ambiguous cross peaks by visual inspection. This is particularly valuable in allowing the assessment of missing or excess α-resonance correlations. The resulting sorted peak information is then written to a file for input into the neural network training or analysis programs.

Training the network

The training set was composed of spin systems extracted from assigned heteronuclear TOCSY-HSQC spec-

TABLE 1
SAMPLE CONDITIONS AND ACQUISITION PARAMETERS OF TOCSY-HSQC SPECTRA USED FOR GENERATING NEURAL NETWORK INPUT VECTORS FOR TRAINING AND TESTING

Protein	Number of residues	Secondary structure	Spin systems	Buffer conditions	Mixing time (ms)	Reference
E. coli ACP	77	Four helices	66	pH 6.6, 30 °C, 100 mM NaOAc, 7 mM Ca^{2+}	60	Andrec et al. (1995); Holak and Prestegard (1986)
E. coli ACP	77	Four helices	59	pH 7.0, 25 °C, 100 mM NaOAc	60	As above
E. coli ACP	77	Four helices	61	pH 6.0, 25 °C, 100 mM NaOAc, 10 mM Ca^{2+}	60	As above
E. coli $DnaJ_{1-78}$	78	Four helices	52	pH 6.0, 30 °C, 50 mM phosphate	60	Hill et al. (1995)
E. coli $DnaJ_{1-103}$	103	Four helices	59	pH 6.0, 30 °C, 50 mM phosphate	60	K. Huang, J.M. Flanagan and J.H. Prestegard (in preparation)
BPTI	58	Antiparallel β-sheet, α-helix	48	pH 5.1, 25 °C, no salt or buffers	80	Wagner and Wüthrich (1982)
NodF	88	Four helices	73	pH 6.0, 25 °C, 50 mM phosphate	60	Ghose et al. (1996)
Ubiquitin	76	Five-stranded β-sheet, α-helix	50	pH 5.4, 25 °C, 10 mM OAc, 10 mM Ca^{2+}, 100 mM K^+	59	Weber et al. (1987); Wang et al. (1995)
Raf_{56-132}	76	Five-stranded β-sheet, α-helix	62	pH 7.2, 25 °C, 20 mM Tris, 5 mM Mg^{2+}, 50 mM Na^+	80	Emerson et al. (1994)

TABLE 2
DISTRIBUTION OF THE TRAINING DATA SET BY AMINO ACID TYPE AND SECONDARY STRUCTURE

Amino acid type	Percentage in the training data	Total number of spin systems	Number of spin systems in α-helices	Source of $^3J_{H^N H^\alpha}$ values[a]		
				HNHA	PDB	NOE
Alanine	11.1	52	28	36	6	0
Glycine	6.6	31	9	13	4	0
Serine	2.4	11	2	4	1	0
Valine	8.1	38	19	19	1	6
Threonine	3.9	18	3	11	1	0
Aspartic acid	6.2	29	15	9	2	1
Leucine	9.0	42	23	19	2	5
Isoleucine	7.3	34	28	23	2	1
Lysine	7.3	34	21	21	4	1
Glutamic acid	12.4	58	37	47	2	1
Methionine	1.1	5	4	4	1	0
Phenylalanine	3.0	14	4	4	4	1
Asparagine	3.6	17	5	10	3	1
Tyrosine	4.1	19	11	8	3	0
Histidine	1.5	7	3	7	0	0
Cysteine	1.7	8	1	0	5	1
Tryptophan	0.4	2	1	2	0	0
Arginine	5.4	25	13	7	6	5
Glutamine	4.7	22	5	9	1	1

[a] Those $^3J_{H^N H^\alpha}$ coupling constants which are unable to be determined and are assigned a 6 Hz value are not included here.

