Protein Engineering vol.8 no.3 pp.225–236, 1995

# Neural network system for the evaluation of side-chain packing in protein structures

**Mariusz Milik[1,2], Andrzej Kolinski[3,4] and Jeffrey Skolnick[3]**

[1]The R.W.Johnson Pharmaceutical Research Institute, 3535 General Atomics Court, San Diego, CA 92121, [3]Department of Molecular Biology, The Scripps Research Institute, 10666 North Torrey Pines Road, La Jolla, CA 92037 and [4]Department of Chemistry, University of Warsaw, Pasteura 1, 02-093 Warsaw, Poland

[2]To whom correspondence should be addressed

**An artificial neural network system is used for pattern recognition in protein side-chain–side-chain contact maps. A back-propagation network was trained on a set of patterns which are popular in side-chain contact maps of protein structures. Several neural network architectures and different training parameters were tested to decide on the best combination for the neural network. The resulting network can distinguish between original (from protein structures) and randomized patterns with an accuracy of 84.5% and a Matthews' coefficient of 0.72 for the testing set. Applications of this system for protein structure evaluation and refinement are also proposed. Examples include structures obtained after the application of molecular dynamics to crystal structures, structures obtained from X-ray crystallography at various stages of refinement, structures obtained from a *de novo* folding algorithm and deliberately misfolded structures.**

*Key words:* protein side-chain contact maps/protein side-chain packing/protein structure quality/protein structure refinement/simplified protein models

## Introduction

The development of criteria for the evaluation of the quality of side-chain packing in a proposed protein structure constitutes an important aspect of the protein modeling problem. This is particularly important for reduced models of protein structure where less well-defined, statistically based interaction parameters must be used (Jernigan, 1992). For example, lattice protein folding programs are able to generate many (often non-native) compact structures with valid side-chain packing densities and low energy (Kolinski *et al.*, 1993; Skolnick *et al.*, 1993). Similarly, there are many methods for generating full atom models of proteins by homology modeling (Hilbert *et al.*, 1993), segment matching (Levitt, 1992) and other techniques (see Maggiora *et al.*, 1991, for a review). A major problem is how to distinguish between 'better' (protein-like) and 'worse' (randomly packed) structures. Unfortunately, in both reduced and full atom models, existing potentials make it difficult to correlate the quality of a structure (measured for example by r.m.s. deviation from the native structure) with its energy (Holm and Sander, 1992; Maiorov and Crippen, 1992; Yun-yu *et al.*, 1993). The importance of the patterns of residue–residue contacts for helical structures was shown by Chothia *et al.* (1981). They analyzed 50 helix–helix packings and found that in these structures contact patterns are formed by rows of residues usually separated in sequence by three or four residues. Presnell and Cohen (1989) analyzed four helical bundles from the point of view of a supersecondary structure. Using this analysis, they formulated semi-empirical rules for packing in this type of protein structure. Furthermore, Chiche *et al.* (1990) focused on solvation free energy as a parameter for the evaluation of protein structure. The solvation free energy, calculated according to Eisenberg and McLachlan (1986), is linearly related to the size of a protein. If the calculated solvation free energy for the given protein structure is smaller than predicted, this is a good sign that the structure is misfolded. Unfortunately, this method of protein structure determination may have problems with evaluating well-packed random structures (Chiche *et al.*, 1990). Finally, the importance of internal packing interactions for protein structure stability was examined in the work of Lim and Sauer (1991). Their analysis of the stability of structures with mutations in the hydrophobic core showed that the problem of protein structure determination may be approached from the point of view of compatibility of patterns of interactions with the local structure and protein sequence.

There are many studies which focus on the characteristic regularities in structures (Vriend and Sander, 1991; Godzik *et al.*, 1993; Holm and Sander, 1993). Some fragments of protein structures appear to be more 'protein-like' than others. It is probable that our mind has the ability to generalize the information about all the protein structures we have seen and to use it to create an idea of what a globular protein should look like. By examining a given protein model, we can say how close a particular fragment is to an image of an ideal protein structure. Here we propose an artificial neural network (ANN) to make this intuitive vision of a 'typical protein' more objective. We have used this network to evaluate the quality of protein structures. The generalization of individual data and pattern classification are areas where ANN systems have proved to be very efficient (Hinton and Anderson, 1981; Rumelhart *et al.*, 1986; Eberhart and Dobbins, 1990).

A good review of the application of neural networks to patterns in chemical problems can be found in Burns and Whitesides (1993). Turning to the subject of protein structure prediction, ANNs were used successfully by Holley and Karplus (1989), Qian and Sejnowski (1989) and Kneller *et al.* (1990) for secondary structure prediction from sequence information. McGregor *et al.* (1989) used an ANN method for the categorization of β-turn types in protein structures. Bohr *et al.* (1988) used an ANN for the prediction of elements of tertiary structures of proteins from sequence information, but in both cases these results confirmed the common belief that local information about the protein sequence is insufficient for the prediction of protein structure.

Our aim was to present information about protein structures to an ANN in a suitable form so that the ANN could learn the difference between protein-like and random-packed structures. This approach is novel in that we focus on the structure of a protein rather than its sequence. We have also used the information about interactions between residues far in sequence. Therefore, our method may be complementary to those methods which use local sequence information.

A simple way of translating information about protein structure into a form suitable for an ANN system is to use a side-chain–side-chain contact map representation (Godzik *et al.*, 1993; Holm and Sander, 1993). In the literature there are many definitions of protein contact maps; here we have used one side-chain–side-chain heavy atom contact representation of protein structures. When the distance between any pair of heavy atoms taken from residues in different side chains is less than some threshold value, the side chains are defined as being in contact. This representation suppresses much of the detailed information about protein structure, but at the same time it concentrates on the packing and ordering of side chains. In our work this value was set to 5 Å (Godzik and Sander, 1989). We have found that the actual value of this cut-off threshold is not very important provided that it is in the range 5.0 ± 0.5 Å.

In earlier work Godzik *et al.* (1993) extracted characteristic patterns from protein contact maps. They showed that some interacting regions in different proteins give very similar side-chain–side-chain contact patterns. Following up on this idea, we are in the process of preparing a full library of interaction patterns (A.Godzik, M.Milik, A.Kolinski and J.Skolnick, manuscript in preparation). In our present work we have used a fragment of this library as the basis for training and testing sets for the pattern recognition system.

For simplicity, we have ignored the identity of the amino acids in contact. Thus, by way of an example, a Leu–Leu contact is treated in the same way as a Lys–Leu contact. Here, we concentrate on the problem of whether a given contact is compatible with its local environment, as defined by the local contact pattern. We do not take into consideration the complicated interactions which are responsible for forming this pattern and protein structure. Thus, this is clearly just the first step in a more complicated procedure that assigns sequence specificity to such patterns.

## Materials and methods

### *Preparation of training and testing sets*

For back-propagation networks, the preparation of input sets is as important for a successful learning procedure as is the architecture of the network. Our goal was to prepare balanced sets of positive and negative patterns. Positive patterns should be representative of structures which are in the Brookhaven Protein Data Bank (PDB; Bernstein *et al.*, 1977). The negative examples should not include just random examples because we want our neural network to be able to distinguish between correct and 'partially correct' structures.

