Prediction of the location and type of β -turns in proteins using neural networks

ADRIAN J. SHEPHERD,^{1,2} DENISE GORSE,² AND JANET M. THORNTON^{1,3}

¹Department of Biochemistry and Molecular Biology, University College London, Gower Street, London WC1E 6BT, United Kingdom

²Department of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom

³Department of Crystallography, Birkbeck College, Malet Street, London WC1E 7HX, United Kingdom

(RECEIVED July 29, 1998; ACCEPTED December 28, 1998)

Abstract

A neural network has been used to predict both the location and the type of β -turns in a set of 300 nonhomologous protein domains. A substantial improvement in prediction accuracy compared with previous methods has been achieved by incorporating secondary structure information in the input data. The total percentage of residues correctly classified as β -turn or not- β -turn is around 75% with predicted secondary structure information. More significantly, the method gives a Matthews correlation coefficient (MCC) of around 0.35, compared with a typical MCC of around 0.20 using other β -turn prediction methods. Our method also distinguishes the two most numerous and well-defined types of β -turn, types I and II, with a significant level of accuracy (MCCs 0.22 and 0.26, respectively).

Keywords: β -turn prediction; feed-forward networks; neural networks; secondary structure prediction

A β -turn is a four-residue reversal in a protein chain that is not within an α -helix (although a β -turn may overlap the end of an α -helix by up to three residues); the distance between $C_{\alpha}(i)$ and $C_{\alpha}(i + 3)$ is less than 7 Å. About a quarter of all protein residues are in β -turns. With respect to the three-dimensional structure of a protein, β -turns tend to be found at the solvent-exposed surface, hence the prevalence of hydrophilic residues. Contrary to the “classic” image of a β -turn as an isolated reversal in a chain, Hutchinson and Thornton (1994) report that 58% of β -turns are multiple turns, i.e., they share one or more residues with another β -turn.

β -turns are typically classified into different types according to the distribution of their ϕ , ψ angles. This paper uses the classification scheme proposed by Hutchinson and Thornton (1994), which recognizes nine distinct types of β -turn: types I, I', II, II', VIa1, VIa2, VIb, VIII, and IV (see Table 1). An important feature of this classification is that the most frequently-occurring type, type IV β -turns (around 35%), is the least well defined; it is, in effect, a repository for all those β -turns that do not fit into one of the other well-defined classes. The rarity of β -turn types VIa1, VIa2, and VIb together with the heterogeneity of type IV β -turns pose severe obstacles to the development of β -turn prediction methods that are both accurate and comprehensive.

β -turns play an important role in the higher-order structure of proteins. They are, for example, an integral component of β -hairpins

(the fundamental building block for antiparallel β -sheets). Moreover, it is widely held that β -turn formation can be an important stage in the folding of a protein (Rose, 1978). The far more radical view that β -turns have an active role in directing folding pathways (Zimmerman & Scheraga, 1977; Yang et al., 1996) is, however, highly contentious.

To sum up, the development of an accurate method for identifying the location and type of β -turns within a protein sequence would not only aid the identification of simple structural motifs, such as β -hairpins; it would be an important step toward the ultimate goal of predicting the detailed three-dimensional structure of a protein from its amino-acid sequence alone.

β -Turn prediction methods

Probably the best-known empirical approach to β -turn prediction is the *positional preference* approach (Lewis et al., 1973; Chou & Fasman, 1974; Wilmot & Thornton, 1988; Hutchinson & Thornton, 1994). For each β -turn type, a set of propensities is derived quantifying the likelihood of a particular amino-acid type occurring at each position (from 1 to 4) within the turn. This approach can be extended by taking into account correlations between the pairing of residues at positions 1–4 and 2–3, the *correlation of coupling-effects* approach (Chou & Blinn, 1997; Zhang & Chou, 1997). An alternative approach is to utilize information about residue hydrophobicity in recognition of the hydrophilic tendencies of β -turns (Rose, 1978).

Another alternative is to use a set of *neural networks* to learn appropriate sets of parameters from the available data (McGregor

Reprint requests to: Dr. Adrian J. Shepherd, Department of Biochemistry and Molecular Biology, University College London, Gower Street, London WC1E 6BT, United Kingdom; e-mail: a.shepherd@biochem.ucl.ac.uk.

Table 1. Number and relative frequency of β -turn types in the data set

β -turn type	Number	%
I	1,153	34.3
I'	117	3.5
II	433	12.9
II'	74	2.2
VIa1	19	0.6
VIa2	11	0.3
VIb	25	0.7
VIII	317	9.4
IV	1,210	36.0
Total	3,359	100.0

et al., 1989). Neural networks (more accurately called “artificial neural networks”) are parallel, distributed information processing structures that draw their ultimate inspiration from neurons in the brain. In the present context, however, a neural network is best viewed as a type of pattern classification tool capable (at least in theory) of identifying relevant correlations in the data without the user having to program them into the system in advance. The neural networks used in this research are described in Materials and methods.

Although a range of techniques has been applied to β -turn prediction, it has been somewhat neglected compared to the secondary-structure prediction problem. This is particularly the case with neural network methods; we know of only a single published paper about β -turn prediction with neural nets (McGregor et al., 1989) compared with at least a dozen for secondary structure prediction.

The β -turn prediction problem can be divided into two subproblems: the ability to identify which residues in a protein chain are in β -turns (the β -turn location problem), and the ability to identify individual β -turn types (the β -turn classification problem). Published results for the β -turn location problem are superficially impressive compared to secondary-structure prediction, with over 70% of residues classified correctly vs. a peak of around 75% for secondary structure prediction. However, this measure of predictive success (known as Q_{total}) is highly misleading owing to the disparity between the number of β -turn residues (around 25%) and non- β -turn residues; hence, it is possible to get a Q_{total} score of about 75% by the trivial strategy of predicting all residues to be not-turn residues. When a more robust measure of predictive performance is used, such as the *Matthew's correlation coefficient* (MCC), β -turn prediction methods appear far less successful than secondary structure prediction methods. For example, McGregor et al. (1989) achieved an MCC of only 0.20 with their β -turn prediction networks, whereas Qian and Sejnowski (1988) reported MCCs of 0.41 (α -helix residues), 0.31 (β -strand residues), and 0.41 (coil residues) for their secondary structure prediction networks. (Note that, throughout this paper, the term “coil” is used to describe all those residues not forming part of an α -helix or β -strand.) Statistical measures of prediction accuracy are discussed in Materials and methods.

