BIOINFORMATICS

Vol. 17 Suppl. 1 2001
Pages S234–S242

Improved prediction of the number of residue contacts in proteins by recurrent neural networks

Gianluca Pollastri[1], *Pierre Baldi*[1, 2,*], *Pietro Fariselli*[3] *and Rita Casadio*[3]

[1]*Department of Information and Computer Science, Institute for Genomics and Bioinformatics, University of California, Irvine, Irvine, CA 92697-3425, USA,* [2]*Department of Biological Chemistry, College of Medicine, University of California, Irvine, and* [3]*CIRB Biocomputing Unit and Lab. of Biophysics, Dept. of Biology, University of Bologna, Bologna, 40126, Italy*

Received on February 6, 2001; revised and accepted on March 21, 2001

ABSTRACT

Knowing the number of residue contacts in a protein is crucial for deriving constraints useful in modeling protein folding, protein structure, and/or scoring remote homology searches. Here we use an ensemble of bi-directional recurrent neural network architectures and evolutionary information to improve the state-of-the-art in contact prediction using a large corpus of curated data. The ensemble is used to discriminate between two different states of residue contacts, characterized by a contact number higher or lower than the average value of the residue distribution. The ensemble achieves performances ranging from 70.1% to 73.1% depending on the radius adopted to discriminate contacts (6Å to 12Å). These performances represent gains of 15% to 20% over the base line statistical predictors always assigning an aminoacid to the most numerous state, 3% to 7% better than any previous method. Combination of different radius predictors further improves the performance.

Server: http://promoter.ics.uci.edu/BRNN-PRED/

Contact: pfbaldi@ics.uci.edu

INTRODUCTION

A major challenge in molecular biology is the elucidation of the functional properties of proteins in terms of structural and dynamical features. In this context, the fundamental problem is posed by the process of protein folding during which the protein settles into a stable and well definite three-dimensional structure. Its knowledge is valuable in determining the structure to function relationship. Moreover it justifies the considerable effort being expended to bridge the gap between the amount of 3D structures known with atomic resolution and the overwhelming quantity of amino acid sequence data (Sanchez and Sali, 1998).

One approach towards predicting the structure of a protein is to predict a number of key attributes, in particular secondary structure, solvent accessibility, and number of contacts. Deriving a good contact map from the primary sequence and these attributes is in fact emerging as a key possible strategy for solving the structure prediction problem. For most of these attributes, machine learning methods in general, and more specifically neural network approaches, have proven to be particularly effective. For instance, the best secondary structure predictors today are neural-network-based with performance in the 75-80% range and continuing to improve year after year (Baldi and Brunak, 2001). In this work, we develop recurrent neural network methods for the prediction of residue contacts.

Knowing the correct positions of residue contacts in proteins has proven to be extremely useful in determining the three-dimensional structure of a given protein, as demonstrated in the CASP3 and CASP4 competitions [http://predictioncenter.llnl.gov/] (Ortiz et al., 1999). The number of stabilizing contacts that residues make in the protein folded globule (see Dill (1999) for a review) is a fundamental aspect of protein structure well worth predicting. In particular, this number could be used to "clean-up" noisy contact map predictions based on primary sequence and secondary structure information. Furthermore, when remote homology is searched, it is profitable to derive a surface potential from the distribution of contact numbers for each residue. This is computed by implementing the inverse of the Boltzman rule (Flokner et al., 1995), or by using the notion of contacts among residues to improve existing threading algorithms (Olmea et al., 1999).

Based on the idea that less exposed residues are preferentially involved in hydrophobically driven chain compaction, solvent accessibility has been routinely used

*To whom all correspondence should be addressed.

also to evaluate the number of residue contacts. In order to simulate the hydrophobic collapse in model proteins, the number of residue contacts is chosen as the inverse measure of the residue solvent accessibility and, in the case of simple lattice protein models, it is the only source of interaction (Sali et al., 1994). In Fariselli and Casadio (2000) it was shown that although a strict connection between accessibility and contact number is commonly accepted, residue surface accessibility has a different distribution from the number of residue contacts, so that residue classification may be different depending on which property is highlighted. Thus at least a partial separation of the problems of predicting number of contacts and solvent accessibility is required.

In an off-lattice context, the number of contacts for each residue is computed inside a spherical cut-off centered on each residue and by counting the number of residues falling inside the sphere (Flokner et al., 1995). In the last few years different attempts to predict contacts (Shindyalov et al., 1994; Olmea and Valencia, 1997; Fariselli and Casadio, 1999) and distances among residues in proteins (Aszodi et al., 1995; Lund et al., 1997; Gorodkin et al., 1999) have been made with some degree of success. In Fariselli and Casadio (2000), a feedforward neural network approach with a local window was developed to discriminate between two different states of residue contacts, characterized by a contact number higher or lower than the average value of the residue distribution. For a contact radius of 6.5Å, this approach achieved a performance of 69% correct prediction, 12% above the level of a simple "base line"classifier. By definition, the base line classifier always outputs the most frequent category for each amino acid independently of its environment (Richardson and Barlow, 1999).