tra on various proteins collected under different conditions. Detailed descriptions of the spectral data sets used are listed in Table 1. The spectra were collected on either a GE Omega 500 MHz spectrometer or a Bruker 600 MHz spectrometer using pulse sequences described in the literature (Marion et al., 1989; Kay et al., 1992). The proton and nitrogen chemical shifts were referenced to DSS (Wishart et al., 1995) to ensure consistency of the chemical shift data among different spectra. A total of 456 spin systems were extracted from these spectra. Data on human ubiquitin were used as a test data set; thus, they were excluded from the training step. Also, only 46 out of 62 spin systems extracted from TOCSY-HSQC data on Raf_{56-132} were included in the training step because the rest were used in preliminary tests. $^3J_{H^N H^\alpha}$ coupling constants of residues in *E. coli* ACP, $DnaJ_{1-78}$, $DnaJ_{1-103}$, and NodF are from the analysis of HNHA spectra. In the case where the $^3J_{H^N H^\alpha}$ could not be determined due to resonance overlap or missing peaks, a 6.0 Hz value was assumed. $^3J_{H^N H^\alpha}$ coupling constants of residues in BPTI and human ubiquitin were calculated from PDB atomic coordinates since no data from an HNHA experiment were available. The coefficients in the Karplus equation used to convert ϕ angles into $^3J_{H^N H^\alpha}$ are the values of Vuister and Bax (1993). As for Raf_{56-132}, neither PDB coordinates nor HNHA experimental data are available. In this case, we set $^3J_{H^N H^\alpha}$ to 3.0 Hz for residues that seemed to be in α-helical regions based on NOE data, 9.0 Hz for residues that seemed to be in a β-sheet structure, and 6.0 Hz for residues that seemed to be in other types of secondary structure (Table 2).

The training set is significantly more diverse than that used in our preliminary work, where only data sets from *E. coli* ACP were used. All amino acid types in both α-helical and β-sheet secondary structures are represented in these proteins. Table 2 shows the distribution of our training data set by amino acid type and secondary structure. The distribution obviously varies a great deal from one amino acid type to another. The number of cross peaks in each spin system also varies due to different mixing time, proton–proton coupling constants, and relaxation properties (Table 3). The spin systems included in our training set represent all of those observed in the TOCSY-HSQC spectra except those with no aliphatic cross peaks or those with uncertainty in their assignments.

We were mostly concerned about two issues in the design of the training set. One is that the members of each class should be balanced in the training set. Otherwise, the network may strive to optimize its performance for the classes which have been overly represented, and perform poorly on other classes. The other issue is to avoid overfitting by presenting the network with enough data. The minimum training set size is suggested to be twice the total number of weights in the network, but, to improve the performance of the network, a doubling of the minimum size is advisable (Masters, 1993). For example, a network with 84 input units, 7 hidden units, and 13 output units would require around 2700 input patterns with a uniform distribution over possible classes. Clearly, the size of our training set is not large enough, and the distribution over amino acid type is not uniform.

The strategy we use to get around the disproportionate

distribution of amino acid classes and the insufficient number of patterns in the training set is the following: in each pattern, we vary each of the proton cross-peak chemical shifts by ±0.1 ppm and vary the nitrogen chemical shift by ±2.0 ppm. The $^{3}J_{H^{N}H^{\alpha}}$ value also is varied by ±1.0 Hz. Thus, each training pattern can produce tens or even hundreds of new patterns with a slight variation from the original pattern. The magnitudes of the above variations were chosen to simulate the scattered variations not correlated with amino acid class or secondary structure. For example, the magnitude of variation in nitrogen chemical shift is justified by the observation that the residue type and sequential effects of the preceding residue contribute up to 5 ppm to the dispersion of nitrogen chemical shift (Braun et al., 1994). Adding synthetic data in the training set in this way could bias the output of the network if the mean of the synthetic data deviates significantly from the mean of true experimental data. This problem is particularly acute in cases where specific amino acids are very poorly represented. The number of such acute cases is small in our data set.

After having generated training patterns, we separate them into 24 pools by amino acid class and secondary structure, e.g. alanines in an α-helix, alanines in a non-α-helical secondary structure, etc. Then 500 patterns are randomly picked from each pool and put together to be used as the training set. In the case where there are less than 500 patterns in a pool, all the patterns in that pool are picked. Patterns needed to make up the difference between 500 and the size of the pool are then randomly picked from the same pool. The total training set size approaches 12 000.