Initially a set of contact maps for 240 globular protein structures in the PDB (Bernstein *et al.*, 1977) was prepared. These structures were chosen from a list of representative protein structures published by Hobohm and Sander (1994). A sliding window of seven × seven residues was then used to scan every map from this set. The size of the window was set by a preliminary analysis of the side-chain contact maps of well-refined protein structures from the PDB. A seven × seven residue window appears to be the smallest size which includes most of the popular patterns that reflect the side-chain packing of various structural elements. When the number of contacts between residues in the window exceeded four, the contents of the window were stored as a pattern. Patterns are represented by strings of binary numbers.

The procedure to go from the protein structure to the set of seven×seven patterns is presented schematically in Figure 1. The distances between all side-chain atoms were calculated for the example protein structure (Figure 1A). If the distance between any atom from side chain '1' and any atom from side chain 'a' is less than some threshold value (in our case 5 Å), residues '1' and 'a' are in contact. Information about contacts

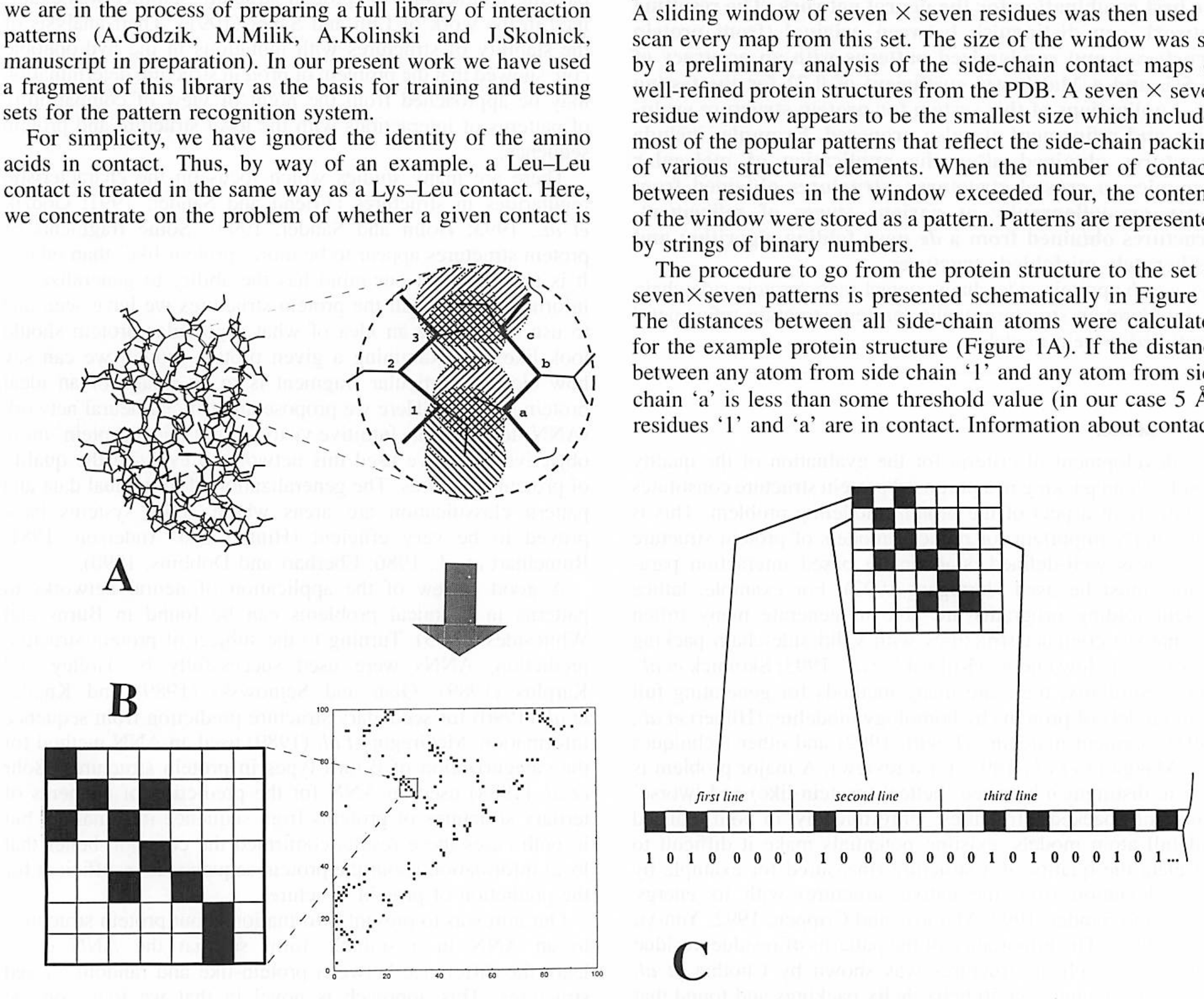


**Fig. 1.** Schematic illustration of the procedure for going from a protein structure to a set of binary patterns. See text for details.

is presented in condensed form in the contact map (Figure 1B). [On the horizontal and vertical axes are the numbers of residues in the protein sequences; a black square denotes contact. By way of an example, the point with coordinates (40, 78) denotes that residue 40 is in contact with residue 78 according to our definition.] These contact maps were then scanned using the seven × seven window and the information about the pattern was stored in the form of a binary vector (Figure 1C).

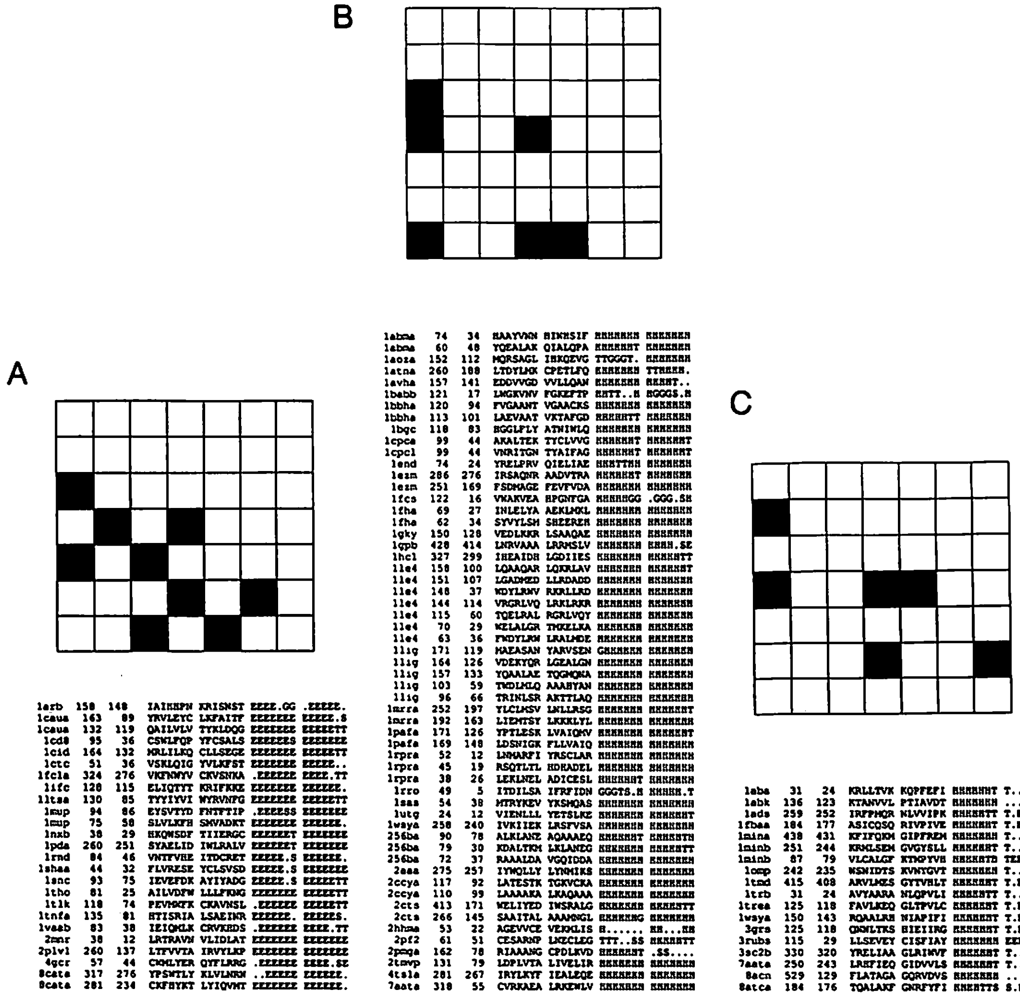
As shown in Figure 1C, the string was obtained by scanning the window starting from the upper left-hand corner; a '1' denotes a contact and a '0' indicates that no contact occurs. In our present work we have focused on long-range contacts, defined when the residues are more than five residues apart down sequence. Thus, each seven × seven contact map is converted into a 49-bit string.