A new approach to β -turn prediction with neural networks

Our β -turn prediction method differs from previously published methods in several important respects:

- We incorporate secondary structure information about each residue, not just information about amino-acid types.
- Our method utilizes more local information about the protein chain, whereas other methods make decisions based on information about only four adjacent residues at a time (i.e., the four potential β -turn residues). Our predictions take into account information from up to 11 adjacent residues.
- Our method treats the β -turn location problem and the β -turn classification problem as logically separate (see Materials and methods for details). All other methods attempt to predict β -turn location and type in a single operation.

In addition, our method utilizes neural networks in a substantially different way to McGregor et al. (1989). For example, we use a second filtering (structure–structure) network to postprocess the outputs from the main turn/not-turn prediction network. Details of the differences between our respective methodologies are given in Materials and methods.

Materials and methods

Formulating the β -turn prediction problem

Our methodology for predicting β -turns can be divided into four distinct stages:

1. *Turn/not-turn prediction.* A neural network is used to predict whether or not a given residue is part of a β -turn.
2. *Turn/not-turn filtering.* We use a mixture of filtering networks and rules to improve the accuracy (or adjust the rate of over-prediction) of the Stage 1 predictions.
3. *Turn-type prediction.* The individual turn-type of the “potential-turns” (as predicted at Stages 1 and 2) is predicted using a separate neural network for each turn type to be predicted.
4. *Resolution of conflicting turn-type predictions.* Any conflicting predictions arising at Stage 3 (when the outputs from two or more networks is greater than 0.5 for the same protein tetramer) are resolved on a simple “winner-takes-all” basis, i.e., the turn type is deemed to be the same as that of the network with the largest output.

Rather than attempt to predict all the β -turn types defined in Hutchinson and Thornton (1994) at Stages 3 and 4, we focus on the four most numerous types, well-defined β -turn types I, II and VIII, and (as a control) heterogeneous type IV β -turns. Note that whereas the predictions at Stages 1 and 2 are residue-by-residue, the predictions at Stages 3 and 4 are turn-by-turn.

The protein data

The data set used in this research comprised 300 nonhomologous protein domains (less than 25% pairwise sequence identity) with resolution 2 Å or better. The domains were chosen to be structurally nonhomologous as defined in the CATH classification of known protein structures (Orengo et al., 1997). The set contains 45,532 residues and 3,359 β -turns (excluding turns that span a domain boundary); 24% of residues in the data set are found in β -turns. A breakdown of these β -turns into the nine β -turn types defined in Hutchinson and Thornton (1994) is given in Table 1.

Table 2. Comparison of different β -turn prediction data sets

Paper	Protein chains	β -Turns	n -Fold cross-validation	Nonhomologous
Wilmot and Thornton (1988)	59	714	No	No ^a
McGregor et al. (1989)	58 ^b	714	6 ^c	No ^d
Hutchinson and Thornton (1994)	205	3,899	N/A	Yes
Zhang and Chou (1997)	96	678	No ^e	Yes? ^f
Chou and Blinn (1997)	205	2,215 ^g	Jackknife	Yes
Present work	300 ^h	3,359	10	Yes

^aAlthough the data set contained homologous proteins, no identical β -turn tetrapeptides occurred in the set (after the removal of 11 such turns).

^bThis is the same set of proteins as (Wilmot & Thornton, 1988) with one chain removed.

^cIn carrying out their n -fold cross-validation, McGregor et al. (1989) retained the global proportion of each target class (type I turn, type II turn, other turn, and nonturn) within each training and validation set. The reported accuracy of their method is therefore likely to be better than it would have been with a standard, randomly-proportioned validation set.

^dNote: McGregor et al. (1989) erroneously claim that the set is nonhomologous.

^eZhang and Chou (1997) used a single training set (30 proteins) and a single validation set (66 proteins). Non- β -turn tetrapeptides in the validation set that appear as β -turns in the training set were removed!

^fAlthough Zhang and Chou (1997) do not specify the degree of homology, they did ensure that none of the 233 β -turn tetrapeptides in the validation set were simultaneously present in the training set.

^gThis is the same data set as that in Hutchinson and Thornton (1994) but with type IV turns removed (as well as any β -turns containing uncertain or nonstandard amino acid residues).

^hDomains, not chains.

It is worth noting that the data set used here is much larger than that used to devise and test other β -turn prediction methods (see Table 2). There are good reasons for expecting an increase in the amount of protein data to produce a small but significant improvement in predictive performance; Chandonia and Karplus (1996) achieved a 4% improvement in their secondary structure prediction Q_{total} by increasing their data set by a factor of 3.

We evaluated our method using three different combinations of residue-level data:

- *Amino-acid types only.* This is the type of input data used by McGregor et al. (1989) and in the development of most methods based on the positional-propensity approach.
- *Amino-acid types and secondary structure predictions,* as provided by the PHDsec program (Rost & Sander, 1993, 1994). PHDsec's multiple-alignment option was enabled.
- *Amino-acid type and secondary structure assignments* (as specified by the SSTRUC program). We used this data combination to establish an upper bound for the performance of our method with predicted secondary structure information.

Although our data set was strictly nonhomologous, it did contain homologues of some of the protein chains used to train the PHDsec networks. As a consequence, the PHDsec predictions for our data set were slightly more accurate than one would anticipate with a data set containing no such homologues. However, the difference is small, 73.5% of residues correctly classified compared with 72.1%, and Matthews correlation coefficients of 0.66 (helix), 0.57 (strand), and 0.53 (coil) compared with 0.63, 0.53 and 0.52,⁴ so the impact on our results is almost certainly negligible (we estimate an upward distortion of <1%).

⁴These statistics for the anticipated performance of the PHDsec program when applied to nonhomologous protein data were taken from information provided by the PredictProtein server at EMBL.