Here we first extract a large curated data set of contact information from the PDB database and build a set of corresponding profiles. We compute detailed contact statistics on this set and, in particular, examine the effect of contact radius, ranging from 6Å to 12Å. We then develop and apply a class of recurrent neural network architectures capable of partially capturing long-ranged information to the problem of contact prediction.

DATA PREPARATION

As always the case in machine learning approaches, the starting point is the construction of a well-curated data set. The data sets used here were extracted from the PDB_select list (Hobohm et al., 1992) of June 2000. The list of structures and additional information can be obtained from the ftp site: ftp://ftp.embl-heidelberg.de/pub/databases. The redundancy threshold to eliminate a sequence is based on the distance derived in Abagyan and Batalov (1997) and corresponds to a sequence identity of roughly 22% for long alignments, and higher for shorter ones. This set was further reduced by excluding those chains whose backbone is interrupted. We ran Kabsch and Sander's DSSP program (Kabsch and Sander, 1983) on all the PDB files in the PDB_ select list, and excluded the ones on which DSSP crashed. The final set consisted of 1086 protein chains containing 166750 residues.

Table 1. Average and range of number of contacts for each radius across all amino acids. Avg = average over all aminoacids. Min = lowest average number for any aminoacid with corresponding amino acid in brackets. Max = highest average number for any amino acid with corresponding amino acid in brackets.

	6Å	8Å	10Å	12Å
Avg	5.33	9.55	16.93	27.20
Min	4.21(P)	8.36(E)	14.41(E)	22.97(E)
Max	6.08(C)	11.50(C)	20.27(C)	32.08(I)

The number of inter-residue contacts for each residue in the data set is computed by defining a spherical volume centered on its C_α atom, with a given radius R Å, and counting the number of additional C_α atoms contained in the sphere. For a given radius R, we computed the average number of contacts for each amino acid over the entire set (Table 1). Each residue in a chain was then assigned to class 0 if the number of neighbors within the radius R was less than average, and to class 1 if above average. The process was repeated for radiuses of 6, 8, 10 and 12 Angstrom.

In order to perform three-fold cross-validation experiments, the data was split evenly into three subsets, each containing 362 proteins (Table 2). In all 3 subsets the two classes are distributed almost evenly. Class 0 is slightly more numerous than class 1 for all four radiuses, ranging from a minimum of 50.91% for 10Å to a maximum of 52.12% for 8 Å, over the total set. This effect is to be expected since the possible contact values below the average have a more restricted range than the values above the average.

The correlations of the numbers of contacts in the 4 different categories are shown in Table 3. There are high correlations between the 8Å, 10Å and 12Å categories, while the 6Å category is less correlated to the others. For each radius, the range, average, and per amino acid distribution of the number of contacts is displayed in Table 1 and Figures 1 and 2.

It is well known that evolutionary information in the form of multiple alignments and profiles significantly improves the accuracy, for instance, of secondary structure prediction methods. This is because the secondary structure of a family is more conserved than the primary amino