This procedure provided an initial file containing 31 851 'dense' patterns. These patterns were then lexicographically sorted and the frequency of occurrence in the database of every pattern was calculated. To focus our attention on statistically important (non-random) patterns, we employed those patterns which occurred more than twice in the database. There were 1476 such patterns (available by ftp).

As an illustration, Figure 2A–C presents examples of three popular patterns in our database. Additional information about fragments of proteins where these patterns could be found is also provided, and includes the name of the protein, the numbers of the central residue, the sequences of both fragments in contact and the secondary structures of these fragments according to the Kabsch and Sander (1983) definition.

Typically, each pattern is related to the specific type of secondary structure in contact. Figure 2A represents a typical β–β pattern, Figure 2B an α–α pattern and Figure 2C an α–β pattern. A more detailed analysis of the pattern library and the relationship between patterns and secondary structure will be published in subsequent work.

The set of filtered patterns (with at least more than two representatives in our database) was then used to build target and training sets for the neural network program. Since a feed-forward back-propagation architecture of neural network was used, both positive and negative examples are required for training. To prepare examples of negative patterns two methods were used. Random patterns with the same density distribution of contacts as in the set of positive patterns were generated. In the second method of scrambling side-chain protein contact patterns, the algorithm tries to move every point in the pattern to its neighboring position. If this new randomly chosen position is empty, then the move is accepted. If five or more