The training and validation sets

Before presentation to a neural network, the chosen protein data must be encoded in a set of P input patterns together with their associated target outputs. In this case the input pattern represents a window, W residues in size, onto part of a protein domain. Our experience is that increasing the amount of local information available to the network by increasing W improves prediction accuracy (see Results for details).

Information about individual amino acid types is encoded using the same *unary encoding* scheme as Riis and Krogh (1996); every amino-acid type is represented by a single one (which varies in position depending on the type) and 19 zeros, e.g., alanine $\rightarrow$ (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0). For windows that overlap either end of a domain, a special "null" vector of 20 zeros is used to fill in the blank position(s). Secondary structure assignment information is encoded as follows: helix $\rightarrow$ (1,0,0), strand $\rightarrow$ (0,1,0), and coil $\rightarrow$ (0,0,1). Secondary structure prediction information is encoded using the *reliability indices* provided by the PHDsec program. The reliability indices are just the strengths of the prediction for each of the three target states (helix, strand, coil) represented by an integer in the range 0–9; we simply divide each index by 10 to get three real numbers between 0 and 1 for each secondary structure prediction.

In contrast to the size of input pattern, which varies in size from training set to training set, the target outputs always consist of a single binary number, i.e., the target output is 1 or 0 (true or false) for all patterns in all of the training sets used in this research.

An important feature of both the turn/not-turn and the turn-type prediction tasks is that each target class accounts for a comparatively small proportion of the available data (for example, around 1 in 4 residues are in β -turns, and around 1 in 3 β -turns are type I). If the data in the training set retain the observed proportions, the neural network will tend to overpredict the most numerous class (in this case "not-turn" or "not-turn-type-X"). If, on the other hand,

an equal number of examples from each class are selected for the training set, the net will tend to overpredict the least numerous class when presented with data that retains the naturally-occurring proportions. We call the former an *unbalanced training set* and the latter a *fully-balanced training set*. The optimal proportions (which typically lie somewhere between “fully-balanced” and “unbalanced”) are task-specific, and also depend on the chosen measure of predictive accuracy (see Materials and methods). McGregor et al. (1989) obtained their best results using a training set with patterns in the ratio 6:3:8:34 (for type I turns, type II turns, other turns, and nonturns, respectively).

One implication of the way in which McGregor et al. (1989) “balance” their training set is that data containing potentially-valuable turn information is withheld from the network. We use a different balancing strategy that modifies the network weights but leaves the training set unchanged, thereby preventing loss of information. For a training set with P patterns in C target classes, the gradient contribution (weight change) $\mathbf{g}^p$ for a pattern p with target class c ($1 \leq p \leq P$, $1 \leq c \leq C$) is multiplied by the factor β^c given by

$$\beta^c = \frac{\alpha}{\alpha - \lambda(\alpha - N^c)}, \quad (1)$$

where α is given by

$$\alpha = \frac{P}{C}, \quad (2)$$

N^c is the number of training patterns in class c and λ is a user-defined *balancing factor* in the range $[0, 1]$. With $\lambda = 1$ the weight-change contribution for each class is fully-balanced; with $\lambda = 0$ the weight-change contributions remain in their original, “fully-unbalanced” state (i.e., $\beta^c = 1$).

Generally, we have used settings of $\lambda = 0$ and $\lambda = 1$. Rather than engage in a lengthy process of optimizing parameter λ , we prefer to interpolate between over- and underprediction using the computationally inexpensive “state-shifting” filtering rules described below.

To gain an unbiased measure of predictive performance, it is necessary to test a neural network using data that was not used to train it. To this end, we carried out *10-fold cross-validation* with all our networks. This involved training a given network 10 times, on each occasion withholding a different set of 30 domains from the training set; this set of 30 domains, known as the *validation set*, is used exclusively to test the network. No “balancing” is applied to the validation set. By combining the results from the 10 validation sets, it is possible to produce unbiased results for the entire data set of 300 domains.

Note, however, that it is not legitimate to optimize an algorithm on the basis of its n -fold validation performance. All optimizations made to our algorithm were done so on the basis of the training set performance, or independent criteria (e.g., whether a given prediction is “ β -turn-like”). A comparison of the data sets used by various researchers in this field is given in Table 2.

The neural networks

All of the networks used for this research are *feed-forward networks* (Rumelhart et al., 1986), which have a characteristic layered

architecture. The simplest form is the *single layer perceptron* (SLP) with just an input layer (not counted because it has no computational function) and an output layer. SLPs are limited in the range of functions (or mappings) they can represent (for example, they are incapable of solving linearly-inseparable problems). *Multi-layer perceptrons* (MLPs), with one or more *hidden layers* in-between the input and output layers, are capable of finding solutions to a much wider range of problems. We have used a mixture of SLPs and two-layer MLPs for this research.

The notation used to denote the different network architectures is illustrated by the following example. The 9(23)–10–1 network used extensively in this research has an input window 9 residues wide with 23 network inputs per window position (20 for the amino-acid information plus 3 for the secondary-structure information). This gives a total of 208 inputs to the network (including an extra input for the *bias unit*, a standard feature of feed-forward nets). The network has 10 units in a single hidden layer and one output unit.

Training a feed-forward network is an iterative process that involves repeatedly presenting the training set (containing exemplar patterns with known target outputs) to the network. After each presentation the network parameters (or *weights*) are adjusted so that the network’s total error for all patterns in the set (as measured by an appropriate error function) is progressively reduced. This type of training is known as *supervised learning* and the algorithm for adjusting the network weights is the *training method*. The “traditional” method for training feed-forward networks is *back-propagation* (Rumelhart et al., 1986), which derives from the oldest and simplest of classical optimization techniques, *steepest descent*. For this research we chose a more modern training method, the *scaled conjugate gradient algorithm* (Møller, 1993), as it frequently requires far fewer training iterations than backpropagation. In practice, we required at most 100 training iterations compared with McGregor et al.’s 10,000 iterations of backpropagation. However, conjugate gradients (in line with other second-order training methods) is known to be more likely to get trapped in local minima than backpropagation with some training tasks (Gorse et al., 1997; Shepherd, 1997). We occasionally encountered problems consistent with the network getting trapped in a local minimum when applying a two-layer network to the turn/not-turn problem.

