Protein Engineering vol.6 no.4 pp.383–390, 1993

Evaluation of secondary structure of proteins from UV circular dichroism spectra using an unsupervised learning neural network

M.A.Andrade, P.Chacón, J.J.Merelo[1] and F.Morán

Departamento de Bioquímica y Biología Molecular I, Facultad de Ciencias Químicas, Universidad Complutense de Madrid, 28040 Madrid and [1]Departamento de Electrónica y Tecnología de Computadores, Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain

An optimized self-organizing map algorithm has been used to obtain protein topological (proteinotopic) maps. A neural network is able to arrange a set of proteins depending on their ultraviolet circular dichroism spectra in a completely unsupervised learning process. Analysis of the proteinotopic map reveals that the network extracts the main secondary structure features even with the small number of examples used. Some methods to use the proteinotopic map for protein secondary structure prediction are tested showing a good performance in the 200–240 nm wavelength range that is likely to increase as new protein structures are known.

Key words: neural networks/prediction/secondary structure/unsupervised learning

Introduction

The knowledge of the secondary structure of a protein has great importance in the study of protein functionality. The structures of some hundreds of proteins have been completely resolved by analysing the X-ray diffraction patterns of the crystallized molecule. However, crystallization of a protein is a difficult and not always feasible task. Therefore, techniques for prediction of secondary structure from more readily measurable protein characteristics have been developed.

A first approach estimates protein structure from amino acid sequence, since the secondary structure of a protein region can be taken exclusively as a function of its amino acid composition. In turn, these methods can be subdivided into those which use only statistical methods and those which incorporate physico-chemical theory.

Some other methods have been proposed to estimate the secondary structure of a protein in solution. They are based on the dependence of the optical activity of proteins between 170 and 240 nm on the peptide chain, with almost no influence of the side chains (except for some contributions of the aromatic amino acids) (Hennessey and Johnson, 1981; Manavalan and Johnson, 1983; Perczel *et al.*, 1991). The problem is finding the correspondence between the circular dichroism (CD) spectra and the percentages of secondary structure. Classically, the spectrum of a protein is assumed to be the result of the addition of the effects produced by regions with different secondary structure conformations (e.g. α-helix, β-sheet and random coil).

There are statistical methods that compute the secondary structure following the latter model (for a review see Yang *et al.*, 1986). Some of them use linear combinations of reference spectra of proteins, with 100% α-, β- or random structure, measured from model polypeptides. Others use linear combinations of spectra of proteins whose secondary structure is known from X-ray diffraction patterns analysis (Hennessey and Johnson, 1981; Provencher and Glöckner, 1981; Manavalan and Johnson, 1987; Menéndez-Arias *et al.*, 1988; van Stokkum *et al.*, 1990). Although these methods are fairly successful and their use widely extended, they sometimes give negative percentages of β-structure or predict dissimilar structures for quite similar spectra. The assumption of a linear summation of the fragment spectra is not completely correct: the CD values of a polypeptide do not only depend on the relative quantities of structure, but also on the length of the chain segments with different secondary structures; also, the interactions between amino acids far off in the sequence (tertiary structure interactions) influence the CD values.

The failure of the classical statistical methods suggests the use of non-linear methods like neural network algorithms that are able to perform learning from examples and to generalize from the learned data.

Neural networks have become an increasingly used tool in the field of protein structural and functional analysis (see a recent review of Hirst and Sternberg, 1992).

Neural networks with back-propagation learning have been shown to be useful in obtaining mapping between sequence and both protein secondary structure (Qian and Sejnowski, 1988; Holbrook *et al.*, 1990; Kneller *et al.*, 1990) and protein tertiary structure (Bohr *et al.*, 1990), in recognizing other protein characteristics (Holley and Karplus, 1989; Muskal *et al.*, 1990; Fessenden, 1991) and for protein homology analysis (Bohr *et al.*, 1988; Petersen *et al.*, 1990).

Recently Böhm *et al.* (1992) have used back-propagation learning from CD spectra of proteins in dissolution for secondary structure prediction purposes, improving the previous results in the calculation of the structure. Nevertheless, this kind of network could not have a generalization capability in the calculation of other proteins since the number of connections in the network exceeded by far the advisable empirical ratio examples/connections for this kind of network (Rumelhart and McLelland, 1988).

Since in this problem only a small number of examples (proteins) is available and there is a great amount of information for every example (data spectra values), an unsupervised learning algorithm like Kohonen's self-organizing map (SOM) (Kohonen, 1982, 1986; Kohonen *et al.*, 1984) seems to be more appropriate. Kohonen developed this algorithm, inspired by the self-organization of the topological maps of the sensorial nervous system during the development of an animal. It compresses a training set of high-dimensional vectors to low-dimensional ones arranging the set of vectors on a map. This arrangement depends on the features implicit in the set of training vectors that the network is able to extract. This algorithm has been used to classify proteins using either protein sequences (Ferrán and Ferrara, 1991) or protein ultraviolet CD spectra (Merelo *et al.*, 1991a,b). In the latter case, the SOM algorithm main parameters were optimized to obtain maximum efficiency and the extracted features were strongly correlated with three types of secondary structure.

In this paper, the term proteinotopic mapping is introduced to design the classification of proteins in a bidimensional map.

The optimized SOM algorithm is used proving the invariance of the proteinotopic mapping previously described (Merelo *et al.*, 1991a,b). Several methods to evaluate these maps are proposed. From this evaluation the secondary structure map corresponding to a concrete proteinotopic map is obtained. It allows the prediction of the structure of problem proteins not included in the training set used to form the proteinotopic map. Finally, in order to compare the SOM methods to other methods, they were tested with several sets of example proteins, making several maps for every set and finding a significant invariability in the deduced structure for each problem protein. In addition, this method calculates a theoretical spectrum according to the prediction and gives an estimation of the error in the values of the determined secondary structure.