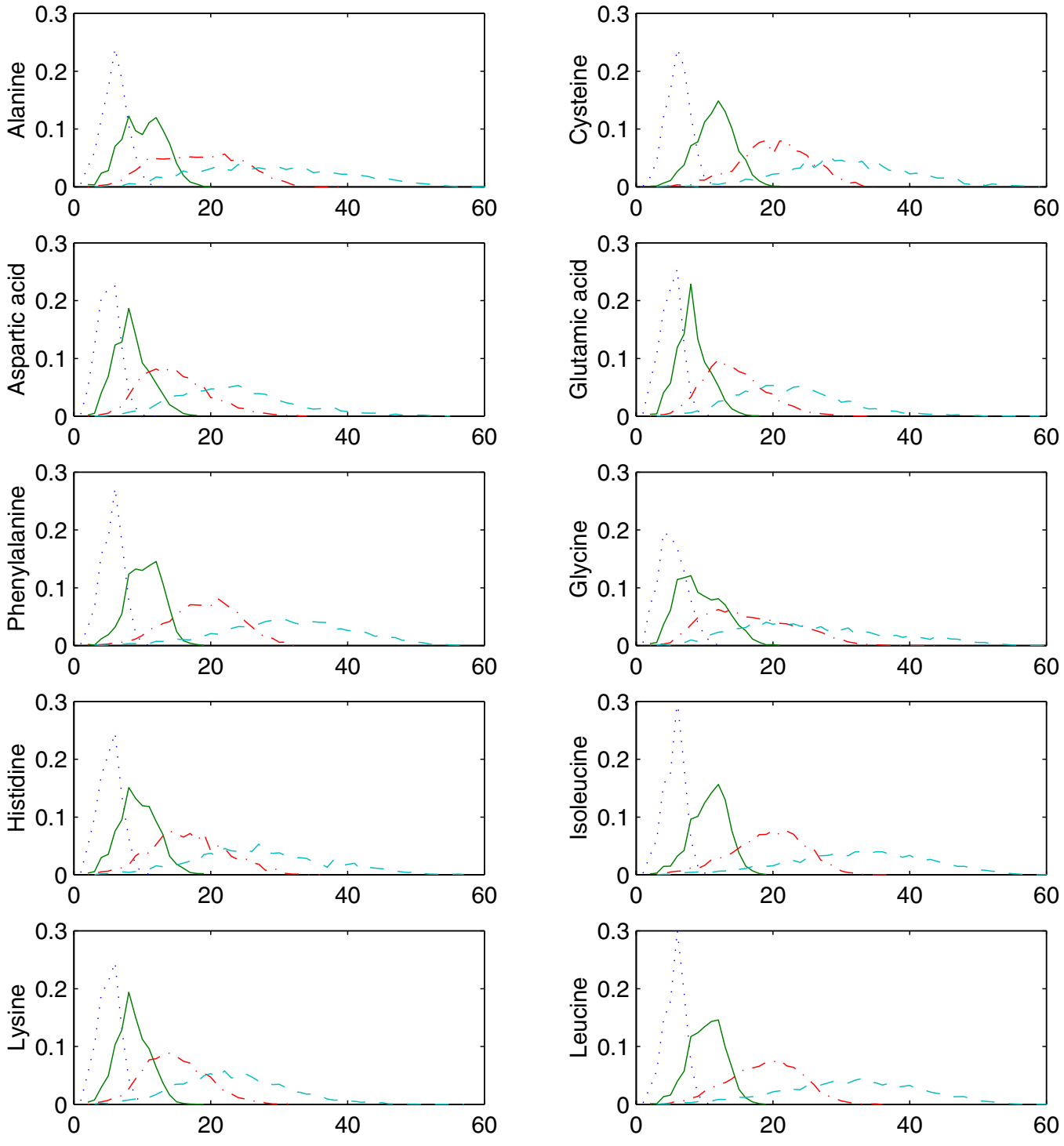

Fig. 1. Distribution of number of contacts for amino acids A to L in alphabetical order: dotted=6Å, solid=8Å, dashdot=10Å, dash=12Å [x-axis = number of contacts, y-axis = probability].