Filtering

An important feature of the predictions generated by the turn/not-turn network is that they are *uncorrelated* (i.e., the network makes each prediction for a residue in isolation without reference to neighboring predictions). One way to ensure correlated predictions is to feed the outputs from the turn/not-turn net into a second network, called a *filter network* (or structure-structure network). An additional benefit is that the number of inputs per residue is much smaller (one, or four if secondary structure information is included) than for the turn/not-turn network, meaning much wider input windows for the same number of weights. Qian and Sejnowski (1988) achieved a 2% improvement in secondary structure prediction accuracy using a filter network.

Another approach to filtering (postprocessing) is to devise arbitrarily-complex rules for improving the network predictions. We have experimented with two types of filtering rule. The first type, dubbed “*state-shifting*” rules, is designed to exploit the fact that the strength of a given prediction from the turn/not-turn network is a good indication of its reliability (see Fig. 2), i.e., the

stronger the prediction, the more likely it is to be correct. This means we can justifiably calculate *reliability indices* (RIs) for the network outputs; we use the standard formula

$$\text{integer}[\text{abs}(\text{net}_{\text{output}} - 0.5) \times 20], \quad (3)$$

giving an integer in the range 0–9 (all network outputs are constrained to the range 0–1). We then apply a state-shifting rule to change the states of residues whose prediction strength lies below a chosen threshold. For example, one rule (represented as $T < 1$) flips all turn predictions with $RI = 0$ to not-turn, whereas a different rule ($NT < 4$) flips all not-turn predictions with $RI < 4$ to turn.

“State-shifting” rules make it possible to interpolate between the over- and underprediction of β -turn at will with minimal computational costs (in contrast to balancing the training set). This enables us to:

- Equalize the rates of over- and underprediction for our final turn/not-turn results.
- Increase the rate of overprediction when using our turn/not-turn results to preprocess the data set for our individual turn/not-turn networks. (In other words, by overpredicting β -turn we can allow more “might-be-turn” residues into the turn-type data set.)

The second type of filtering rule, dubbed “*state-flipping*” rules, is used to make the predictions more “ β -turn-like” by ensuring that residues predicted to be turn occur in groups of at least four adjacent residues. After a short analysis of various combination of “state-flipping” rule, we settled on a set of four rules executed in the following order:

1. Flip isolated nonturn predictions to turn (i.e., t-t $\rightarrow$ ttt).
2. Flip isolated turn predictions to nonturn (i.e., -t $\rightarrow$ ---).
3. Flip isolated pairs of turn predictions to nonturn (i.e., -tt $\rightarrow$ ----).
4. Given an isolated triplet of turn predictions, flip the adjacent nonturn prediction with the lowest RI to turn (i.e., -ttt $\rightarrow$ tttt or -ttt).

Measuring performance

There are a number of standard statistical measures used to measure the performance of predictive classification methods in biochemistry. In the context of the turn/not-turn prediction problem, the four most frequently-used measures can be derived from two or more of the following scalar quantities:

p = number of correctly-classified β -turn residues

n = number of correctly-classified non- β -turn residues

o = number non- β -turn residues incorrectly classified as β -turn residues (overpredictions)

u = number β -turn residues incorrectly classified as non- β -turn residues (underpredictions)

$t = p + n + o + u$ (total number of residues).

The straightforward measure of a method’s overall performance is Q_{total} , the percentage of correctly classified residues, given by

$$Q_{\text{total}} = \frac{p + n}{t} \times 100. \quad (4)$$

As noted previously, Q_{total} is liable to give a highly misleading impression of predictive performance when one or more of the target classes contain a relatively small proportion of the data (as is the case with β -turn prediction). Two rather more useful measures (particularly when taken together) are $Q_{\text{predicted}}$, the percentage of β -turn predictions that are correct, given by

$$Q_{\text{predicted}} = \frac{p}{p + o} \times 100 \quad (5)$$

and Q_{observed} , the percentage of observed β -turns that are correctly predicted, given by

$$Q_{\text{observed}} = \frac{p}{p + u} \times 100. \quad (6)$$

$Q_{\text{predicted}}$ penalizes overpredictions, whereas Q_{observed} penalizes underpredictions. We typically seek to equalize the values of $Q_{\text{predicted}}$ and Q_{observed} when using “state-shifting” rules to filter our predictions (see above). For a single measure of performance that takes into account both over- and underpredictions, we use the *Matthews correlation coefficient* (MCC), given by

$$\text{MCC} = \frac{pn - ou}{\sqrt{(p + o)(p + u)(n + o)(n + u)}}. \quad (7)$$

Another useful approach is to compare the accuracy of a method’s predictions with the anticipated accuracy of predictions generated by chance. Here we calculate the total number of residues that are expected to be classified correctly by randomly-generated predictions (R_{total}), ensuring that the proportion of turn to nonturn is the same as for the method to which it is being compared (so the latter does not gain by overpredicting). The requisite formula is

$$R_{\text{total}} = \frac{(p + o)(p + u) + (n + u)(n + o)}{t} \quad (8)$$

To measure how well a method is performing compared with R_{total} in a way that is independent of the scale of t and R_{total} , we calculate the *normalized percentage better than random* (S), given by

$$S = \frac{(p + n) - R_{\text{total}}}{t - R_{\text{total}}} \times 100. \quad (9)$$

Perfect predictions score $S = 100\%$, predictions that are no better than random score $S = 0\%$.

All of the preceding metrics give credit exclusively to turn predictions that are completely accurate; predictions that are displaced by as little as a single residue gain no credit at all. Since predictions that “miss” by only one or two residues contain potentially useful information, it is useful to employ additional metrics that take this into account. Here we have recalculated Q_{observed} and $Q_{\text{predicted}}$ to allow for predictions displaced by up to two residues.

In the case of Q_{observed} , a given observed turn is deemed to have been correctly predicted if there is at least one predicted turn within n ($1 \leq n \leq 2$) residues of that turn; in the case of $Q_{\text{predicted}}$, a given predicted turn is deemed to be correct if there is at least one observed turn within n residues of that prediction.

Results

The results given in this section were calculated by combining the separate results for each of the 10 validation sets, as described in Materials and methods. Training was for 50 epochs (turn/not-turn networks) or 10 epochs (turn/not-turn filtering networks and turn-type networks). In every case the network converged to a stable error-level before training was terminated.