| | | | | | | |
|---|---|---|---|---|---|---|
| 1arb | 158 | 148 | IAIHHPN | KRISNST | EEEE.GG | .EEEEE. |
| 1caua | 163 | 89 | YRVLEYC | LKFAITF | EEEEEEE | EEEEE.S |
| 1caua | 132 | 119 | QAILVLV | TYKLDQG | EEEEEEE | EEEEETT |
| 1cd8 | 95 | 36 | CSWLFQP | YFCSALS | EEEEEES | EEEEEEE |
| 1cid | 164 | 132 | HRLILKQ | CLLSEGE | EEEEEEE | EEEEETT |
| 1ctc | 51 | 36 | VSKLQIG | YVLKFST | EEEEEEE | EEEEE.. |
| 1fcla | 324 | 276 | VKFNWYV | CKVSNKA | .EEEEEE | EEEE.TT |
| 1ifc | 128 | 115 | ELIQTYT | KRIFKKE | EEEEEEE | EEEEEE. |
| 1ltsa | 130 | 85 | TYYIYVI | WYRVNPG | EEEEEEE | EEEEETT |
| 1mup | 94 | 86 | EYSVTYD | FNTFTIP | .EEEESS | EEEEEEE |
| 1mup | 75 | 58 | SLVLKFH | SMVADKT | EEEEEEE | EEEEEE. |
| 1nxb | 38 | 29 | HKQWSDF | TIIERGC | EEEEEET | EEEEEEE |
| 1pda | 260 | 251 | SYAELID | IWLRALV | EEEEEET | EEEEEEE |
| 1rnd | 84 | 46 | VNTFVHE | ITDCRET | EEEEE.S | EEEEEE. |
| 1shaa | 44 | 32 | FLVRESE | YCLSVSD | EEEEE.S | EEEEEEE |
| 1snc | 93 | 75 | IEVEFDK | AYIYADG | EEEEE.S | EEEEETT |
| 1tho | 81 | 25 | AILVDFW | LLLFKNG | EEEEEEE | EEEEETT |
| 1tlk | 118 | 74 | PEVHGKFK | CKAVNSL | .EEEEEE | EEEEETT |
| 1tnfa | 135 | 81 | HTISRIA | LSAEINR | EEEEEE. | EEEEES. |
| 1vaab | 83 | 38 | IEIQHLK | CRVKEDS | .EEEEEE | EEEE.TT |
| 2mnr | 38 | 12 | LRTRAVN | VLIDLAT | EEEEEEE | EEEEEEE |
| 2plv1 | 260 | 137 | LTFVVTA | IRVYLKP | EEEEEEE | EEEEEEE |
| 4gcr | 57 | 44 | CKMLYER | QYFLRRG | .EEEEEE | EEEE.SE |
| 8cata | 317 | 276 | YPSWTLY | KLVLNRN | ..EEEEE | EEEEEE. |
| 8cata | 281 | 234 | CKFHYKT | LYIQVMT | EEEEEEE | EEEEEE. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1abma | 74 | 34 | HAAYVKN | HIKHSIF | HHHHHHH | HHHHHHH |
| 1abma | 60 | 48 | YQEALAK | QIALQPA | HHHHHHT | HHHHHHH |
| 1aoza | 152 | 112 | MQRSAGL | IHKQEVG | TTGGGT. | HHHHHHH |
| 1atna | 260 | 188 | LTDYLMK | CPETLFQ | HHHHHHH | TTHHHH. |
| 1avha | 157 | 141 | EDDVVGD | VVLLQAN | HHHHHHH | HHHHT.. |
| 1babb | 121 | 17 | LWGKVNV | FGKEFTP | HHTT..H | HGGGS.H |
| 1bbha | 120 | 94 | FVGAANT | VGAACKS | HHHHHHH | HHHHHHH |
| 1bbha | 113 | 101 | LAEVAAT | VKTAFGD | HHHHHTT | HHHHHHH |
| 1bgc | 118 | 83 | HGGLPLY | ATHIWLQ | HHHHHHH | HHHHHHH |
| 1cpca | 99 | 44 | AKALTEK | TYCLVVG | HHHHHHT | HHHHHHT |
| 1cpcl | 99 | 44 | VNRITGN | TYAIFAG | HHHHHHT | HHHHHHT |
| 1end | 74 | 24 | YRELPRV | QIELIAE | HHHTTHH | HHHHHHH |
| 1ezm | 286 | 276 | IRSAQNR | AADVTRA | HHHHHHT | HHHHHHH |
| 1ezm | 251 | 169 | FSDMAGE | PEVFVDA | HHHHHHH | HHHHHHH |
| 1fcs | 122 | 16 | VKAKVEA | HPGNFGA | HHHHHGG | .GGG.SH |
| 1fha | 69 | 27 | INLELYA | AEKLMKL | HHHHHHH | HHHHHHH |
| 1fha | 62 | 34 | SYVYLSM | SHEEREH | HHHHHHH | HHHHHHH |
| 1gky | 150 | 128 | VEDLKKR | LSAAQAE | HHHHHHH | HHHHHHH |
| 1gpb | 428 | 414 | LNRVAAA | LRRMSLV | HHHHHHH | HHHH.SE |
| 1hcl | 327 | 299 | IHEAIDH | LGDIIES | HHHHHHH | HHHHHTT |
| 1le4 | 158 | 100 | LQAAQAR | LQKRLAV | HHHHHHH | HHHHHHT |
| 1le4 | 151 | 107 | LGADMED | LLRDADD | HHHHHHH | HHHHHHH |
| 1le4 | 148 | 37 | WDYLRNV | RKRLLRD | HHHHHHH | HHHHHHH |
| 1le4 | 144 | 114 | VRGRLVQ | LRKLRKR | HHHHHHH | HHHHHHH |
| 1le4 | 115 | 60 | TQELRAL | RGRLVQY | HHHHHHH | HHHHHHH |
| 1le4 | 70 | 29 | KELALGR | TMKELKA | HHHHHHH | HHHHHHH |
| 1le4 | 63 | 36 | FWDYLRW | LRALMDE | HHHHHHH | HHHHHHH |
| 1lig | 171 | 119 | HAEASAN | YARVSEN | GHHHHHH | HHHHHHH |
| 1lig | 164 | 126 | VDEKYQR | LGEALGN | HHHHHHH | HHHHHHH |
| 1lig | 157 | 133 | YQAALAE | TQGMQNA | HHHHHHH | HHHHHHH |
| 1lig | 103 | 59 | TKDLMLQ | AAAHYAN | HHHHHHH | HHHHHHH |
| 1lig | 96 | 66 | TRINLSR | AKTTLAQ | HHHHHHH | HHHHHHH |
| 1mrra | 252 | 197 | YLCLMSV | LNLLRSG | HHHHHHH | HHHHHHT |
| 1mrra | 192 | 163 | LIEMTSY | LKKKLYL | HHHHHHH | HHHHHHH |
| 1pafa | 171 | 126 | YPTLESK | LVAIQMV | HHHHHHH | HHHHHHT |
| 1pafa | 169 | 148 | LDSNIGK | FLLVAIQ | HHHHHHH | HHHHHHH |
| 1rpra | 52 | 12 | LNHARFI | YRSCLAR | HHHHHHH | HHHHHHH |
| 1rpra | 45 | 19 | RSQTLTL | HDHADEL | HHHHHHH | HHHHHHH |
| 1rpra | 38 | 26 | LEKLNEL | ADICESL | HHHHHHT | HHHHHHH |
| 1rro | 49 | 5 | ITDILSA | IFRFIDN | GGGTS.H | HHHHH.T |
| 1saa | 54 | 38 | MTRYKEV | YKSHQAS | HHHHHHH | HHHHHHH |
| 1utg | 24 | 12 | VIENLLL | YETSLKE | HHHHHHH | HHHHHHT |
| 1waya | 258 | 240 | IVKIIEK | LRSFVSA | HHHHHHH | HHHHHHH |
| 256ba | 90 | 78 | ALKLANE | AQAAAEQ | HHHHHHT | HHHHHTH |
| 256ba | 79 | 30 | KDALTKM | LKLANEG | HHHHHHH | HHHHHTT |
| 256ba | 72 | 37 | RAAALDA | VGQIDDA | HHHHHHH | HHHHHHH |
| 2aaa | 275 | 257 | IYWQLLY | LYNMIKS | HHHHHHH | HHHHHHH |
| 2ccya | 117 | 92 | LATESTK | TGKVCKA | HHHHHHH | HHHHHHH |
| 2ccya | 110 | 99 | LAAAAKA | LKAQAAA | HHHHHHH | HHHHHHH |
| 2cts | 413 | 171 | WELIYED | IWSRALG | HHHHHHH | HHHHHTT |
| 2cts | 266 | 145 | SAAITAL | AAAHNGL | HHHHHHG | HHHHHHH |
| 2hhma | 53 | 22 | AGEVVCE | VEKKLIS | H...... | HH...HH |
| 2pf2 | 61 | 51 | CESARNP | LKECLEG | TTT..SS | HHHHHTT |
| 2pmga | 162 | 78 | RIAAANG | CPDLKVD | HHHHHHT | .SS.... |
| 2tmvp | 131 | 79 | LDPLVTA | LIVELIR | HHHHHHH | HHHHHHH |
| 4tsla | 281 | 267 | IRYLKYF | IEALEQE | HHHHHHH | HHHHHHH |
| 7aata | 318 | 55 | CVRKAEA | LRKEWLV | HHHHHHH | HHHHHHH |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1aba | 31 | 24 | KRLLTVK | KQPFEFI | HHHHHHT | T..EEEE |
| 1abk | 136 | 123 | KTANVVL | PTIAVDT | HHHHHHH | ......H |
| 1ads | 259 | 252 | IRFPHQR | NLVVIPK | HHHHHTT | T.EE..B |
| 1fbaa | 184 | 177 | ASICQSQ | RIVPIVE | HHHHHHT | T.EEEEE |
| 1mina | 438 | 431 | KFIFQKM | GIPFREM | HHHHHHH | T..EEET |
| 1minb | 251 | 244 | KRHLSEM | GVGYSLL | HHHHHHT | T..EEES |
| 1minb | 87 | 79 | VLCALGF | KTKPYVH | HHHHHTB | TEEEEEE |
| 1omp | 242 | 235 | WSNIDTS | KVNYGVT | HHHHHHH | T..EEEE |
| 1tmd | 415 | 408 | ARVLMES | GYTVHLT | HHHHHHT | T.EEEEE |
| 1trb | 31 | 24 | AVYAARA | NLQPVLI | HHHHHTT | T...EEE |
| 1trea | 125 | 118 | FAVLKEQ | GLTPVLC | HHHHHHT | T.EEEEE |
| 1wsya | 150 | 143 | RQAALRH | MIAPIFI | HHHHHHT | T.EEE.. |
| 3gra | 125 | 118 | QNNLTKS | BIEIIRG | HHHHHHT | T.EEEES |
| 3rubs | 115 | 29 | LLSEVEY | CISFIAY | HHHHHHH | EEEEEEE |
| 3sc2b | 330 | 320 | YRELIAA | GLRIWVF | HHHHHHT | T...EEE |
| 7aata | 250 | 243 | LRHFIEQ | GIDVVLS | HHHHHHT | T...EEE |
| 8acn | 529 | 129 | FLATAGA | GQRVDVS | HHHHHHH | ....... |
| 8atca | 184 | 176 | TQALAKF | GNRFYFI | HHHHTTS | S.EEEEE |

Fig. 2. Examples of popular contact patterns in our subset of PDB protein structures, with information where this pattern could be found in PDB protein structures. The information contains the PDB name of the file, the numbers of residues in contact and the fragments of sequences in the seven × seven residue window. (A) β–β pattern; (B) α–α pattern; (C) α–β pattern.