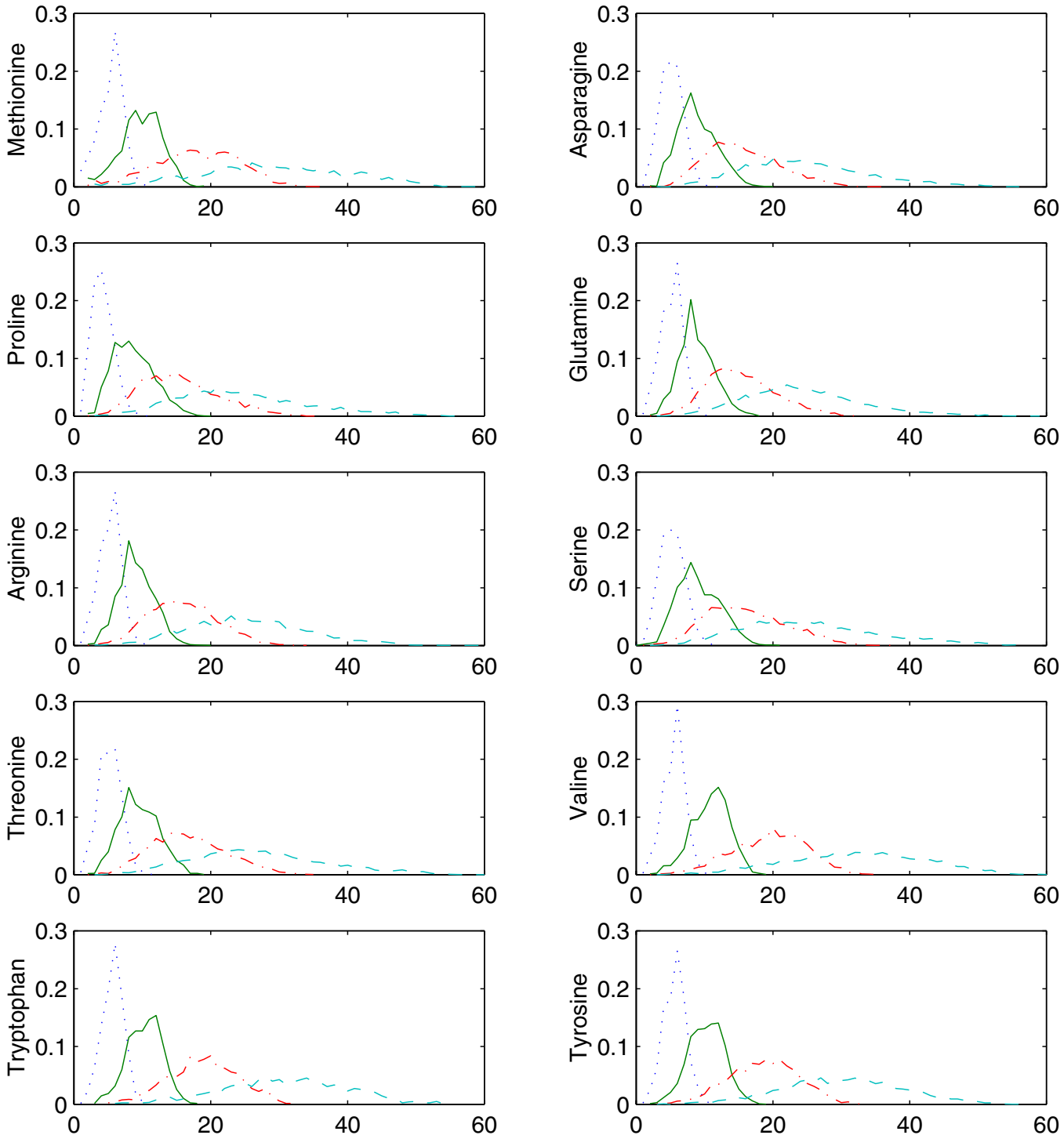

Fig. 2. Distribution of number of contacts for amino acids M to Y in alphabetical order: dotted=6Å, solid=8Å, dashdot=10Å, dash=12Å [x-axis = number of contacts, y-axis = probability].

Table 2. Three-fold cross validation subset statistics with number of amino acids in each class.

	Class	6Å	8Å	10Å	12Å
Total set	0	85119	86906	84886	86401
166750 AA	1	81631	79844	81864	80349
Subset 1	0	28415	29344	28675	29357
55859 AA	1	27444	26515	27184	26502
Subset 2	0	28072	28008	27430	27860
54355 AA	1	26283	26347	26925	26495
Subset 3	0	28632	29554	28781	29184
56536 AA	1	27904	26982	27755	27352

Table 3. Correlations between the number of contacts for different radius values.

	6Å	8Å	10Å	12Å
6Å	1.0	0.63	0.53	0.46
8Å		1.0	0.86	0.79
10Å			1.0	0.92
12Å				1.0

acid sequence. Similar effects hold for the prediction of contact numbers. In Fariselli and Casadio (2000), an improvement of 3% was reported as a result of using profiles instead of single sequences.

In the case of multiple sequence inputs, the sequence profiles were generated using the BLAST program (Altschul et al., 1990) with standard default parameters (E=10.0, BLOSUM62 matrix) run on the standard NR (non-redundant) database. The version used was available on line in October 1999 and contained approximately 420,000 protein sequences. Every sequence in the alignment is assigned a weight proportional to the information the sequence carries with respect to the unweighted profile. A weighted profile matrix is then compiled and used as input for the system.

RECURRENT NEURAL NETWORK ARCHITECTURES

Feedforward neural networks have been one of the major machine-learning tools used in protein structure prediction problems, ranging from the prediction of secondary structure to the number of contacts. The major weakness of feedforward neural networks, however, is the use of a local input window of fixed size that cannot provide any access to long-ranged information. Networks for contact prediction, for instance, have windows of size 1-15. Larger

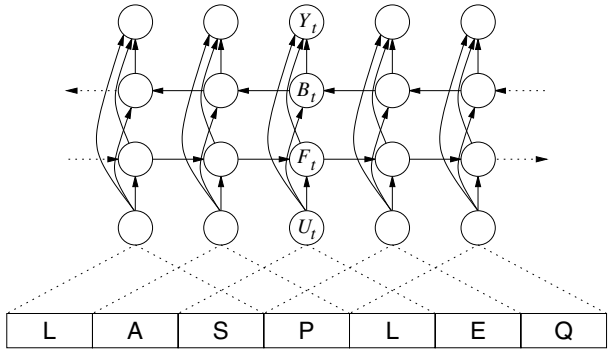

Fig. 3. A bidirectional graphical model.

windows usually do not work, in part because the corresponding increase in the number of parameters leads to overfitting. Increase in the number of parameters, however, is not necessarily the main obstacle per se because data is becoming abundant and techniques, such as weight sharing, can be used to mitigate the risk of overfitting. The main problem is that long-ranged signals are very weak compared to the additional "noise" introduced by a larger window. Thus larger windows tend to dilute sparse information present in the input that is relevant for the prediction.