Predicting the location of turn residues

Turn/not-turn predictions

To establish a “baseline” for our formulation of the turn/not-turn prediction problem, we trained a single-layer network with a window 5 residues wide using an (unbalanced) training set that contained no secondary structure information. This minimal 5(20)-1 network achieved an MCC of 0.200. This compares with the MCC = 0.177 achieved by McGregor et al. (1989) using their 4(20)-1 turn-by-turn prediction networks (with far less training data), suggesting that reformulating the turn/not-turn prediction problem for residue-by-residue prediction makes it, at most, slightly easier to solve.

Starting from this baseline, we found that the single biggest improvement in performance was achieved by adding secondary structure information. Hence, with a 5(23)-1 network we raised the MCC from 0.2 to 0.297 using predicted secondary structure information and to 0.484 using observed secondary structure information. Small additional improvements were achieved by increasing the window size up to limit of nine (see Fig. 1) and by adding a hidden layer of 10 units. The combined result was a 9(23)-10-1 network that achieved MCCs of 0.315 (predicted) and 0.512 (observed).

As expected, changing the balancing factor from $\lambda = 0$ to $\lambda = 1$ typically produced a dramatic increase in the overpredictions rate and a corresponding reduction in the underprediction rate. What is less intuitive is that the Matthews correlation coefficient consistently favored networks that overpredicted the occurrence of β -turn, i.e., the MCC is higher with $\lambda = 1$ than $\lambda = 0$ (with the exception of networks with hidden units, see below). This is illustrated by the

Fig. 1. The correlation between window size and prediction accuracy. Results are for networks with zero hidden units and balancing factor $\lambda = 0$.

results for a 5(23)-1 network with predicted secondary structure information given in Table 3.

This effect ultimately stems from the fact that, with the benefit of secondary structure information, the network predicts most turn residues to be in coil regions, where around half of the residues are in β -turns. Because the vast majority of turn residues occur in coil regions, it is possible to score a highly respectable MCC = 0.48, merely by predicting all observed coil residues to be turn and all noncoil residues to be not-turn. This trivial strategy fails completely, however, if we calculate scores for observed coil residues alone (see Table 3, “observed coil = turn” and “predicted coil = turn” columns).

These observations demonstrate that even the most respectable statistical measures can be misleading. To offset this problem, we have included coil-residue-only scores in our tabulated results.

Although setting $\lambda = 1$ generally gave better overall results than $\lambda = 0$, our 9(23)-10-1 network appeared to become trapped in a local minimum with $\lambda = 1$ (using both predicted and observed secondary structure information). We, therefore, chose to defer any “balancing” until the filtering phase of our method (see below). Our best turn/not-turn predictions with a single network, the 9(23)-10-1 network with $\lambda = 0$, are given in Table 4.

Table 3. Balancing parameter λ^a

	$\lambda = 0$		$\lambda = 1$		Observed coil = turn	Predicted coil = turn
Q_{total}	78.0	(60.3)	68.6	(55.2)	73.8 (47.4)	65.7 (51.8)
Q_{observed}	29.5	(32.5)	70.5	(74.7)	85.7 (100.0)	74.4 (79.8)
$Q_{\text{predicted}}$	58.1	(66.5)	41.0	(51.9)	47.4 (47.4)	38.8 (49.5)
MCC	0.297	(0.210)	0.334	(0.132)	0.480 (0.0)	0.320 (0.076)

^aResults are for a 5(23)-1 network using predicted secondary structure information. Column 4 records the scores achieved if all observed coil residues are predicted as turn and all noncoil residues are predicted as not-turn. Column 5 records the scores achieved if all predicted coil residues are predicted as turn and all noncoil residues are predicted as not-turn. Results in brackets are for observed coil residues only.

Table 4. Best turn/not-turn predictions with a single network^a

	Observed	Predicted
Q_{total} (%)	82.6 (66.9)	78.3 (61.3)
Q_{observed} (%)	60.2 (66.5)	31.8 (34.5)
$Q_{\text{predicted}}$ (%)	64.8 (64.7)	58.8 (68.2)
MCC	0.512 (0.337)	0.315 (0.234)

^aResults in columns 1 and 2 are for a 9(23)-10-1 network using either observed or predicted secondary structure information. “Balancing” parameter λ was set to zero. Results in brackets are for observed coil residues only.

Finally, neural networks trained on the turn/not-turn prediction problem displayed a strong correlation between prediction strength and prediction accuracy. An analysis of the reliability indices generated by our 9(23)-10-1 network is presented in Figure 2.

Filtered turn/not-turn predictions

Having experimented with a small number of network architectures, we found the best filtered turn/not-turn predictions were generated using a 11(4)-1 network (i.e., a single-layer net with an input window 11 residues wide plus secondary structure information in the data set). “Balancing” factor λ was set to 1, for the reasons given previously.

During the development of our method’s filtering phase, our main focus was on the choice of “state-shifting” rule. Here there were two aims:

1. to generate final turn/not-turn predictions that over- and under-predict turn at roughly equal rates (our preference is for a slightly higher rate of overprediction)
2. to generate results with a comparatively high rate of overprediction suitable for preprocessing the turn-type prediction data set (see Materials and methods).

The effect of filtering the (observed) predictions from our 9(23)-10-1 network using different state-shifting rules is illustrated in Figures 3 and 4. After analyzing these effects for different sets of network predictions, we chose the rule $T < 2$ to generate our “best” turn/not-turn results with predicted secondary structure information and the rule $NT < 1$ in all other cases. It is worth noting that whereas the choice of state-shifting rule had a significant impact on our results, the percentage of residues that had their states changed (from turn to not-turn or vice versa) by one of the four “state-flipping” rules described in Materials and methods was very small (around 3% maximum), demonstrating that the predictions were already well correlated.