Materials and methods

Architecture of the network

We consider a network proposed by Kohonen to perform a dimensionality reduction of the input signal patterns (Kohonen, 1990). In this paper, each input signal corresponds to one protein spectrum, an n-dimensional vector ($\vec{X}_k$ for $k = 1, \ldots, s$, s being the total number of pattern samples of the training set) whose components are the CD values at each wavelength. We want to reduce a spectrum vector to a structure vector, whose components are the α-, β- and random percentage values. Actually, this vector has only two independent components since the sum of the three components is one.

The network has an input layer consisting of n neurons, one for each of the corresponding components of the input vectors and a second layer consisting of a lattice of $m \times m$ neurons. The input layer is connected to every lattice neuron (N_{ij} for i, $j = 1, \ldots, m$). The connections from the input layer to an N_{ij} neuron are described by an n-component weight vector, $\vec{W}_{ij}$.

In this work, the input vector has 41 components corresponding to CD spectra values for wavelengths from 200 to 240 nm. Data above 240 nm is not used, since in this part of the spectra there is no significant contribution of the peptide bond. But, the limitation of the analysis of the CD data to values above 200 nm could be discussed since it has been shown (Toumadje *et al.*, 1992) that extending the analysis down to 168 nm shows an improvement in the prediction allowing the determination of different β-structures.

However, most of the functional protein studies use physiological media in order to measure, for example, conformational changes associated to either activity-changes or other functional characteristics. These experimental conditions, frequent in a biochemistry laboratory, do not allow access to CD data below 200 nm due to media absorption. Therefore, a method that does not need low wavelength CD values seems to be worthy. Classical statistical methods fail when this part of the spectrum (185−200 nm range) is not used.

Hence, the 200−240 nm range is not reliable for the determination of different β-structures. According to Manavalan and Johnson (1985) and Perczel *et al.* (1991) the information content of the spectrum truncated at 200 nm allows the calculation of only three secondary structure fractions (i.e. it only contains two independent variables). This circumstance has led to the use of a neural network that makes a two-dimensional mapping of the features of the CD spectra used as an example.

As a training set, 24 CD input vectors have been used [see Yang *et al.* (1986) for references of the spectra and the determination of the secondary structure values from the X-ray results]. Eighteen of them correspond to proteins whose secondary structure is known. Three are the spectra of a synthetic polypeptide, poly(L-lysine), whose CD spectra at different pH and temperature in aqueous solution is used as a model system for α-, β and random conformation. The remaining three are the reference spectra of α-, β- and random coil conformation structures based on 15 proteins of known structure taken from Chang *et al.* (1978).

The network is able to interpolate among the given spectra, but not to extrapolate. So, a complete structure map could not be obtained unless spectra of pure secondary structure are included in the training set.

A square lattice of 13 × 13 neurons was used (following Merelo *et al.*, 1991a,b).

Learning rule

The weight vectors are initialized with small random values and may take continuous values. Each time an input pattern ($\vec{X}_k$) is presented to the network, a winning lattice neuron, i.e. the one whose weight vector $\vec{W}_{ij}$ is the closest to $\vec{X}_k$, is chosen. Then, the weight vectors of both the winning neuron and its neighbourhood are updated making them closer to $\vec{X}_k$ with the following rule:

$$\vec{W}_{ij}(t+1) = \vec{W}_{ij}(t) + \alpha(t)[\vec{X}_k - \vec{W}_{ij}(t)] \qquad (1)$$

where $\alpha(t)$ is a time-dependent parameter that is decreased to impose convergence on the weights. Its value is calculated in the following way:

$$\alpha(t) = \begin{cases} \alpha_0 - k_1 t, & \text{for } 0 < t < t_1 \\ \alpha_0 - k_1 t_1, & \text{for } t_1 < t < t_2 \end{cases} \qquad (2)$$

where k_1 is a constant that describes how fast α is decreased and α_0 is the initial α-value.

For the neighbourhood of a lattice neuron, a square region of the lattice centred in that neuron is taken. At the beginning of the self-organizing process, the side of this square region is taken to be half the lattice side. The side of this square is linearly decreased to one from $t = 0$ to $t = t_1$. The learning ends at $t = t_2$.

In each step of the algorithm all examples are successsively presented to the network. Since there are many more lattice neurons than samples, as time progresses, a self-organizing process occurs and clusters of neurons tending to equal their weight vectors to one of the spectrum vectors appear.

To test the evolution of the self-organization of the map the distortion parameter (D) is used. This parameter is defined as the sum of the distances from every input vector to the weight vector of its corresponding winning neuron (Merelo *et al.*, 1991a). The logarithmic decrease of the D value indicates that a healthy self-organization process is taking place.

In this case, since the initial weight values are taken in a random fashion, different runs of the algorithm yield different proteinotopic maps. However, a local similarity could be observed, i.e. proximal neighbour relationships are maintained. The most striking result is that neurons of different corners of the lattice approximate their weight vectors to the six spectrum samples corresponding to pure structures and to those examples of proteins with high values of one of the structures: one corner to high α-values, another one to high β-values and another one to high random coil values. Therefore, it can be assumed that the weights of a given neuron at the end of the learning process must be some function of the structures of many of the data

proteins. A problem arises: how can the secondary structure map corresponding to a given proteinotopic map be made?

Evaluation of the map

A structure map can be made, assigning to each neuron the structure percentages of the sample protein whose CD spectra is the closest to the weight vector of that neuron (see an example in Figure 2a). The map obtained clearly shows how the neurons of the corners tend to point to proteins with extreme secondary structure values and how close neurons tend to point to proteins with similar secondary structure. The prediction of the structure percentages of a problem protein not included in the training set is made taking the structure values of the neuron which has the closest weight vector to the CD spectra of that problem protein.