In Baldi et al. (1999) and Baldi et al. (2000) BRNNs (Bidirectional Recurrent Neural Networks) have been proposed as a class of recurrent neural network architectures that can address some of the limitations of simple feedforward networks. The BRNN approach can first be described in terms of the Bayesian network shown in Figure 3. In terms of graphical models, BRNNs can be viewed as sequential Markovian models for the translation of an input sequence into an output sequence. The translation is mediated by: (1) an input layer where the amino acid sequence is presented; (2) two hidden-state Markov chains: a feedforward chain (as in HMMs or IOHMMs) and a backward chain that can transmit information in both directions along the sequence, and between the input and the output sequence; and (3) an output layer consisting here of classification units. Because inference in these Bayesian networks is too slow, we replace the diagram by a recursive neural network, using the techniques described in detail in Baldi et al. (1999) and Baldi and Brunak (2001).

To be more specific, the graphical model is implemented using local feedforward networks, resulting in the BRNN architecture described in Figure 4. In this architecture, the output decision or classification is determined by three components. There is first a central component associated with the local window at the location of the current prediction. This component of the architecture with its hid-

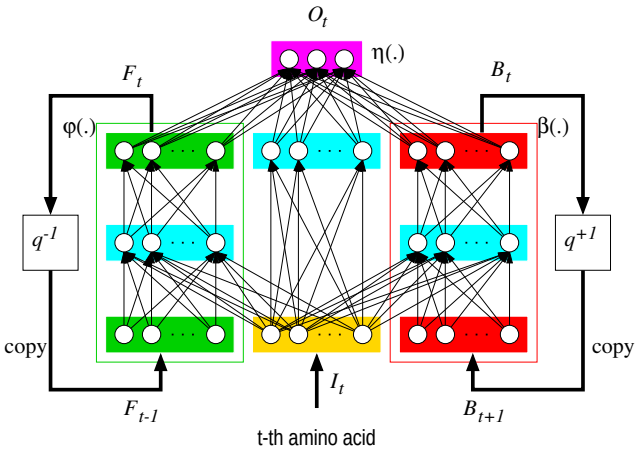

Fig. 4. A BRNN architecture with a left (forward) and right (backward) context associated with two recurrent networks (wheels).

den layer is very similar to the standard feedforward networks used in computational molecular biology applications. The main difference is the contribution of the left and right contexts. The left and right context are produced by two similar recurrent networks which intuitively can be thought in terms of two "'wheels" that are rolled along the protein chain, starting from the N and the C terminus. All the weights of the BRNN architecture, including the weights in the recurrent wheels, can be trained in a supervised fashion from examples by a generalized form of gradient descent or backpropagation through time, or rather space, because of the forward and backward nature of the chains. Architectural variants can be obtained by changing the size of the input windows, the size of the window of hidden states considered to determine the output, the number of hidden layers, the number of hidden units in each layer and so forth. As in standard secondary-structure and other protein prediction architectures, we use sparse encoding for the 20 amino acids.

In what follows, we use the following notation:
Ct = size of semi-window of context states considered by the output network;
NFB = number of output units in the left (forward) and right (backward) context networks (wheels);
NH = number of hidden units in the output network;
NHT = number of hidden units in the context networks.

In the contact prediction application there is a single logistic output unit that estimates the probability that the contact number is higher or lower than the average in the center of the corresponding input location. The architecture is trained by back-propagation on the relative entropy between the output and target probability distributions.

BRNNs have been used for secondary structure prediction and to develop a web server called SSpro available at http://promoter.ics.uci.edu/BRNN-PRED/ currently ranked as one of the best by an independent web-based tester implemented by B. Rost [http://cubic.bioc.columbia.edu/~eva/]. They have also been used for the prediction of amino acid partners in beta sheets (Baldi et al., 2000). In Baldi et al. (1999) evidence is provided that these architecture extend the range over which information can be effectively captured with respect to feed-forward neural network, up to an effective window size of about 30 amino acids in the case of secondary structure prediction.