Combining the best predictions with the filtering strategy outlined here, we achieved final MCCs for the turn/not-turn prediction task of 0.533 (observed secondary structure information) and 0.348 (predicted secondary structure information), see Table 5. This represents a substantial improvement upon the “baseline” MCC of around 0.2 for turn/not-turn prediction using earlier methods. The normalized better-than-random performance is 53.2 and 34.6% for observed and predicted secondary structure information, respectively (Table 6). Finally, when a displacement in prediction accuracy of only a single residue is permitted, the number of correctly-identified β -turn residues rises from 67.9 to 78.5% with observed secondary structure information and from 55.3 to 64.5% with predicted secondary structure information (see Table 7). In both cases, there is a corresponding reduction in the overprediction rate. Sample turn/not-turn predictions for a single protein chain are given in Figure 5.

Predicting β -turn types

All of the turn-type results presented in this section were generated using a simple 4(20)-1 network (i.e., a single-layer net with an input window four residues wide containing amino-acid information only) with balancing factor $\lambda = 1$. The training sets contained

Fig. 2. Prediction strength vs. prediction accuracy. The prediction accuracy ($Q_{\text{predicted}}$) plotted for all predictions above a given reliability index (RI) threshold. Results are for the 9(23)-10-1 network (with balancing parameter $\lambda = 0$). Note that with predicted secondary structure information only eight turn predictions had an $RI > 6$. These have been omitted from the graph.

Fig. 3. The effect of “state-shifting” on MCC. Results are for the 9(23)-10-1 network with balancing factor $\lambda = 0$ and observed secondary structure information.

all the data identified by turn/not-turn network number (3) in Table 5 as (possibly) belonging to a β -turn. Conflicting predictions from the four turn-type networks were resolved using the simple “winner-takes-all” strategy outlined in Materials and methods.

Statistical summaries of our turn-type prediction results, both before and after conflict resolution, are presented in Table 8. These show a clear distinction between our ability to predict types I and II (MCC greater than 0.2) and types VIII and IV (MCC less than 0.1). The same distinction applies to how well our networks perform compared to chance (Table 9); the normalized better-than-random performance exceeds 18% for types I and II, but is less than 10% for types VIII and IV.

These results are, to a considerable extent, in line with our expectations; the most numerous, well-defined types (I and II) are predicted more accurately than less-numerous type VIII and ill-defined type IV. It is worth noting that the disparity between the predictive accuracy for types II and VIII may be attributable to two factors: the smaller number of type VIII turns and the fact that type VIII turns are characterized by a high degree of conformational variability.

With respect to our strategy for resolving conflicting predictions, Table 8 shows that, with the notable exception of type IV turns, it has little impact on predictive performance. This is primarily because the number of conflicting predictions is compara-

Fig. 4. The effect of “state-shifting” on Q_{observed} and $Q_{\text{predicted}}$. Results are for the 9(23)-10-1 network with balancing factor $\lambda = 0$ and observed secondary structure information.

Table 5. *Summary of filtered turn/not-turn predictions*^a

Method	Q_{total} (%)	Q_{observed} (%)	$Q_{\text{predicted}}$ (%)	MCC
(1)	78.3 (61.3)	31.8 (34.5)	58.8 (68.2)	0.315 (0.234)
(2)	67.6 (55.5)	75.3 (79.1)	40.5 (52.0)	0.347 (0.147)
(3)	63.1 (53.4)	82.2 (85.0)	37.6 (50.5)	0.335 (0.124)
(4)	74.9 (59.8)	55.3 (59.5)	48.0 (57.3)	0.348 (0.195)

^aPredictions were generated using the following network architectures and filtering rules:

- (1) 9(23)-10-1 net
- (2) As (1) + 11(4)-1 filter net
- (3) As (2) + $NT < 1$ shift + flipping
- (4) As (2) + $T < 2$ shift + flipping.

Method number (3) was used to preprocess the data prior to turn-type prediction. Method number (4) generated our “best” turn/not-turn results. Results in brackets are for observed coil residues only.

tively low. For example, out of 2,783 predicted type I turns using observed secondary structure information, only 84 were “shifted” to another turn type by conflict resolution (of which 11 were, in fact, type I turns). However, the situation is quite different with our type IV predictions, which tend to be both weak and inaccurate; out of only 809 predicted type IV turns using observed secondary structure information, 553 were “shifted” to another turn type by conflict resolution (of which 75 were, in fact, type IV turns). In other words, the main effect of our simple conflict resolution strategy was the elimination of most our type IV predictions. Sample type I and II predictions for a single protein chain are given in Figure 6.

To sum up, it is clear that our individual turn-type predictions are less accurate than our turn/not-turn predictions, and that only types I and II are predicted with sufficient accuracy to be useful. Nevertheless, the final MCCs achieved for β -turn types I and II, 0.296 (observed) and 0.219 (predicted) for type I, and 0.317 (ob-

Table 6. Performance relative to random, turn/not-turn predictions ^a

	Correct ($p + n$)	R_{total}	S (%)
Observed	37,527	28,414.3	53.2
Predicted	34,114	28,065.1	34.6
Observed coil = turn	33,588	24,342.6	43.6
Predicted coil = turn	29,929	23,720.8	28.5

^aResults are for our “best” turn/not-turn networks (number (4) in Table 5). Column 2 records the number of correctly identified residues ($p + n$), Column 3 the anticipated number of residues correctly identified by chance (R_{total}), and Column 4 the normalized percentage better than random (S). The “observed coil = turn” row records the scores achieved if all observed coil residues are predicted as turn and all noncoil residues are predicted as not-turn. The “predicted coil = turn” row records the scores achieved if all predicted coil residues are predicted as turn and all noncoil residues are predicted as not-turn.

Table 7. *Displaced turn/not-turn predictions*^a

Displacement	Observed Q_{observed} (%)	Observed $Q_{\text{predicted}}$ (%)	Predicted Q_{observed} (%)	Predicted $Q_{\text{predicted}}$ (%)
0	67.9	62.2	55.3	48.0
± 1	78.5	73.0	64.5	60.1
± 2	83.0	78.6	70.2	67.2

^aResults are for our “best” turn/not-turn networks (number (4) in Table 5).

served) and 0.253 (predicted) for type II, compare favorably with those of around 0.2 achieved by other methods when applied to the (apparently easier) turn/not-turn problem. Moreover, when a displacement in prediction accuracy of only a single residue is permitted, the number of correctly-identified turns of type I and II rises above 60% with both observed and predicted secondary structure information (see Tables 10, 11).