TABLE 3
DISTRIBUTION OF THE NUMBER OF CROSS PEAKS IN A SPIN SYSTEM IN THE TRAINING AND TEST DATA

Amino acid type	Training data						Test data			
	1	2	3	4	5	6	1	2	3	4
Alanine	4	48	0	0	0	0	0	2	0	0
Glycine	12	19	0	0	0	0	1	3	0	0
Serine	3	4	4	0	0	0	0	0	0	0
Valine	2	3	13	20	0	0	0	1	2	0
Threonine	6	8	4	0	0	0	0	0	0	0
Aspartic acid	1	5	23	0	0	0	0	2	2	0
Leucine	1	10	15	12	3	0	0	6	1	0
Isoleucine	2	3	13	13	0	0	0	3	2	0
Lysine	1	5	9	11	6	2	0	2	2	2
Glutamic acid	9	8	22	12	6	1	0	2	3	0
Glutamine	0	6	3	8	0	0	0	3	2	0
Methionine	0	2	0	3	0	0	0	0	0	0
Phenylalanine	0	7	6	1	0	0	0	0	2	0
Asparagine	0	8	9	0	0	0	0	0	2	0
Tyrosine	4	8	7	0	0	0	0	0	1	0
Histidine	0	4	3	0	0	0	0	0	1	0
Cysteine	0	1	6	1	0	0	0	0	0	0
Tryptophan	0	1	1	0	0	0	0	0	0	0
Arginine	4	2	7	9	3	0	0	1	1	1

Permuted training, in which the order of presenting each input to the network is varied from cycle to cycle, was used. The momentum value, which is the fraction of the previous weight increment incorporated in each new weight increment, was set to 0.9 to damp the side-to-side oscillation on the error surface. The learning rate, which scales the size of the changes made to the weights during optimization, was set to a small number 0.05 to allow proper convergence to a minimum. Using this procedure, a well-trained network was obtained after presenting the training set to the network 4000 times (epochs). Training was a time-consuming step and required about 48 h of CPU time on a Silicon Graphics Iris INDY workstation with a 150 MHz R4400 processor (97.5 SPECfp92). The actual application of a trained network to amino acid class prediction using the program *pats* took an insignificant amount of CPU time.

Results and Discussion

The various trained networks were tested on 50 spin systems extracted from a TOCSY-HSQC data set on human ubiquitin and 15 spin systems extracted from a TOCSY-HSQC data set on Raf_{56-132}. Since the sampling of methionine, threonine and serine were clearly inadequate, we did not include these three amino acid types in our test data. We also excluded spin systems with only one cross peak and with a nitrogen chemical shift greater than 115.0 ppm (non-glycines), because these spin systems contain too little information to expect reasonable amino acid classification. With more extensive training, it may not be necessary to exclude these cases. The test results for two different versions of our new network (network I and network II), using between 5 and 8 hidden units and various numbers of training epochs, are shown in Table 4. The primary difference between the two network architectures is in the number of units used to represent ^{15}N chemical shifts. In network I, 5 input units were used to represent nitrogen shift values from 90.0 to 135.0 ppm with a grid size of 7.0 ppm, while in network II 12 input units were used to represent nitrogen shift values with a grid size of 3.0 ppm. Table 4 also shows the test results on a third network (network III), which was the previously optimized network architecture for predicting amino acid type using only aliphatic cross peaks (Hare and Prestegard, 1994). The same training data and test data were used on all three networks.

Both network I and network II show a pronounced improvement over network III in an ability to correctly predict amino acid class. While the architecture of network III did quite well when trained on data from a highly helical protein and tested on spin systems from a homologous protein, it seems that this architecture is not good enough to handle more varied input data. There is also a substantial difference in the performance by net-

TABLE 4
TEST RESULTS FOR THE NETWORKS USED TO IDENTIFY AMINO ACID TYPE AND SECONDARY STRUCTURE

Neural network	Hidden units	Epochs	Percentage of correct amino acid type prediction (%)		Percentage of correct secondary structure prediction (%)
			1st choice	1st/2nd	
I	5	4000	35	54	75
	5	8000	43	54	74
	6	4000	35	52	78
	7	4000	48	65	80
II	5	4000	23	31	86
	5	8000	26	37	97
	6	4000	32	45	94
	6	6000	35	46	95
	7	2000	29	51	94
	7	4000	35	62	94
	7	6000	37	55	94
	7	8000	34	55	94
	8	4000	28	45	82
	8	6000	32	52	85
	8	8000	35	57	85
III	4	2000	8	29	N/A

work I and network II in their ability to correctly predict secondary structure. Network II, which has a finer grid representing nitrogen chemical shifts, does much better. Apparently, the useful secondary structure information in the nitrogen shifts cannot be utilized if resolution is too low. Hence, network II seems to have the best architecture for our purpose. Within both network I and network II, the number of hidden units makes a difference in network performance. In general, increasing the number of hidden units improves the performance of the networks. For network II, there is a marginal improvement when the number of hidden units is increased from 5 to 6 and from 6 to 7. However, the performance, particularly for secondary structure prediction, deteriorates when 8 hidden units are used. On the one hand, this may result from the overinterpretation of data in our relatively small training set. On the other hand, it may result from an inadequate number of presentation epochs with the large number of weights in this network. Note that we do see a noticeable improvement in its performance when the number of epochs is increased from 6000 to 8000, the largest number of epochs used. Since training with more than 8000 epochs is impractical, we have decided to use network II with 7 hidden units. With amino acid class prediction being more than 60% correct and secondary structure prediction exceeding 90%, this network structure seems quite viable.