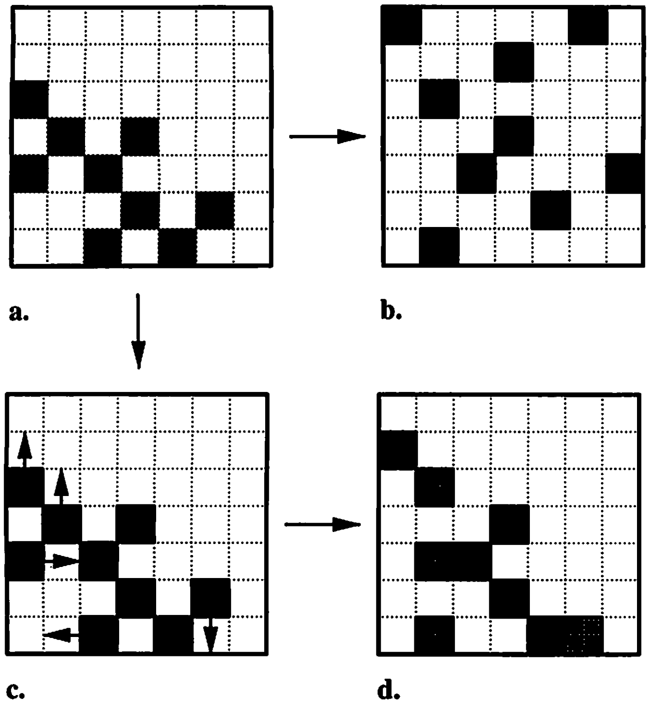


**Fig. 3.** Example of preparation of negative examples for the learning procedure. (**a**) A starting 'native' pattern. (**b**) An example of a random pattern. (**c**) An example of the 'scrambled' pattern. (**d**) Pattern resulting from (c), containing shifted (grey) and native (black) contacts.

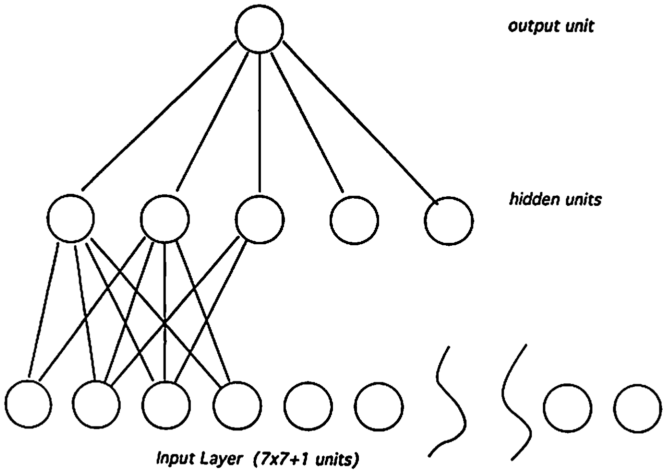


**Fig. 4.** Schematic view of the ANN architecture used in this study. For the sake of clarity, only a subset of input and hidden units and weights is presented. The number of hidden units varies from seven to 30.

of the points in the pattern are moved, then the new pattern is accepted as an example of a disrupted pattern; otherwise, a random pattern was generated in its place.

Figure 3 shows how negative contact patterns were created during the dataset preparation procedure. For the example pattern from the database (Figure 3a), the random negative pattern was obtained by placing contacts in random positions in the seven × seven window. The procedure of preparing a scrambled pattern starts by shifting contacts in the positive pattern in random directions (shown as arrows in Figure 3c). The resulting negative pattern (Figure 3d) contains both shifted (gray) and 'native' (black) contacts. This procedure was used to teach the neural net to discriminate against close to native contact patterns in partially dissolved structures.

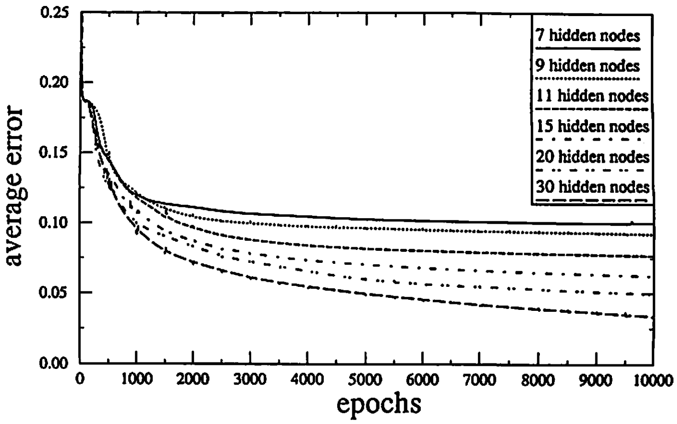


**Fig. 5.** Average error for the ANN as a function of different numbers of hidden nodes, calculated for the training set during the learning procedure.

### *Neural network architecture*

A feed-forward neural network with error back-propagation was used in this work. This system has been described extensively in the literature (Rumelhart *et al.*, 1986; Eberhart and Dobbins, 1990; McClelland and Rumelhart, 1990). Therefore, we present here only those features which are characteristic of our network. Figure 4 shows a schematic view of the architecture. The network contains 50 input nodes, seven to 30 hidden layer nodes and one output node. Our network gives a binary answer (i.e. yes or no) to the question: 'Is the given pattern popular in the structure database or not?' Every hidden layer node and the output node had an additional weight for a bias parameter.

### *Learning procedure*

The neural network was initiated with random weights in the range −0.3 to 0.3. This weight range is widely used for the initiation of back-propagation networks. However, according to the literature, the learning process and final performance of the network do not depend on the initial set of weights (Eberhart and Dobbins, 1990; Masters, 1993). Every epoch of the learning algorithm started with the presentation of an input pattern from the training set. The patterns were presented as vectors of 49 binary values (Figure 1) to 49 input units of our network (Figure 4). Every input unit was connected to every hidden unit with weight $w_{ij}$.

The activations in the hidden layer were calculated in the next step. First, the net input ($net_i$) was calculated for every unit as the sum of activations of the input units multiplied by the corresponding connection weights, plus a bias term associated with the $i$-th unit:

$$net_i = \Sigma_j w_{ij} + bias_i \qquad (1)$$

where the sum is calculated over all input units and $w_{ij}$ denotes the connection weight between the $j$-th and $i$-th units. The net input was then used in the calculation of the activation of the unit according to a sigmoidal activation function:

$$o_i = 1/(1 + e^{-net_i}) \qquad (2)$$

where $o_i$ is the actual activation of the $i$-th unit related to net input from Equation 1.

The activations of the hidden layer nodes were then used as input signals for the output node. The output node activation was calculated using the same procedure as that applied to the hidden layer nodes. The activations of the output nodes for the entire set of training patterns were used for calculating the error function according to the equation:

$$E = \Sigma_p E_p = \Sigma_p (t_p - o_p)^2 \qquad (3)$$

where the index $p$ ranges over the set of input patterns, and $E_p$ denotes the individual error for the $p$-th pattern. The parameter $t_p$ denotes the target value of the output node for the $p$-th pattern. The target values were assigned as 0.99 for positive examples and 0.01 for false examples. The goal of the learning procedure is to minimize the value of the error calculated for the training set. A variant of the gradient descent method was used for this purpose. After the presentation of the full set of input patterns, the error function was computed and each weight was moved down the error gradient towards the minimum. The momentum rule was used to speed up the minimization procedure and to give the system the possibility of escaping from small local minima. According to the momentum rule, information about the change of weights in the $n - 1$-th step is used in the calculation of the next value of the weights. The change of weight $w_{ij}$ was then determined according to the formula:

$$\Delta w_{ij}(n) = -\eta(\partial E/\partial w_{ij}) + \alpha \Delta w_{ij}(n - 1) \qquad (4)$$

(see Rumelhart *et al.*, 1986) where $\eta$ is the learning rate parameter, $\alpha$ is the momentum and $\Delta w_{ij}(n - 1)$ is the change of the weight in the previous step. Using the new hidden/output weights and the back-propagation rule, we can calculate $\Delta w_{ij}$ for the input and hidden layer matrices. This procedure is repeated several hundred or several thousand times until the system reaches a stable point.