PKGVRVAVLLGPPGAGKGTQAPKLAKNFVCVCHLATGDMRLAMRASVAGSELG
 -----EEEE-----HHHHHHHHHHHH-----EEEEHHHHHHHHHHHH-----HHHH
 -----eeee-----hhhhhhhhhhhh-----hhhhhhhhhhhh-----hh
 TTTT
 tttt tttt
 1211366311222113678899887511111113578998775333357
 KLKATMDAGKLVSDVMVLELIEKNLETPPCKNGFLLDGFPRPTVRQAEMLD
 HHHHHHH-----HHHHHHHHHHHH-----HH-----EEEE-----HHHHHHHH
 hhhhhhh-----hhhhhhhhhhhhhh-----eee-----hhhhhhhh
 TTTT TTTTTT
 tttt tttt
 8888763311113578888888876310221111101111367889999
 DLMEKRKEKLDSVIEFSPIDSLIRRITGRLIHPQSGRSYHEEFNPPEP
 HHHHHH-----EEEEEE-----HHHHHHHHHH-----EE-----EE-----
 hhhhh-----eeeeeee-----hhhhhhhh-----eee-----
 TTTTT TTTTTTTTTT
 tttt ttttt tttttt
 988651122113565311113678765331110222013311122222
 MKDDITGEPLIRRSDDNKKALKIRLEAYHTQTTPLVYYSKRGISAI
 -----EE-----HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH-----EEEE--
 -----hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh-----eee--
 TTTTTTTT TT
 tttttttttttttttt tt
 2222222100001001578999988887766677887633111331112
 SQTPDVVFASILAAFSKATS
 ---HHHHHHHHHHHHHHHH-----
 ---hhhhhhhhhhhhhh-----
 TT
 tt
 10136889999888753111

Fig. 5. Sample turn/not-turn predictions. Predictions are for chain 1ak2 (220 residues) using predicted secondary structure information. The predictions are grouped in blocks of 50 residues. For each block: Row 1 records the amino acid sequence; Row 2 records the observed secondary structure (H = helix, E = strand, - = coil), as provided by the SSTRUC program; Row 3 records the predicted secondary structure (h = helix, e = strand, - = coil), as provided by the PHDsec program; Row 4 records the observed turn residues; Row 5 records the turn/not-turn predictions, and Row 6 records the reliability index for each turn/not-turn prediction. The statistical accuracy of the predictions for chain 1ak2 is as follows: $Q_{\text{total}} = 88.2\%$, $Q_{\text{observed}} = 85.0\%$, $Q_{\text{predicted}} = 63.0\%$, and $\text{MCC} = 0.662$.

Table 8. Summary of turn-type predictions^a

Turn type	Q_{total} (%)	Q_{observed} (%)	$Q_{\text{predicted}}$ (%)	MCC
I	91.2 (90.8)	46.6 (48.0)	13.9 (13.6)	0.219 (0.220)
II	95.5 (95.4)	58.4 (60.3)	12.2 (12.1)	0.253 (0.256)
VIII	95.7 (94.9)	18.0 (21.5)	3.3 (3.2)	0.062 (0.067)
IV	96.8 (95.3)	2.2 (9.4)	9.3 (10.1)	0.033 (0.073)

^aResults are for our final turn-type predictions (i.e., after the resolution of conflicting predictions), with those generated prior to conflict resolution given in brackets.

Discussion and conclusion

The β -turn prediction method described in this paper yields predictions that are significantly more accurate than those of earlier methods. Taking the Matthews correlation coefficient as the most robust single-figure performance metric, we achieved final MCCs for the turn/not-turn prediction task of 0.533 (with observed secondary structure information) and 0.348 (with predicted secondary structure information) compared with the 0.202 achieved by McGregor et al. (1989) and 0.204⁵ by Zhang and Chou (1997). With respect to the prediction of individual turn types, no MCCs are available for other methods; however, it is unlikely that they would match the levels achieved here for turn types I and II, 0.296 (type I) and 0.317 (type II) with observed secondary structure information, 0.219 (type I), and 0.253 (type II) with predicted secondary structure information.

This improvement in predictive accuracy is primarily attributable to the addition of secondary structure information to our protein data, although the use of more local information (i.e., a wider input window) and the development of simple filtering strategies also proved beneficial. It is clear from these results that incremental improvements in predictive accuracy will be possible as more effective secondary structure prediction methods are developed.

Although our β -turn predictions are not as accurate as standard three-state secondary structure predictions, they are nevertheless

⁵Zhang and Chou (1997) do not record MCCs for their β -turn predictions; this figure was calculated from the raw data presented in their paper.

Table 9. Performance relative to random^a

Turn type	Correct ($p + n$)	R_{total}	S (%)
I	40,685 (40,513)	39,810.8 (39,617.4)	18.1 (17.9)
II	42,637 (42,565)	42,171.1 (42,084.8)	18.9 (18.9)
VIII	42,710 (42,358)	42,620.4 (42,251.7)	4.5 (4.5)
IV	43,185 (42,518)	43,146.8 (42,351.4)	2.6 (7.3)

^aResults are for our final turn-type predictions (i.e., after the resolution of conflicting predictions), with those generated prior to conflict resolution given in brackets. Column 2 records the number of correctly identified residues ($p + n$), column 3 the anticipated number of residues correctly identified by chance (R_{total}) and column 4 the normalized percentage better than random (S).