In this case, the structure values of the sample protein closest to this neuron are taken. Nevertheless, it should be taken into account that a neuron could have come to an intermediate situation, in which that neuron had not chosen one example protein or another, reaching a compromise situation. Therefore, the prediction of the structure of a problem protein could be improved if the participation of more than one protein example in the structure values pointed by a lattice neuron is considered. Then, the expressions to compute the map values (α_i, β_{ij}, r_{ij}) of a given neuron N_{ij} for a set of q sample proteins are the following

$$a_{ij} = \sum_{k=1}^{n} p_k\alpha_k, \quad b_{ij} = \sum_{k=1}^{n} p_k\beta_k, \quad r_{ij} = \sum_{k=1}^{n} p_k R_k \quad (3)$$

where p_k stands for the participants of each sample protein in the characteristics of the considered neuron and α_k, β_k and r_k are the secondary structure percentages of the kth example protein.

The normalization of the coefficients a_{ij}, b_{ij} and r_{ij} gives the structure values

$$\alpha_{ij} = a_{ij}/s, \quad \beta_{ij} = b_{ij}/s, \quad R_{ij} = r_{ij}/s \quad (4)$$

where $s = a_{ij} + b_{ij} + r_{ij}$.

Several methods to calculate, the p_k values for every N_{ij} can be proposed. One of these methods is to take the structure values corresponding to the closest CD example, which can be described as

$$p_k = \begin{cases} 1 \text{ for the closest protein} \\ 0 \text{ otherwise} \end{cases}$$

But other methods could be defined, namely, the distance method and the rank method. In both, for a considered neuron with a concrete weight vector, the proteins are ordered depending on the euclidian distance of their CD spectra to that weight vector.

(i) Distance method. The structure values of those l closest proteins weighed with their distance are considered and their p_k values are:

$$p_k = \begin{cases} 1/d_k \text{ for the } l \text{ closer proteins} \\ 0 \text{ otherwise} \end{cases}$$

d_k being the euclidian distance between the weight vector and the kth protein vector.

(ii) Rank method. The structure values of the l closer proteins are weighed with their rank order and the p_k values are

$$p_k = \begin{cases} 1/n_k \text{ for the } l \text{ closer proteins} \\ 0 \text{ otherwise} \end{cases}$$

where n_k is the rank order of the sample CD spectrum $\vec{X}_k$ as compared with the weight vector of the considered neuron.

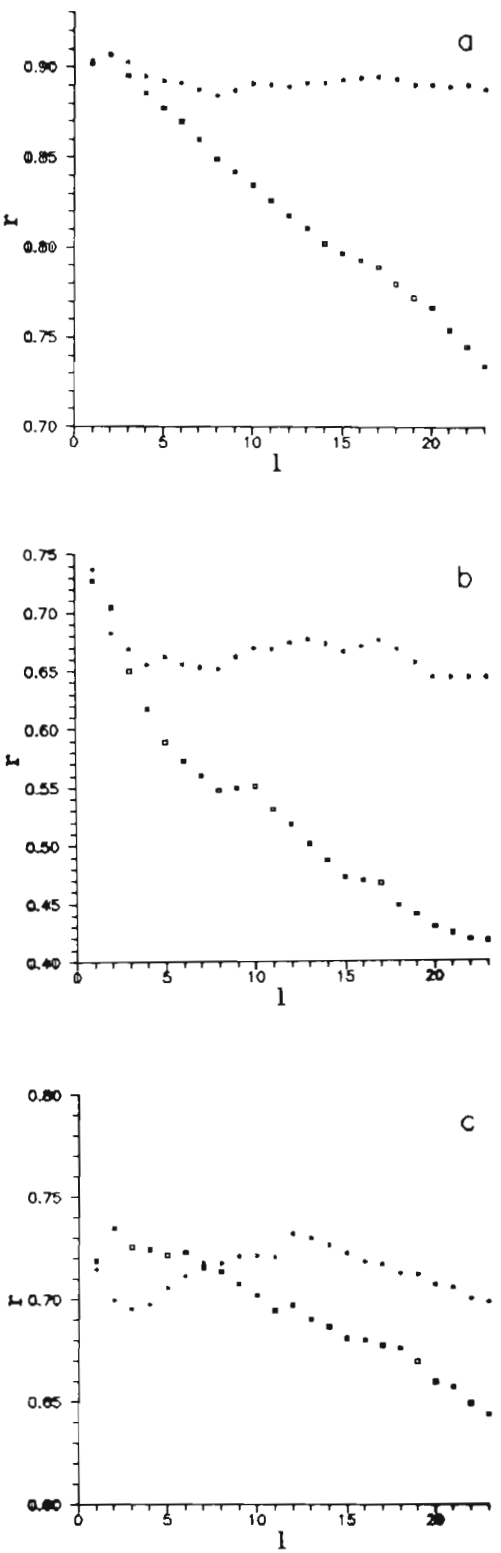

Fig. 1. Representation of the Pearson correlation coefficient, r, versus the scope of the method, l, for the distance method (□) and the rank metod (*). Each point corresponds to the mean over a 100 estimations of the training set structures. Correlations for the calculated (**a**) α-, (**b**) β- and (**c**) random values are shown. The Pearson correlation coefficient is defined as

$$r = [\sum X_i Y_i - \sum X_i \sum Y_i]/\{[\sum X_i^2 - (\sum X_i)^2/n] \times [\sum Y_i^2 - (\sum Y_i)^2/n]\}^{1/2}$$

where X_i and Y_i are the experimental and calculated values respectively and n is the number of samples studied.

The parameter l describes the scope of the method. For example, the two proposed methods are the same for $l = 1$, since this implies that only the closest protein is considered in the evaluation of the structure pointed out by a given neuron. In principle, with a large l value more information is extracted from the proteinotopic map, as more sample proteins are considered. On the other hand, the increment of the l value has a smoothing effect, which is more pronounced at the extremes of the map. The loss of the extremes is critical, since the estimation of the structures of problem proteins close to these extreme values will be less accurate.