*Performance measure*

The first problem which must be solved to find the optimal performance of an ANN system is the question of interpretation of the output from the network. In our case, this output is a real number in the range (0.0, 1.0). This number has to be translated into a binary (yes/no) answer. The simplest way of solving this problem is to use a decision threshold. When the output exceeds the decision threshold value the answer is interpreted as 'yes'; otherwise, it is interpreted as 'no'.

The error back-propagation algorithm is based on the minimization of average sum-squared error, $E$ (Equation 3). The error parameter $E$ is taken as an efficiency meter and is measured and plotted during the training procedure.

The simplest and most commonly used measure of performance is the percent of correct predictions. This parameter depends strongly on the characteristics of the training set. When it is the only method of evaluating performance, it may lead to misleading conclusions (see, for example, Eberhart and Dobbins, 1990; Masters, 1993). For example, one can guess that the answer is always protein-like. This would be correct for exactly the percentage of the protein-like patterns in the training set. However, this has no predictive value and thus other measures of success that address how well the system performs are required.

One popular method of measuring the efficiency of a prediction scheme was proposed by Mathews (1975). This method defines the Mathews' correlation coefficient:

$$C_M = (pn - uo)/\sqrt{[(n + u)(n + o)(p + u)(p + o)]} \qquad (5)$$

where $p$ is number of true positive predictions (a protein-like pattern from the training set was classified by the network as being protein-like), $n$ is the number of true negative predictions

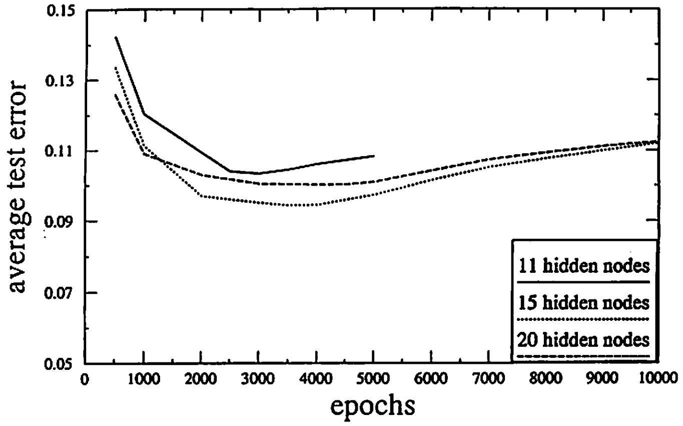


**Fig. 6.** Average error for an ANN with different numbers of hidden nodes, calculated for the testing set during the learning procedure.

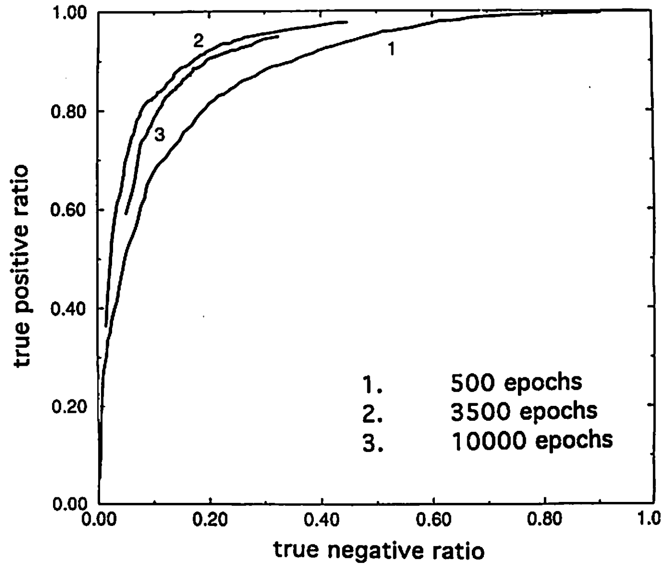


**Fig. 7.** Examples of ROC curves determined for an ANN with 15 hidden nodes, calculated for three stages of the learning procedure.

(a non-protein-like pattern was classified as non-protein-like), $u$ is the number of underpredicted cases (a protein-like pattern was classified as non-protein-like) and $o$ is the number of overpredicted cases (a non-protein-like pattern was classified as being non-protein-like).

A very good tool for measuring the accuracy of an automated diagnostic or prediction system is the receiver operating characteristic (ROC) curve. The ROC curve is widely used for measuring the performance of electronic communications systems (Eberhart and Dobbins, 1990; Masters, 1993). The results obtained with this method do not depend on the probability distribution of the training/test set patterns or decision bias. To define the ROC curves we have to classify the possible answers of the system.

The ROC curve is defined in terms of two ratios of parameters. The first ratio, $p/(p + u)$, is called the true positive ratio; the second ratio, $n/(n + o)$, is called the true negative ratio. The ROC curve is a plot of the true positive ratio as a function of the true negative ratio, calculated for different values of the threshold parameter, $\gamma$. Using the ROC curve one can measure visually how the ANN performance depends on $\gamma$. The ROC curve for a totally random network is a major

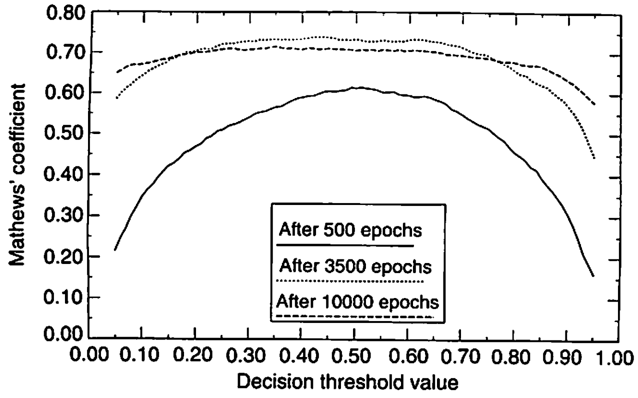


**Fig. 8.** Values of Matthews' coefficient as a function of the decision threshold value calculated for three stages of the learning procedure. The values were obtained using the ANN with 15 hidden nodes for the testing set of patterns.

**Table I.** Neural network prediction efficiencies for cross-validation sets

| Dataset | Matthews' coefficient | Percentage of good answers |
|---|---|---|
| 1 | 0.7376 | 86.80 |
| 2 | 0.7387 | 86.93 |
| 3 | 0.7224 | 86.04 |
| 4 | 0.7108 | 85.49 |
| 5 | 0.7379 | 86.66 |

diagonal; ROC curves for better trained networks always lie above the major diagonal. The quality of performance of the network is demonstrated by the degree to which the ROC curve pushes upwards and to the left, and can be measured by the area under the curve (Eberhart and Dobbins, 1990; Masters, 1993).