A closer look at those test spin systems which are not correctly recognized by the network sheds light on how we may further improve the performance of this network. Three major causes of failure can be identified. First, there are cases where ambiguity may arise from extra peaks in spin systems. Ile^{3} in ubiquitin, for example, is classified as a serine. A careful manual inspection of the spectrum reveals that this spin system has only two strong peaks at 4.15 and 3.81 ppm with very slight differences in the nitrogen chemical shifts. This suggests that these two peaks may come from Ile^{3} in two different molecular forms, possibly due to N-terminal modification. Thus, these two peaks should have been sorted into two separate spin systems. Clearly, an improved spin system sorting program would reduce such mistakes. Secondly, quite a few other spin systems belonging to amino acids with long side chains only contain α- and β-proton cross peaks due to a poor TOCSY transfer. The input information clearly is not adequate for the network to distinguish these from other amino acid types. Leu^{43}, for example, is mistakenly classified as an arginine or a lysine because it only has two cross peaks at 5.33 and 1.56 ppm. A third misclassification of amino acid type results from unusual chemical shift values that may reflect unique secondary or tertiary environments. The methyl peak in Ala^{46} in ubiquitin, for example, is at 0.87 ppm while the average chemical shift of the methyl peaks of alanines in the training set is about 1.4 ppm. Therefore, it is not surprising that the network predicts this spin system to be a valine instead of an alanine.

We expect the performance of the network will be improved if more real data are used to replace the simulated input patterns, and a greater variety of spin system patterns for a given amino acid class is presented to the network. However, difficult cases, such as those discussed above, will persist. It is probably fair to say that with no additional information, no other assignment strategy would perform better than the neural network method in those cases. A very convenient source of additional information could be carbon chemical shifts from intraresidue α- and β-carbon cross peaks in an HNCACB spectrum since α- and β-carbon chemical shifts strongly correlate with both amino acid type and secondary structure (Grze-

siek and Bax, 1993). An approach employing these data may, however, be restricted to proteins showing high levels of expression in minimal medium.

The above results do, however, show that the neural network is a useful algorithm for determining amino acid class and secondary structure from NMR data. Although both nitrogen chemical shifts and α-proton chemical shifts are influenced strongly by amino acid class and secondary structure, the network is able to decouple the amino acid class information from that of secondary structure, and make useful predictions. It is also significant that the additional data have actually improved the performance over our preliminary network designed for identification of amino acid class only. The number of spin systems whose amino acid class is correctly predicted in the top two choices by a retrained network III is only 29%. This shows that instead of being confused by the complexity of the amino acid class information in α-proton and nitrogen chemical shift values, network II is able to make use of the new information to improve its ability to classify amino acids. The 60% occurrence of the correct amino acid class in the top two choices is in fact quite impressive given that 52% of the spin systems examined contain less than three cross peaks. The reduced ambiguity in amino acid class prediction is critical to the acceleration of the next step in sequential assignment. Furthermore, the secondary structure information given by network II can be useful in anticipating possible NOE connectivity patterns.

The programs as currently described are available by anonymous ftp at glyco.chem.yale.edu. Utilization of amino acid class assignments and secondary structure prediction from a trained neural network in the second step of sequential assignment is under way.

Acknowledgements

We thank Blake Hill, Joel Tolman, and Ranajeet Ghose for supplying assigned spectra for the training of the network. This work was supported by a Charles Goodyear Cooperative Research and Development Grant administrated by Connecticut Innovations Inc.

References

Andrec, M., Hill, R.B. and Prestegard, J.H. (1995) *Protein Sci.*, **4**, 983–993.

Bartels, C., Xia, T.-h., Billeter, M., Güntert, P. and Wüthrich, K. (1995) *J. Biomol. NMR*, **6**, 1–10.

Braun, D., Wider, G. and Wüthrich, K. (1994) *J. Am. Chem. Soc.*, **116**, 8466–8469.

de Dios, A.C., Pearson, J.G. and Oldfield, E. (1993) *Science*, **260**, 1491–1496.