Table 4. Typical training set statistics taken from the first set.

Sets	Class	6Å	8Å	10Å	12Å
Train	0	56704	57562	56211	57044
Train	1	54187	53329	54680	53847
Test	0	28415	29344	28675	29357
Test	1	27444	26515	27184	26502

EXPERIMENTS AND RESULTS

A cross validation procedure was adopted by splitting the available data into three subsets of comparable size (Table 2). The total number of amino acids in each cross validation experiment is approximately 165,000: 110,000 used as a training set and 55,000 as a test set (Table 4). We used a network of 16 workstations for training and testing, roughly equivalent to one year of CPU-time, excluding preliminary experiments.

Learning is by generalized gradient descent using the relative entropy error function (Baldi et al., 1999). In a typical case, we used a hybrid between on-line and batch training, with 300 batch blocks (2-3 proteins each) per training set. Thus weights are updated 300 times per epoch after each block. The learning rate per block is initially set at about 2.7×10^{-4}, corresponding to the number of blocks divided by ten times the number of residues ($0.1 \times 300/110000$), and is progressively decreased. The training set is also shuffled at each epoch, so the error is not decreasing monotonically. There is no momentum term or weight decay. When the error does not decrease for 50 consecutive epochs, the learning rate is divided by 2 and training is restarted from the lowest error model. Training stops after 8 or more reductions, corresponding to a learning rate that is 256 times smaller than the initial one, which usually happens after 1500-2500 epochs.

Preliminary tests were conducted with a number of different BRNNs architectures. We finally focused on 7 BRNNs, with the same structure as those used in the early version of the SSpro software (Baldi et al., 1999) for protein secondary structure prediction. The basic

Table 5. Total number of weights and size parameters of the 7 BRNNs models. Ct = size of semi-window of context states considered by the output network. NFB = number of output units in the left and right context networks. NH = number of hidden units in the output network. NHT = number of hidden units in the context networks.

model #	Weights	Ct	NFB	NH	NHT
0	2241	3	8	11	9
1	1959	2	9	11	8
2	3009	3	12	11	9
3	2615	3	12	9	9
4	4232	3	15	12	13
5	4896	3	17	12	15
6	5430	3	17	14	15

Table 6. Three-fold cross validation results obtained with several BRNNs and the corresponding ensemble on the test set. Performance results expressed in percentages of correct prediction (Q2). Ens = ensemble of models in a given radius category. Comb = combination of 4 ensembles associated with different radius categories. Filter = bare filter applied to 4 ensembles associated with different radius categories.

model #	6Å	8Å	10Å	12Å
0	71.20%	68.91%	69.85%	72.06%
1	71.20%	68.97%	70.37%	71.95%
2	70.78%	68.45%	70.11%	72.10%
3	70.50%	68.67%	70.15%	72.05%
4	69.51%	67.57%	69.38%	71.80%
5	69.53%	67.36%	68.65%	71.96%
6	69.02%	66.42%	69.36%	72.42%
Ens	72.54%	70.09%	71.20%	73.03%
Comb	72.64%	70.28%	71.44%	73.27%
Filter	72.51%	70.08%	71.17%	73.11%

parameters of each architecture are given in Table 5. The number of parameters in each architecture ranges from 1959 to 5430. The 7 architectures were combined by simple averaging of the outputs into an ensemble predictor. For a given radius category, each ensemble is the average of 21 predictors (7 networks $\times$ 3 cross validations subsets).

Several indices can be used to score the efficiency (Baldi et al., 2000) of the algorithm. Here we use Q2, the number of correctly predicted residues divided by the total number of residues, and the Matthew's correlation coefficient. The results of three-fold cross validation, corresponding to $3 \times 4 \times 7 = 84$ tests, for each one of the 7 BRNNs and for the ensemble are summarized in Table 6 for the test sets. For completeness, we also report the difference of performances between training and test sets in Table 7.

Overall, compact models tend to show better performance. Larger models perform worse because they overfit

Table 7. Three-fold cross validation results obtained with several BRNNs and the corresponding ensemble. Performance differences between the training set and the tests set, expressed in percentages of correct prediction. Ens = ensemble of models in a given radius category.

model #	6Å	8Å	10Å	12Å
0	2.76%	2.30%	1.54%	1.12%
1	2.41%	1.99%	1.00%	1.17%
2	4.22%	3.69%	1.88%	1.79%
3	4.34%	3.17%	2.20%	1.35%
4	7.13%	6.29%	3.26%	2.43%
5	7.79%	7.11%	6.27%	2.35%
6	8.75%	8.95%	4.64%	1.67%
Ens	6.10%	5.56%	3.38%	1.81%

the training set. The effect of overfitting is considerable in the 6Å and 8Å categories, moderate for 10Å, and small for 12Å. Although large models sometimes have significantly poorer performance, they still prove to be useful when combined in an ensemble. In all cases, the ensemble architecture gives a sizeable improvement over each individual architecture.