## Results

First, we have to decide on the optimal architecture of the ANN. While the numbers of input and output nodes are strictly defined by the problem at hand (49 input nodes, one output node), the number of hidden nodes is arbitrary; there is no universal formula to determine this number. Thus, we proceed by varying the number of hidden nodes. Training runs for five different architectures having seven, 11, 15, 20 and 30 nodes were undertaken.

The average error for the training sets during the training process is shown in Figure 5. As can be observed, the average error calculated for the training set decreases with learning time and with increasing number of hidden layer nodes. However, the average error for the training set cannot be a criterion for choosing the ANN architecture. For more complicated ANNs (with a larger number of nodes in the hidden layer), memorization can occur. Larger ANN systems can literally encode the information contained in the training set instead of developing the ability to generalize it. Therefore, the average error calculated for a testing set gives much better information about the performance of an ANN. The testing set should be prepared using the analogous procedure as the training set, but it should be disjoint with the training set.

Figure 6 shows that the average error for the testing set drops rapidly during the initial stage of the learning process, analogous to the training set case, but after some time the value of this parameter grows, most probably due to memorization of the training set. At around the minimum of the error as a function of the number of epochs (learning cycles), the ANN has extracted most of the important features from the training set but the memorization effect is minimized. Examining the values of the minimal average errors of the testing set as a function of the different numbers of hidden nodes, the best performance is obtained using an ANN with 15 hidden nodes. For a less complicated ANN, it may be too difficult to find some features which distinguish the positive from the negative patterns. When the ANN is too large, memorization becomes too strong relative to the generalization process.

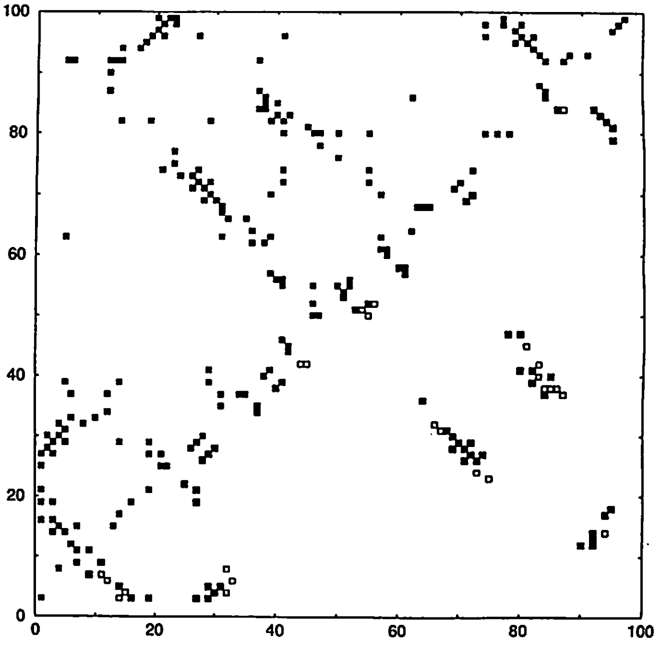


**Fig. 9.** Graph representing the result of the evaluation of the 1pcy.pdb structure using our ANN with 15 hidden nodes. The graph contains information about the contact map of the protein (upper left quadrant) and values of outputs from ANN for these contacts (lower right quadrant). The solid squares represent contacts which were predicted to be 'positive'; the open squares represent the contacts classified as 'negative'. Only the central contacts of every pattern were used in this procedure. This convention is used for every contact map discussed in this paper.

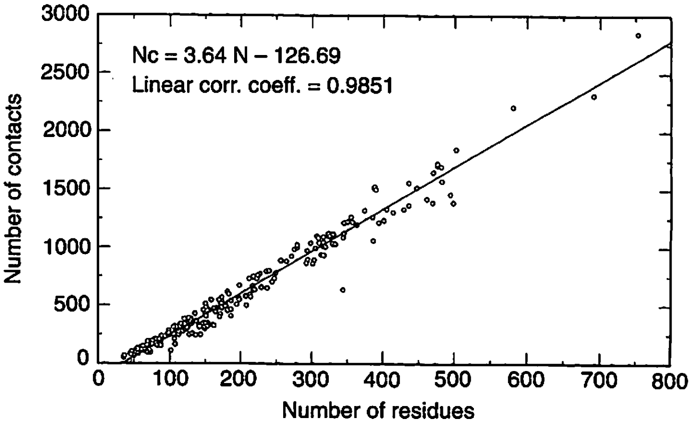


**Fig. 10.** Plot of the number of contacts (defined according to the definition in the text) as a function of the number of residues in the protein structures used in this work.

Analysis of the ROC curves obtained for the ANN with 15 hidden nodes (Figure 7) confirms that the performance of the network for the training set is optimal after 3500 learning

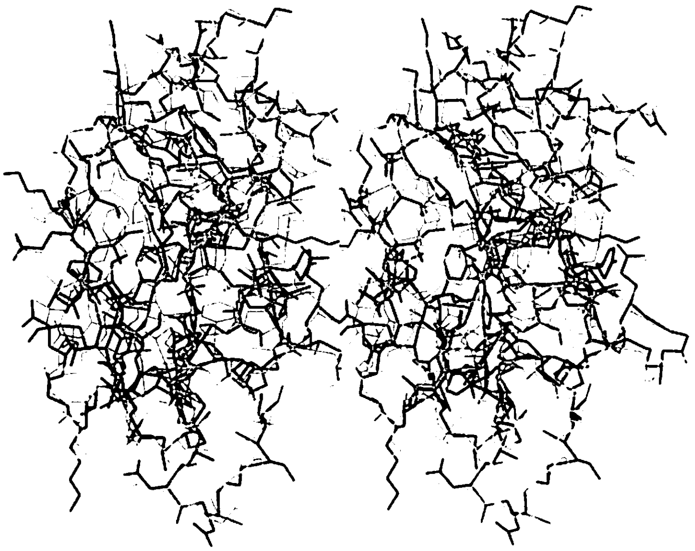


Fig. 11. Stereoview of the distorted structure of 1pcy (black) superimposed on the native structure (grey). The example shows that the overall hydrophobic density and backbone traces are very similar in both structures.

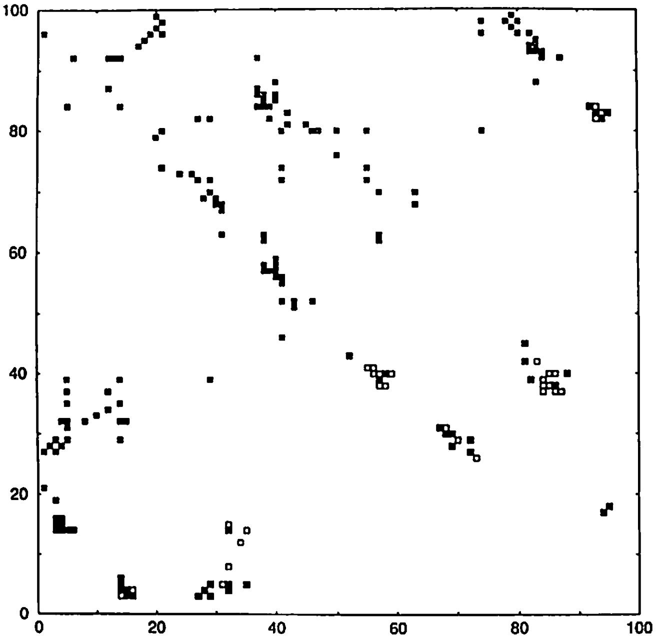


Fig. 12. Evaluation of the disturbed structure of 1pcy.pdb with a r.m.s. deviation on Cα equal to 0.6 Å. See the legend to Figure 9 for additional details.

epochs. The area under this ROC curve, which is a measure of the network performance (Eberhart and Dobbins, 1990), achieves the maximal value.