Emerson, S.D., Waugh, D.S., Scheffler, J.E., Tsao, K.L., Prinzo, K.M. and Fry, D.C. (1994) *Biochemistry*, **33**, 7745–7752.

Friedrichs, M.S., Mueller, L. and Wittekind, M. (1994) *J. Biomol. NMR*, **4**, 703–726.

Ghose, R., Geiger, O. and Prestegard, J.H. (1996) *FEBS Lett.*, **388**, 66–72.

Glushka, J., Lee, M., Coffin, S. and Cowburn, D. (1989) *J. Am. Chem. Soc.*, **111**, 7716–7722.

Glushka, J., Lee, M., Coffin, S. and Cowburn, D. (1990) *J. Am. Chem. Soc.*, **112**, 2843.

Grzesiek, S. and Bax, A. (1993) *J. Biomol. NMR*, **3**, 185–204.

Hare, B.J. and Prestegard, J.H. (1994) *J. Biomol. NMR*, **4**, 35–46.

Hill, R.B., Flanagan, J.M. and Prestegard, J.H. (1995) *Biochemistry*, **34**, 5587–5596.

Holak, T.A. and Prestegard, J.H. (1986) *Biochemistry*, **25**, 5766–5774.

Karplus, M. (1959) *J. Chem. Phys.*, **30**, 11–15.

Kay, L.E., Keifer, P. and Saarinen, T. (1992) *J. Am. Chem. Soc.*, **114**, 10663–10665.

Kuboniwa, H., Grzesiek, S., Delaglio, F. and Bax, A. (1994) *J. Biomol. NMR*, **4**, 871–878.

Le, H. and Oldfield, E. (1994) *J. Biomol. NMR*, **4**, 341–348.

Marion, D., Driscoll, P.C., Kay, L.E., Wingfield, P.T., Bax, A., Gronenborn, A.M. and Clore, G.M. (1989) *Biochemistry*, **28**, 6150–6156.

Masters, T. (1993) *Practical Neural Network Recipes in C++*, Academic Press, New York, NY, U.S.A.

McClelland, J.L. and Rumelhart, D.E. (1988) *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, MA, U.S.A.

Meadows, R.P., Olejniczak, E.T. and Fesik, S.W. (1994) *J. Biomol. NMR*, **4**, 79–96.

Mittard, V., Morelle, N., Brutscher, B., Simorre, J.-P. and Marion, D. (1995) *Eur. J. Biochem.*, **229**, 473–485.

Neidig, K.-P., Geyer, M., Goerler, A., Antz, C., Saffrich, R., Beneicke, W. and Kalbitzer, H.R. (1995) *J. Biomol. NMR*, **6**, 255–270.

Olson Jr., J.B. and Markley, J.L. (1994) *J. Biomol. NMR*, **4**, 385–410.

Oschkinat, H., Holak, T.A. and Cieslar, C. (1991) *Biopolymers*, **31**, 699–712.

Van de Ven, F.J.M. (1990) *J. Magn. Reson.*, **86**, 633–644.

Vuister, G.W. and Bax, A. (1993) *J. Am. Chem. Soc.*, **115**, 7772–7777.

Wagner, G. and Wüthrich, K. (1982) *J. Mol. Biol.*, **155**, 347–366.

Wang, A.C., Grzesiek, S., Tschudin, R., Lodi, P.J. and Bax, A. (1995) *J. Biomol. NMR*, **5**, 376–382.

Weber, P.L., Brown, S.C. and Mueller, L. (1987) *Biochemistry*, **26**, 7282–7290.

Wishart, D.S., Sykes, B.D. and Richards, F.M. (1991) *J. Mol. Biol.*, **222**, 311–333.

Wishart, D.S., Bigam, C.G., Yao, J., Abildgaard, F., Dyson, J.H., Oldfield, E., Markley, J.L. and Sykes, B.D. (1995) *J. Biomol. NMR*, **6**, 135–140.

Wüthrich, K. (1986) *NMR of Proteins and Nucleic Acids*, Wiley, New York, NY, U.S.A.

Zimmerman, D., Kulikowski, C., Wang, L., Lyons, B. and Montelione, G.T. (1994) *J. Biomol. NMR*, **4**, 241–256.

Zimmerman, D. and Montelione, G.T. (1995) *Curr. Opin. Struct. Biol.*, **5**, 664–673.