Results

The effect of the l value in both the distance and rank methods has been described in Figure 1. There, the accuracy of these methods in the prediction of the structure of problem proteins of known secondary structure (but obviously not included in the sample set) is represented versus the l value. The influence of increasing l is stronger in the distance method than in the rank method, since the d_k values for the further proteins are quite similar and the whole group of sample proteins takes a great part in the composition of the different p_k, leading to flat maps

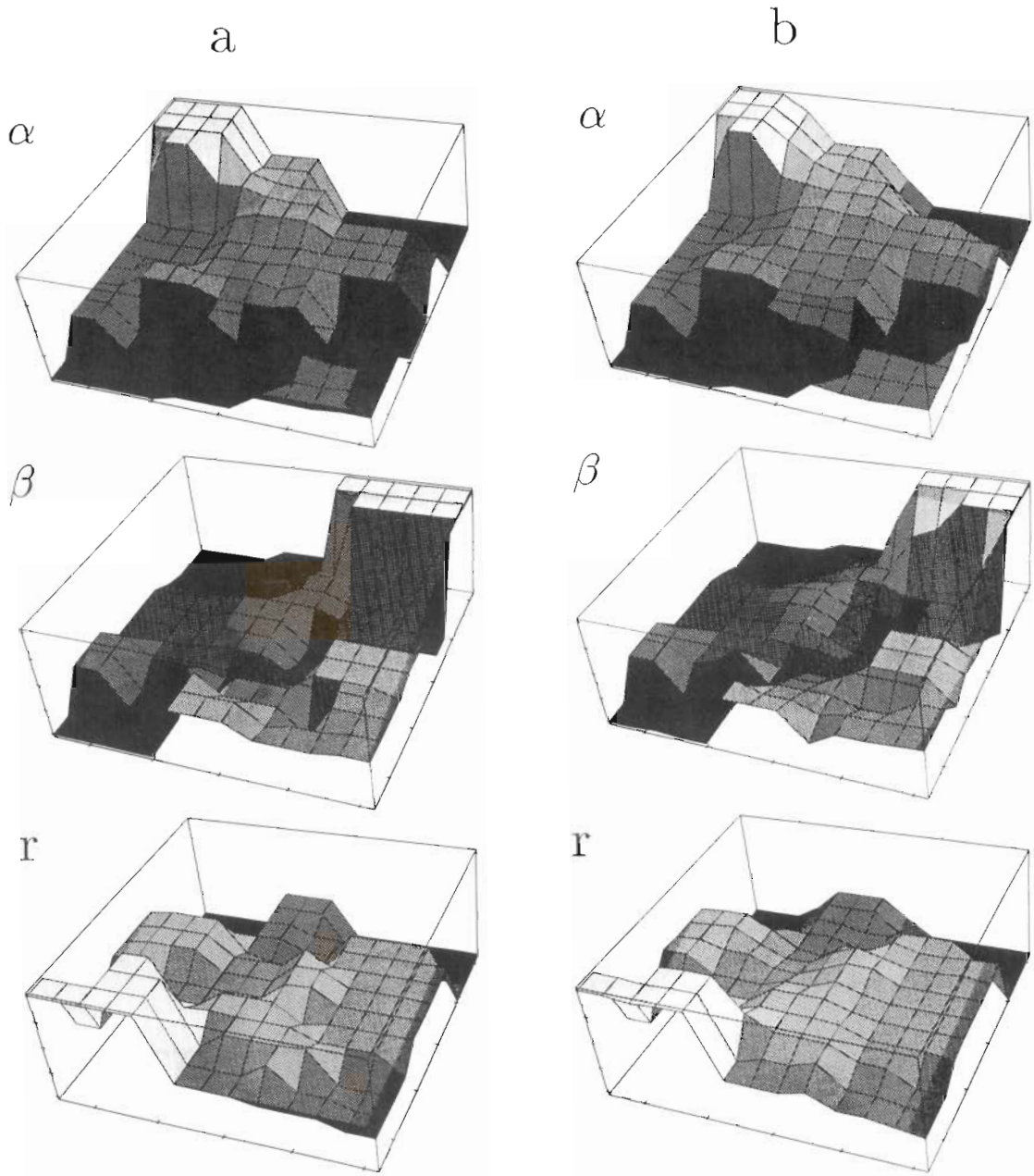

Fig. 2. A proteinotopic map was made, training the network with the set of 24 spectra except that of the myoglobin. (**a**) The three structure maps made from the proteinotopic map, assigning to each neuron the structure percentage values of the closest protein. The height of the surface respresents the structure value and the x and y axes stand for the lattice neurons. The neurons of the upper-left corner have high α-values, those of the upper-right corner have high β-values, and those of the lower-left corner have high random values. (**b**) Estimated structure maps made from the same proteinotopic map using the distance method with $l = 2$.

useless for estimation tasks. In general, the distance method has better performance than the rank method. For the former method an optimal l value can be empirically observed ($l \simeq 2$). Hereafter this method will be used.

An example of the application of the distance method is shown in Figure 2b. The main features of the maps with $l = 1$ are maintained, but the sharp points have been smoothed out.

In order to illustrate how the method works let us suppose that we want to estimate the structure values of the myoglobin ($\alpha = 0.79$, $\beta = 0.21$ and $r = 0.00$). First, the network is trained with the set of protein examples (except the myoglobin spectrum). Then the neuron whose weight vector is the closest to the myoglobin spectrum vector ($N_{10,3}$) is chosen. The myoglobin spectrum and the winning neuron weight vector are represented in Figure 3 with squares and with a continuous line respectively. The proteinotopic map formed in the learning process has a neuron with a weight vector close to that of the myoglobin. This weight vector has been obtained from interpolation among the learned examples. There are no such similar spectra in the training set. The closest sample spectrum to that neuron is the pure α-reference spectrum (represented in Figure 3 by a long-dashed line). Note that this spectrum is very different from the myoglobin spectrum. Then, using the map represented in Figure 2a the structure values of that spectrum ($\alpha = 1.00$, $\beta = 0.00$ and $r = 0.00$) would be assigned to myoglobin, obtaining considerable erroneous structure values.