The ensemble predictor achieves a performance of 73.03% correct prediction in the 12Å category, with a correlation coefficient of 0.45. The ensemble predictor for each category performs considerably better than the simple base-line predictor that always assigns a residue to its most abundant class independently of its environment (Richardson and Barlow, 1999), ranging from a gain of 15.5% for the 6Å ensemble to 20.4% for 12Å. The best previously known predictor (Fariselli and Casadio, 2000), trained and tested only on a 6.5Å radius data set, achieved a performance of 69%, 12% better than the corresponding base line predictor. Here, in the 12Å category we improve over this performance by over 4% in terms of Q2 and 8% additional points over the base line predictor. In the 6Å category, closer to the one used in (Fariselli and Casadio, 2000), the gain in Q2 exceeds 3.5% with a similar additional gain over the base line.

At least two reasons ought to be considered to explain performance differences across the four radius categories. First the performance of the base line predictors decreases with radius size. This particularly affects the 6Å predictor, whose base level is 3% higher than the others. Second, as the radius is increased, the total length of the chain becomes an increasingly relevant piece of information. The average number of contacts in the 12Å dataset is comparable with the length of short proteins, making it less likely or even impossible sometimes to have residues belonging to class 1. Isoleucine, for instance, requires 33 contacts to be classified as 1, which is of course impossible in proteins shorter than 34 residues, and

Table 8. Comparison to base line predictor. Q2 percentage measure. Ens = ensemble of models in a given radius category. Comb = combination of ensembles across categories. Base = base line predictor which outputs the most numerous class for each amino acid. Diff = difference in Q2 between Comb and Base.

Q2	6Å	8Å	10Å	12Å
Ens	72.54%	70.09%	71.20%	73.03%
Comb	72.64%	70.28%	71.44%	73.27%
Base	57.01%	54.11%	52.86%	52.66%
Diff	15.63%	16.17%	18.58%	20.61%

Table 9. Same as previous table but using correlation measure.

corr	6Å	8Å	10Å	12Å
Ens	0.452	0.400	0.424	0.460
Comb	0.454	0.404	0.429	0.465
Base	0.195	0.119	0.063	0.051

unlikely for proteins that are just slightly longer. Tests to study the capabilities of the models to capture long-ranged information, similar to those carried in Baldi et al. (1999) and not reported here, reveal that the signal of the protein terminus is in fact propagated beyond 70 amino acids in the 12Å system by the BRNNs architectures. This signal implicitly provides a sense of protein size during the classification process. No similar effect can be found in the 6Å architectures, where the average number of contacts is small.

It is natural to wonder whether performance could be further improved by combining predictors across the four radius categories. Thus we can combine the previous ensembles using a small BRNN (a small feedforward neural network gives similar results) with parameters Ct = 2, NFB = 3, NH = 4, and NHT = 3. To avoid retraining on the same training set, we perform a two-fold cross validation on each of the 3 subsets of the previous cross validation. The results (Comb) are reported in the last row of Table 6. Each number is the average of 6 different values, since each of the 3 subsets of the previous cross validation experiment is split into 2, and the 2 resulting subsets are used alternatively as test and training sets in this experiment, yielding a total of $6 \times 4 = 24$ numbers. The improvements obtained by pooling different radius categories range from 0.1% for 6Å to 0.24% for the 10Å and 12Å categories.

To make sure that these improvements are due to the combination of diverse information and *not* to a filtering effect associated with the additional BRNN used in the combination, we also used the same BRNN architecture as a filter for each single-category predictor (Filter in Table 6). The latter simple output filtering approach gives results that are extremely similar to the unfiltered case with differences in the -0.01% or -0.03% range, except for the 12Å category, where a small improvement of 0.08% is observed. Thus the small but significant improvements observed with Comb can be imputed to the combination of different information associated with the 6Å, 8Å, 10Å, and 12Å categories.

DISCUSSION

We have used recursive neural network techniques to develop a new contact prediction system that improves previous systems and achieves correct prediction performance in the 70-74% range. We have also collected contact statistics and studied the effect of contact radius on prediction. Our system has been implemented as a web server available through http://promoter.ics.uci.edu/BRNN-PRED/.

In the present system, the architecture must learn to discriminate whether in a given context the number of contacts of a given amino acid is above or below its average across a large set of proteins, *without* being given this average explicitly. Better performance, of faster learning, perhaps could be achieved if this average, which can easily be computed offline, were given in explicit form to the networks, for instance as a direct input to the output layer.