PKGVRAVLLGPPGAGKGTQAPKLAKNFCVCHLATGDMRAMVASGSELGK				
-----EEEE-----HHHHHHHHHHHH-----EEEEHHHHHHHHHH-----HHHH				
-----e-----h-----h-----h-----h-----h-----h-----h-----h				
	B			
b	ab			
KLKATMDAGKLVSDVMLELIEKNLETPPCCKNGFLDGFRTVRQAEMLD				
HHHHHH-----HHHHHHHHHHHH-----HH-----EEEE-----HHHHHHHH				
hhhhhh-----h-----h-----h-----h-----h-----h-----h-----h				
	A			
	a ab ab			
DLMEKRKEKLDVIEFSIPDSLLIRITGRLIHPQSGRSYHEEFNPPKEP				
HHHHHH-----EEEE-----HHHHHHHHHH-----EE-----EE-----				
hhhhh-----e-----h-----h-----h-----h-----h-----h-----h				
	A		A	B
	aab		aa	a
MKDDITGEPLIRSDDNKALKIRLEAYHTQTTPLEVEYYSKRGHSAIDA				
-----EE-----HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH-----EEEE-----				
-----h-----h-----h-----h-----h-----h-----h-----h-----h				
	A			A
aaaab a aa			b	a a
SQTPDVVFASILAAFSKATS				
-----HH				
-----h-----h-----h-----h-----h-----h-----h-----h-----h				

Fig. 6. Sample type I and II turn predictions. Predictions are for chain lak2 (220 residues) using predicted secondary structure information. The predictions are grouped in blocks of 50 residues. For each block: Row 1 records the amino acid sequence; Row 2 records the observed secondary structure (H = helix, E = strand, - = coil), as provided by the SSTRUC program; Row 3 records the predicted secondary structure (h = helix, e = strand, - = coil), as provided by the PHDsec program; Row 4 records the observed location of turn-types I (first residue of tetramer labeled “A”) and II (first residue of tetramer labeled “B”), and Row 5 records the predicted location of turn-types I (first residue of tetramer labeled “a”) and II (first residue of tetramer labeled “b”) after conflict resolution.

of sufficient accuracy to be useful in the ab initio determination of protein structure. When credit is given to predictions displaced by only a single residue, the number of turn residues correctly identified rises to 78.5% (with observed secondary structure information) and 64.5% (with predicted secondary structure information). Moreover, between 60–65% of β -turn types I and II are correctly identified (with both observed and predicted secondary structure information) when predictions displaced by a single residue are accepted. In all these cases, the corresponding rate of overprediction is comparatively modest.

Finally, these results illuminate both the strengths and the limitations of neural networks when applied to pattern classification tasks within the field of biochemistry. When there is an abundance of data belonging to well-defined classes, neural networks can be

Table 10. Displaced turn predictions, type I^a

Displacement	Observed Q_{observed} (%)	Observed $Q_{\text{predicted}}$ (%)	Predicted Q_{observed} (%)	Predicted $Q_{\text{predicted}}$ (%)
0	49.4	21.1	46.6	13.9
± 1	64.7	30.0	61.3	20.0
± 2	70.6	37.5	67.4	26.3

^aResults are for our final type I predictions (after conflict resolution).

Table 11. Displaced turn predictions, type II^a

Displacement	Observed Q_{observed} (%)	Observed $Q_{\text{predicted}}$ (%)	Predicted Q_{observed} (%)	Predicted $Q_{\text{predicted}}$ (%)
0	60.7	17.7	58.4	12.2
± 1	63.5	19.7	61.7	13.7
± 2	66.3	22.3	65.4	13.7

^aResults are for our final type II predictions (after conflict resolution).

extremely powerful. However, neural nets perform poorly when the available data are sparse (e.g., type VIII β -turns) or ill-defined (e.g., type IV β -turns). Moreover, it is clear that our method is highly dependent on the accuracy of the secondary structure predictions it exploits, and that any method that aims to incrementally build upon these results (for example, to predict higher-order protein structures) is liable to experience a further degradation in predictive accuracy. Ultimately, the development of higher-order prediction methods that are genuinely useful must include a substantial improvement in the accuracy of secondary structure prediction.

Note: The β -turn prediction program developed in the course of this research is available on the World Wide Web at the following address: <http://www.biochem.ucl.ac.uk/bsm/btpred/index.html>.

Acknowledgments

We would like to thank Dr. Gail Hutchinson and Dr. Burkhard Rost for helping us to acquire and prepare our training data, and Dr. Roman Laskowski for useful discussions. This work was supported by the EPSRC/BBSRC Bioinformatics Programme.

References

- Chandonia JM, Karplus M. 1996. The importance of larger data sets for protein secondary structure prediction with neural networks. *Protein Sci* 5:768–774.
- Chou KC, Blinn JR. 1997. Classification and prediction of β -turn types. *J Protein Chem* 16(6):575–595.
- Chou PY, Fasman GD. 1974. Conformational parameters for amino acids in helical, β -sheet and random coil regions calculated from proteins. *Biochemistry* 13:211–223.
- Gorse D, Shepherd AJ, Taylor JG. 1997. The new ERA in supervised learning. *Neural Networks* 10(2):343–352.
- Hutchinson EG, Thornton JM. 1994. A revised set of potentials for β -turn formation in proteins. *Protein Sci* 3:2207–2216.
- Lewis PN, Momany FA, Scheraga HA. 1973. Chain reversals in proteins. *Biochim Biophys Acta* 303:211–229.
- McGregor MJ, Flores TP, Sternberg MJE. 1989. Prediction of β -turns in proteins using neural networks. *Protein Eng* 2(7):521–526.
- Møller M. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6(4):525–533.
- Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. 1997. CATH: A hierarchic classification of protein domain structures. *Structure* 5:1093–1108.
- Qian N, Sejnowski TJ. 1988. Predicting the secondary structure of globular proteins using neural network models. *J Mol Biol* 202:865–884.
- Riis SK, Krogh A. 1996. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J Comput Biol* 3:163–183.
- Rose GD. 1978. Prediction of chain turns in globular proteins on a hydrophobic basis. *Nature* 272:586–590.
- Rost B, Sander C. 1993. Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol* 232:584–599.
- Rost B, Sander C. 1994. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins* 19:55–72.
- Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, eds. *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1. Cambridge, MA: MIT Press. pp 318–362.
- Shepherd AJ. 1997. *Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons*. London: Springer.
- Wilmot CM, Thornton JM. 1988. Analysis and prediction of the different types of β -turn in proteins. *J Mol Biol* 203:221–232.
- Yang AS, Hitz B, Honig B. 1996. Free energy determinants of secondary structure formation: III. β -turns and their role in protein folding. *J Mol Biol* 259:873–882.
- Zhang CT, Chou KC. 1997. Prediction of β -turns in proteins by 1-4 and 2-3 correlation model. *Biopolymers* 41(6):673–702.
- Zimmerman SS, Scheraga HA. 1977. Local interactions in bends of proteins. *Proc Natl Acad Sci USA* 74:4126–4129.