The results can be improved by making the map taking into account two sample proteins instead of one ($l = 2$). In this example, the second closest spectra to the $N_{10,3}$ neuron is that of the parvalbumin (represented in Figure 3 with a short-dash line). The distances of these two spectra to the weight vector of the $N_{10,3}$ neuron give the normalized coefficients $p_1 = 0.52$ and $p_2 = 0.48$ respectively. Now, the structure values pointed out by the $N_{10,3}$ are $\alpha = 0.82$, $\beta = 0.02$ and $r = 0.16$ and thus these values would be assigned to the myoglobin. The interpolated spectrum between the pure α-reference and parvalbumin spectra

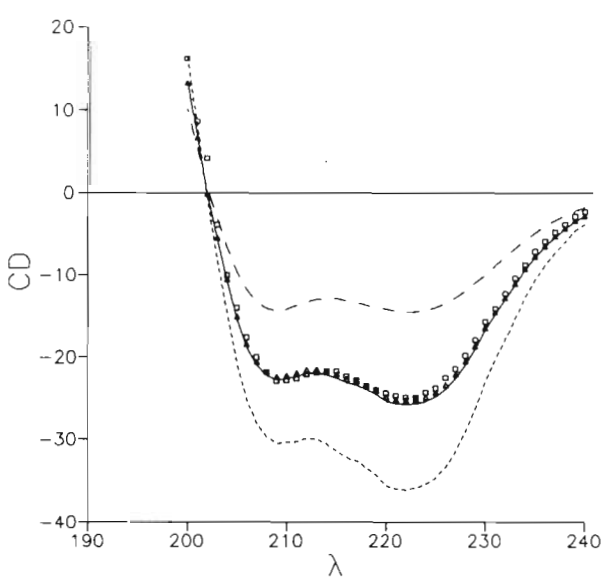

Fig. 3. CD spectrum of myoglobin (—). Weight vector of the winning neuron for myoglobin in the proteinotopic map calculated for Figure 2 (□). CD reference spectra of α-conformation (---), parvalbumin(·– – –) and the interpolated spectrum among these (△).

(represented in Figure 3 with triangles) is calculated using the p_k values:

$$[\theta](\lambda) = \sum_{k=1}^{2} p_k [\theta]_k(\lambda) \qquad (5)$$

where $[\theta]_k(\lambda)$ is the CD spectrum of a sample protein k. Although the spectra of the more similar sample proteins and that of the interpolated spectrum are very different to those of both the winning neuron and the problem protein, the resulting structure values are near the real ones.

In Figure 4, the spectra of the winning neuron in the estimation of each of the 18 proteic examples are shown. The spectra are the means for 20 algorithm runs. The problem protein was excluded from the training set in all cases. The worst results are obtained with the trypsin inhibitor, concanavalin A and carboxypeptidase A.

The results of the distance method, with $l = 2$ in the estimation of the 18 proteic examples, are shown in Table I. The accuracy is quite different from one protein to another. The standard deviation values are shown only to indicate that the method is quite invariable (even when it fails). The worst calculated percentages are those of the concanavalin A, elastase and cytochrome c.

One important result comes up from Figure 4 and Table I and it is explained through Figure 5. There, the mean absolute error in the three structure calculated values is plotted, against the square distance from the real spectra to the winning neuron. It can be observed that in spite of the dispersion of the points, it is possible to find several distance thresholds that allow us to determine a maximal error value (e.g. if the square distance is below 66 then the absolute error is below 0.12). However, large distances do not necessarily mean bad estimations. For example, the estimation of the papain structure is quite good (see Table I) but the calculated spectrum is rather poor (see Figure 4f).

The representation of Figure 5 is the key to determining the reliability of new predictions. Assuming that the behaviour of the method is going to be similar in the prediction of new protein secondary structure percentages to that in the 24 training spectra, the representation of Figure 5 is supposed to allow the assignation of a maximal error value for several ranges of distances that would not be surpassed in the prediction of any new problem protein. So, in every prediction the algorithm not only gives the secondary structure values, it also estimates the maximal error of the prediction, depending on the distance between the weight vector of the winning neuron and the CD spectrum of the problem protein.

The method itself can give hints about the accuracy of the prediction. Good enough spectra (small distances) provide good estimation, as shown in Figure 5. This error–distance relationship depends exclusively on the characteristics of the algorithm used (learning parameters, network geometry and size, estimation parameters) but it is independent of the set of examples (number and quality of the examples, wavelength range, wavelength step).

Discussion

In this paper a new method for protein secondary structure quantification has been presented. The term proteinotopic mapping has been introduced in reference to the classification of proteins in a map.

Firstly, using an optimized self-organizing map algorithm, a proteinotopic map is calculated from a set of CD spectra of pro-