The present work is currently being extended in several directions including:

- The prediction of the exact number of contacts rather than its relative magnitude with respect to the average.
- The construction of a similar BRNN predictor for accessibility.
- The analysis of correlations between number of contacts, accessibility, secondary structure, and residue category on a large data set.
- The integration of contact and accessibility prediction with our existing software for secondary structure prediction, and their further integration for the prediction of contact maps, and ultimately three dimensional structure.

ACKNOWLEDGEMENTS

The work of PB and GP is supported by a Laurel Wilkening Faculty Innovation award and a Sun Microsystems award to PB at UCI. The work of PF and RC is supported by a grant from the Minstero della Università e della Ricerca Scientifica e Tecnologica (MURST) delivered to the project "Structural Functional and Applicative Prospects of Proteins from Termophiles" and by a grant

for a target project in Biotechnology of the Italian Centro Nazionale dell Ricerche (CNR).

REFERENCES

Abagyan, R. and S. Batalov (1997). Do aligned sequences share the same fold? *J. Mol. Biol. 273*, 355–368.

Altschul, S., W. Gish, W. Miller, E. Myers, and D. Lipman (1990). Basic local alignment search tool. *J. Mol. Biol. 215*, 403–410.

Aszodi, A., M. J. Gradwell, and W. R. Taylor (1995). Global fold determination from a small number of distance restraints. *J. Mol. Biol. 251*, 308–326.

Baldi, P. and S. Brunak (2001). *Bioinformatics: the machine learning approach*. Cambridge, MA: MIT Press. Second edition.

Baldi, P., S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics 16*, 412–424.

Baldi, P., S. Brunak, P. Frasconi, G. Pollastri, and G. Soda (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics 15*, 937–946.

Baldi, P., S. Brunak, P. Frasconi, G. Pollastri, and G. Soda (2000). Bidirectional dynamics for protein secondary structure prediction. In R. Sun and C. L. Giles (Eds.), *Sequence Learning: Paradigms, Algorithms, and Applications*, pp. 99–120. New York: Springer Verlag.

Baldi, P., G. Pollastri, C. A. F. Andersen, and S. Brunak (2000). Matching protein β-sheet partners by feedforward and recurrent neural networks. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pp. 25–36. Menlo Park, CA: AAAI Press.

Dill, K. (1999). Polymer principles and protein folding. *Protein Science 8*, 1166–1180.

Fariselli, P. and R. Casadio (1999). Neural network based predictor of residue contacts in proteins. *Protein Engineering 12*, 15–21.

Fariselli, P. and R. Casadio (2000). Prediction of the number of residue contacts in proteins. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pp. 146–151. Menlo Park, CA: AAAI Press.

Flokner, H., M. Braxenthaler, P. Lackner, M. Jaitz, M. Ortner, and M. J. Sippl (1995). Progress in fold recognition. *Proteins 3*, 376–386.

Gorodkin, J., O. Lund, C. A. Andersen, and S. Brunak (1999). Using sequence motifs for enhanced neural network prediction of protein distance constraints. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB99), La Jolla, CA*, pp. 95–105. Menlo Park, CA: AAAI Press.

Hobohm, U., M. Scharf, R. Schneider, and C. Sander (1992). Selection of representative data sets. *Prot. Sci. 1*, 409–417.

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

Lund, O., K. Frimand, J. Gorodkin, H. Bohr, J. Bohr, J. Hansen, and S. Brunak (1997). Protein distance constraints predicted by neural networks and probability density functions. *Prot. Eng. 10*, 1241–1248.

Olmea, O., B. Rost, and A. Valencia (1999). Effective use of sequence correlation and conservation in fold recognition. *J. Mol. Biol. 293*, 1221–1239.

Olmea, O. and A. Valencia (1997). Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold. Des. 2*, S25–32.

Ortiz, A. R., A. Kolinski, P. Rotkiewicz, B. Ilkowski, and J. Skolnick (1999). Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins Suppl. 3*, 177–185.

Richardson, C. J. and D. J. Barlow (1999). The bottom line for prediction of residue solvent accessibility. *Protein Engineering 12*, 1051–1054.

Sali, A., E. Shakhnovich, and M. Karplus (1994). How does a protein fold? *Nature 369*, 248–251.

Sanchez, R. and A. Sali (1998). Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. *PNAS 954*, 13597–13602.

Shindyalov, I. N., N. A. Kolchanov, and C. Sander (1994). Can three-dimensional contacts of proteins be predicted by analysis of correlated mutations? *Protein Engineering 7*, 349–358.