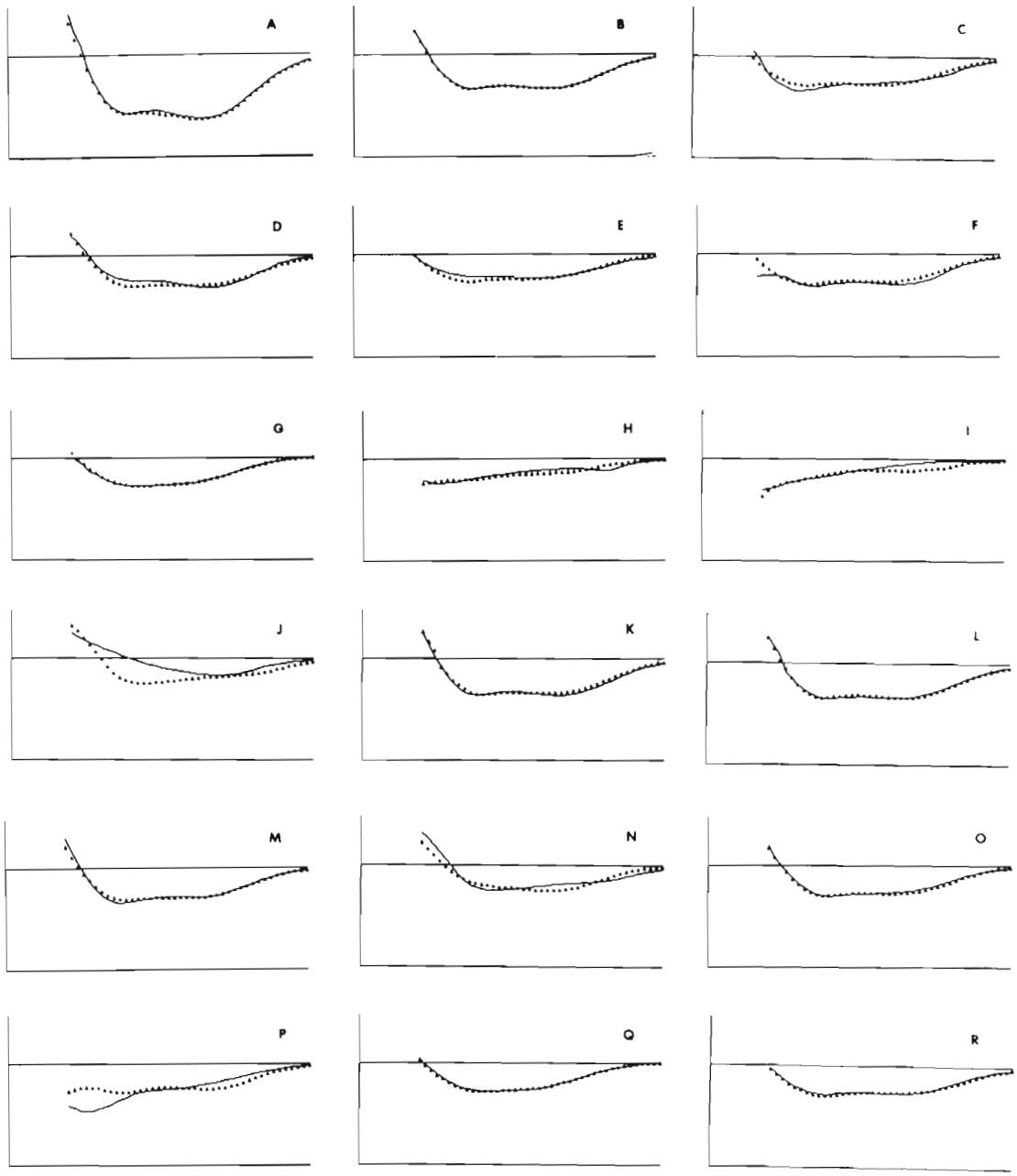

Fig. 4. CD spectra of the winning neuron obtained in the estimation of each of the 18 proteic examples (△) and actual spectrum (—). The winning neuron spectrum is the mean over 20 algorithm runs. The axis limits are the same as those of Figure 3. The corresponding proteins are (**A**) myoglobin, (**B**) lactate dehydrogenase, (**C**) lysozyme, (**D**) cytochrome *c*, (**E**) subtilisin BPN′, (**F**) papain, (**G**) ribonuclease A, (**H**) α-chymotrypsin, (**I**) elastase, (**J**) concanavalin A, (**K**) parvalbumin, (**L**) adenylate kinase, (**M**) insulin, (**N**) carboxypeptidase A, (**O**) thermolysin, (**P**) trypsin inhibitor, (**Q**) ribonuclease S and (**R**) nuclease.

teins of known structure. The map is continuous, i.e. proximal neurons respond to similar spectra. The main features present in the set of CD spectra are extracted, being displayed in an invariable map that suggests the underlying presence of a secondary structure map.

Secondly, a structure map is obtained from the proteinotopic map whose continuity is preserved. Therefore, similar structure values correspond to similar CD spectra. The structure values are interpolated among those of the examples and therefore negative secondary structure values cannot appear.

Given a problem protein, its winning neuron is defined as the one whose weight vector is the closest to the spectrum of this protein. The structure values corresponding to that winning neuron by the estimation method are assigned to the problem protein. It was shown that when the spectrum vector of the winning neuron is quite close to the real spectrum, the error in the estimation is low. This allows us to define a threshold for the distance between these spectra that assures a maximal error.

The method works in the 200−240 nm range, which is the analysed part of the CD spectra when a physiological medium is used. In addition, this method allows us to calculate a spectrum, giving a direct visualization of how the algorithm works. Due to the properties of the SOM algorithm it is not necessary to filter or to make corrections on the set of examples. During the training of the network, clusters of neurons compete for the set of proteins. Those clusters responding to examples with anomalous

Table I. Structure values estimated for eighteen proteins and the corresponding real values

	Calculated			Real		
	α	β	γ	α	β	γ
Myoglobin	0.74 ± 0.19	0.08 ± 0.02	0.17 ± 0.09	0.79	0.00	0.21
Lactate dehydrogenase	0.55 ± 0.04	0.11 ± 0.04	0.34 ± 0.04	0.45	0.24	0.31
Lysozyme	0.24 ± 0.00	0.15 ± 0.00	0.61 ± 0.00	0.41	0.16	0.43
Cytochrome *c*	0.43 ± 0.02	0.23 ± 0.00	0.34 ± 0.03	0.39	0.00	0.61
Subtilisin BPN'	0.25 ± 0.03	0.28 ± 0.09	0.47 ± 0.09	0.31	0.10	0.59
Papain	0.26 ± 0.03	0.15 ± 0.00	0.58 ± 0.03	0.28	0.14	0.58
Ribonuclease A	0.26 ± 0.00	0.43 ± 0.00	0.31 ± 0.01	0.23	0.40	0.37
α-Chymotrypsin	0.12 ± 0.07	0.33 ± 0.06	0.55 ± 0.09	0.09	0.34	0.57
Elastase	0.05 ± 0.02	0.20 ± 0.07	0.75 ± 0.09	0.07	0.52	0.41
Concanavalin A	0.37 ± 0.00	0.15 ± 0.00	0.48 ± 0.00	0.02	0.51	0.47
Parvalbumin	0.53 ± 0.01	0.13 ± 0.01	0.34 ± 0.01	0.62	0.05	0.33
Adenylate kinase	0.55 ± 0.03	0.14 ± 0.04	0.31 ± 0.02	0.54	0.12	0.34
Insulin	0.41 ± 0.04	0.20 ± 0.05	0.40 ± 0.05	0.51	0.24	0.25
Carboxypeptidase A	0.32 ± 0.10	0.15 ± 0.11	0.53 ± 0.05	0.37	0.15	0.48
Thermolysin	0.39 ± 0.09	0.23 ± 0.11	0.38 ± 0.10	0.36	0.22	0.42
Trypsin inhibitor	0.21 ± 0.10	0.21 ± 0.11	0.58 ± 0.06	0.28	0.33	0.39
Ribonuclease S	0.23 ± 0.00	0.39 ± 0.01	0.38 ± 0.00	0.26	0.44	0.30
Nuclease	0.34 ± 0.03	0.16 ± 0.06	0.50 ± 0.05	0.24	0.15	0.61

The estimated values were obtained using the distance method with $l = 2$. The presented values are the mean over 20 algorithm runs. The standard deviation values are shown to indicate the invariability of the results (a 0.00 value means an SD or less than one hundredth).

spectra (i.e. the spectra of proteins affected by other interactions different from those of the secondary structure) cannot cooperate with any other cluster and thus they are excluded from the map. The SOM algorithm imposes the self-organization of the map and these anomalous spectra are not considered (if the number of these bad examples is not excessive).

It is difficult to make a comparison with other methods for secondary structure fractions prediction since they use different wavelength ranges, calculate different β-structures and use a different number of example proteins. For classical methods, the Pearson coefficient is usually used to give a general idea of their fidelity. On the contrary, the algorithm presented in this work gives a particular maximal error measurement for each application.

However, in Table II the performance of several methods for secondary structure prediction and the SOM method are shown in terms of Pearson correlation coefficients. In each prediction of the SOM method, a distance from the problem protein to the winning neuron weight vector can be defined. To signify the correlation between this distance and the error in the prediction, the Pearson coefficients of the prediction of problem proteins which have distances of less than a given distance threshold, are shown.

The reliability of the poly(L-lysine) as a structural molecular model for reference α-, β- and (especially) random coil conformation has been discussed (see Yang *et al.*, 1986; Drake *et al.*, 1988). In any case, we have extracted from the work of Drake *et al.* (1988) the approximate CD values of the poly(L-lysine) dissolved in water at pH 7.6 and 85°C which is ascribed there to a disorganized conformation (since the spectra of the same polypeptide in the presence of urea 4 M is rather similar). The substitution of this spectrum in the set of example proteins, for that of the poly(L-lysine) (at pH 5.7, 22°C) taken from Yang *et al.* (1986), which has been used as a random coil model, does not provoke a significant change in the performance of the method (previous Pearson values: for $l = 2$, 0.91, 0.73, 0.64; new Pearson values: for $l = 2$, 0.91, 0.69, 0.73). The major difference between the two spectra is that the poly(L-lysine) at

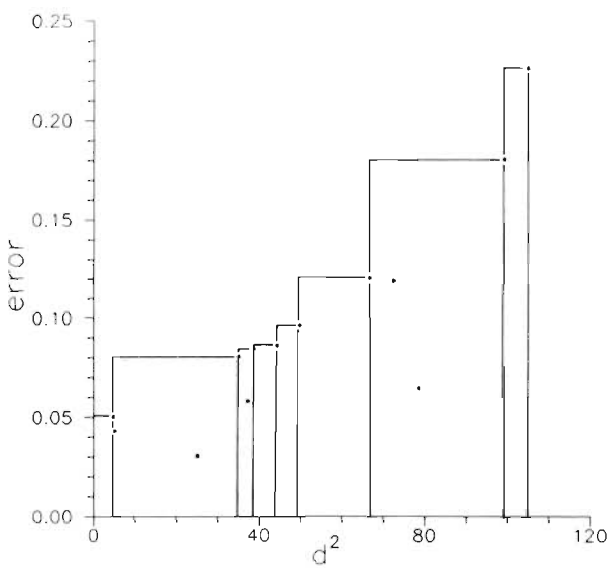

Fig. 5. Mean error in the estimation of the three secondary structure values (sum of the absolute errors in the α-, β- and random values estimation divided by three) plotted against the square distance from the winning neuron spectrum to the corresponding real CD spectrum. Each point corresponds to the mean over 20 estimations of the same protein. Five points are excluded from the representation since although they give a large d^2 value they have mean error values less than 0.25. Distance threshold values are represented as bars. Each bar is calculated by taking the upper right corner as a point having a higher error value than those of the points having lesser d^2 values.

85°C lacks positive CD values in the 200–240 nm range. Nevertheless, another model for random coil conformation [the reference spectra based on 15 proteins taken from Chang *et al.* (1978)], that has a similar shape to the spectra taken from Drake *et al.* (1988), is already present in the set of protein examples.

This algorithm works in seconds in a normal PC. Once a structure map is achieved, the evaluation of the structure of an unknown protein is easy and fast since it is only necessary to

Table II. Comparison of different methods of prediction of secondary structure fractions

Methods	a	b	α	β	t	r
Chang *et al.* (1978)	205–240	18	0.96	0.94	0.31	0.49
Brahms and Brahms (1980)	190–240	14	0.92	0.93	0.33	0.65
Provencher and Glockner (1981)	190–240	18	0.96	0.94	0.31	0.49
Hennesey and Johnson (1981)	178–260	16	0.98	−0.27	0.18	0.24
Manavalan and Johnson (1987)	190–260	16	0.95	0.45	0.54	0.69
Böhm *et al.* (1992)	200–250	13	1.00	−0.36; 0.84[a]	0.59	0.99
SOM method	200–240	24	0.91	0.73	–	0.64
SOM method[b]	200–240	24	0.93	0.97	–	0.82

When some alternative methods are reported, the methods either not using the CD spectra of the example spectra to calculate the correlation coefficient, or using the wavelength range more similar to the range in this work were selected. a, wavelength range used; b, number of example spectra; α, β, t and r, Pearson correlation coefficients for the prediction of the α, β, β-turns and random coil conformation percentages, respectively.
[a]For this method, antiparallel and parallel β-sheet percentages are shown.
[b]SOM method for problem proteins with square distance <44 (six spectra from or example set fit this condition).

find its position on the map and this does not need any new learning.

The addition of new examples for the learning process is straightforward to do. With more examples the learning time increases (the learning process take 20 s per example in an IBM PC with a 80386DX processor and a 25 MHz clock and 2 s per example in a SUN SPARCstation 2), but the resulting maps should be more accurate since the network has more spectra to interpolate. This is another difference of an unsupervised learning method from other estimation methods. Statistical methods are based on an unchangeable strategy. New examples will not improve their accuracy. The SOM method flexibility is based on the fact that it uses the set of known examples to interpolate between them: the more examples it uses the better the learning it will do.

Notes

Programs for PC computer or SUN SPARCstation 2 will be available in summer 1993 via anonymous ftp to solea.quim.ucm.es (internet number 147.96.5.69). To get the PC version enter 'get k2d.PC.tar.z'. To get the SUN version enter 'get k2d.SUN.tar.z'. To get an ASCII documentation file enter 'get k2d.read.me'.

Acknowledgements

This work was partially supported by grants PB89-0108 (DGICYT, Spain) and 90.324 (PRONTIC, Spain). One of the authors (M.A.A.) is a recipient of a fellowship from the Plan de Formacíon de Personal Investigador of the Ministerio de Educación y Ciencia, Spain.

References

Böhm,G., Rudolf,M. and Jaenicke,R. (1992) *Protein Engng*, **5**, 191–195.
Bohr,H., Bohr,J., Brunak,S., Cotterill,R.M.J., Lautrup,B., Norskov,L., Olsen,O.H. and Petersen,S.B. (1988) *FEBS Lett.*, **241**, 223–228.
Bohr,H., Bohr,J., Cotterill,R.M.J., Fredholm,H., Lautrup,B. and Petersen,S.B. (1990) *FEBS Lett.*, **261**, 43–46.
Chang,C.T., Wu,C.S.C. and Yang,J.T. (1978) *Anal. Biochem.*, **91**, 13–31.
Drake,A.F., Siligardi,G. and Gibbons,W.A. (1988) *Biophys. Chem.*, **31**, 143–146.
Ferrán,E.A. and Ferrara,P. (1991) *Biol. Cybern.*, **65**, 451–458.
Fessenden,R.J. (1991) *J. Chem. Soc. Perkins Trans. 2*, **11**, 1755–1762.
Hennessey,J.P. and Johnson,W.C. (1981) *Biochemistry*, **20**, 1085–1094.
Hirst,J.D. and Sternberg,M.J.E. (1992) *Biochemistry*, **31**, 7211–7218.
Holbrook,R., Muskal,S.M. and Kim,S.H. (1990) *Protein Engng.*, **8**, 659–665.
Holley,H. and Karplus,M. (1989) *Proc. Natl Acad. Sci. USA*, **86**, 152–156.
Kneller,D.G., Cohen,F.E. and Langridge,R. (1990) *J. Mol. Biol.*, **214**, 171–182.
Kohonen,T. (1982) *Biol. Cybern.*, **43**, 59–69.
Kohonen,T., Mäkisara,K. and Saramäki,T. (1984) *IEEE 7th Conf. Pattern Recognition*, Montreal, Canada, pp. 182–185.
Kohonen,T. (1986) *Neur. Networks*, **8**, 13–16.
Kohonen,T. (1990) *Proc. IEEF*, **78**, 1464–1480.
Manavalan,P. and Johnson,W.C. (1983) *Nature*, **305**, 831–832.
Manavalan,P. and Johnson,W.C. (1985) *J. Biosci.*, **8** (Suppl.), 141–149.
Manavalan,P. and Johnson,W.C. (1987) *Anal. Biochem.*, **167**, 76–85.
Menéndez-Arias,L., Gómez-Gutiérrez,J., García-Fernández,M., García-Tejedor,A. and Morán,F. (1988) *CABIOS*, **4**, 479–482.
Merelo,J.J., Andrade,M.A., Ureña,C., Prieto,A. and Morán,F. (1991a) In Prieto,A. (ed.), *Artificial Neural Networks*. Lecture Notes in Computer Science 540, Springer Verlag, Berlin, pp. 415–421.
Merelo,J.J., Andrade,M.A., Prieto,A. and Morán,F. (1991b) *Fourth Int. Conf. Neural Networks and Their Applications*, Nimes, France.
Muskal,S.M., Holbrook,S.R. and Kim,S.H. (1990) *Protein Engng*, **3**, 667–672.
Perczel,A., Hollósi,M., Tusnády,G. and Fasman,G.D. (1991) *Protein Engng*, **4**, 669–679.
Petersen,B., Bohr,H., Bohr,J., Brunak,S., Cotterill,R.M.J., Fredholm,H. and Lautrup,B. (1990) *Trends Biotechnol.*, **8**, 304–308.
Provencher,S.W. and Glöckner,J. (1981) *Biochemistry*, **20**, 33–37.
Qian,N. and Sejnowski,T.J. (1988) *J. Mol. Biol.*, **202**, 865–884.
Rumelhart,D.E. and McClelland,J.L. (1988) *Parallel Distributed Processing*. MIT Press, Cambridge.
Toumadje,A., Alcorn,S.W. and Johnson,W.C. (1992) *Anal. Biochem.*, **200**, 321–331.
van Stokkum,I.H.M., Spoelder,H.J.W., Bloemendal,M., van Grondelle,R. and Groen,F.C.A. (1990) *Anal. Biochem.*, **191**, 110–118.
Yang,J.T., Wu,C.C. and Martínez,H.M. (1986) *Methods Enzymol.*, **130**, 208–271.

Received on July 23, 1992; revised on January 15, 1993; accepted on January 28, 1993