Figure 8 shows how the value of the Matthews' coefficient changes as a function of the decision threshold value (defined above) in the three stages of the learning procedure. Qualitatively, longer learning times lead to flat curves; thus, for well-trained networks the efficiency depends only slightly on the threshold value, and every threshold value from 0.3 to 0.7 works equally as well. For these values of the threshold parameter the ANN gives ~87% correct predictions for the testing set and a Matthews' coefficient of 0.74.

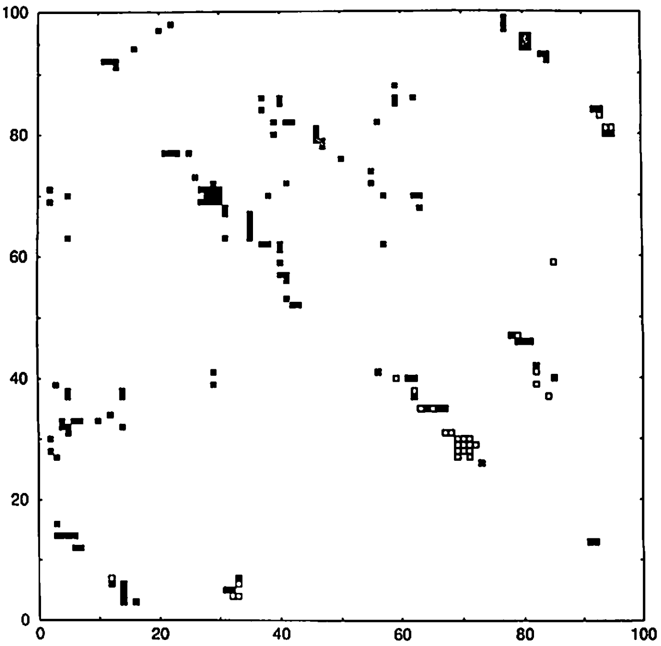


Fig. 13. Evaluation of the perturbed structure of 1pcy.pdb whose Cα r.m.s. deviation from native equals 1.2 Å. See the legend to Figure 9 for details.

In the above procedure, the average error calculated for the testing set of patterns was used as a criterion for searching for the optimal set of parameters for our ANN system. Therefore, there is the possibility that the ANN obtained in this way will be partially biased by the testing set. To eliminate this we used a second testing set. This second set was built in an identical manner to the 'training' and 'testing' sets. We employed a set of 33 structures of globular proteins chosen from the PDB and built it only after completing the training procedures. Our prior decisions about the architecture of the ANN and the parameter set did not depend on the results obtained for this set. Thus, the set was used only to evaluate the predictive ability of the ANN. The results of the second testing were very similar to those for the original testing set. The ANN gives 84.5% correct predictions, with a Matthews' coefficient of 0.717.

The final test of our learning procedure consisted of cross-validation runs to show that our results do not depend on the choice of learning and testing sets. The cross-validation sets contain 10% of patterns chosen from the original testing set and 90% of patterns chosen from the original training set. The first cross-validation set was prepared by taking every 10th pattern (starting from the first one) from the testing set and the rest of the patterns from the training set. The second cross-validation set was prepared by an analogous procedure; this time the testing set was taken from every pattern $j$ such that mod($j$,10) = 2. In this way we prepared five different sets of patterns with different testing patterns.

The learning procedure was repeated for the cross-validation sets and the results are presented in Table I. The results show that for the size of database used in the work presented, the results do not depend on the testing set preparation method.

### *Program for pattern recognition in protein contact maps*

A program for the recognition of popular patterns in protein contact maps was prepared based on the results of the learning procedures described above. The program, written in C and C++, contains two main parts. For each structure a side-

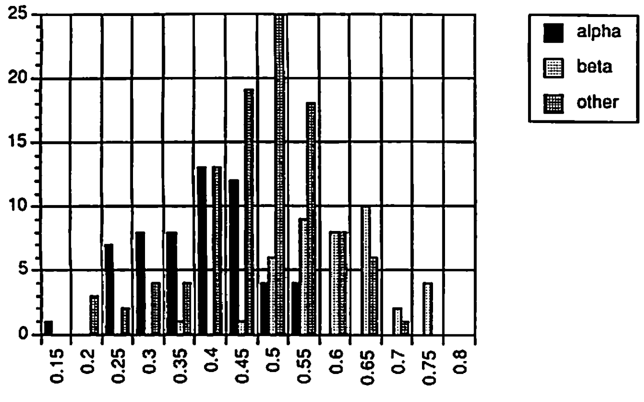


Fig. 14. Histogram of values of the pattern evaluation parameter (defined in the text) for helical proteins (α), β-proteins (β) and mixed proteins (other). For this test we used a subset of 202 protein structures from the PDB.

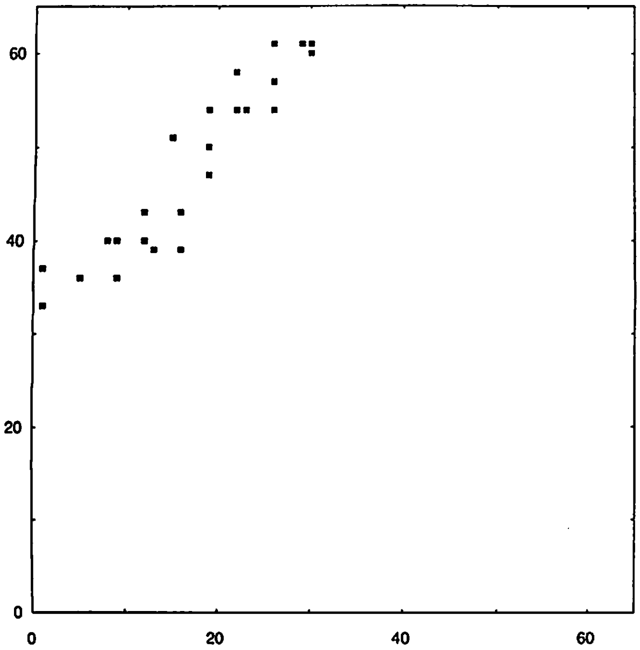


Fig. 15. Evaluation of the structure obtained by adding the backbone and side-chain atoms to the reduced representation structure of the GCN4 leucine zipper dimer obtained from a Monte Carlo procedure. The lower quadrant is empty because no contacts were evaluated as protein-like. The r.m.s. deviation between this structure and the native one is equal to 4.52 Å for all heavy atoms.

chain contact map was prepared. Only information about long-range contacts (more than five residues distant along the sequence) was used. In this way we concentrated on those contacts which form the supersecondary structure of the protein. Then, the contact map was scanned using a seven × seven residue window, and patterns with more than four contacts in this window were presented to the ANN system described above.

When the pattern is one of the popular (protein-like) patterns, the output value of the output node should be high; when it is not popular in the database, then the output should have a low value. The ANN works as an evaluation function and for every pattern gives a value from 0 to 1. This value is assigned to the central contact of the pattern; it represents how well the central contact fits into its environment, or how good local side-chain packing is around the central contact.

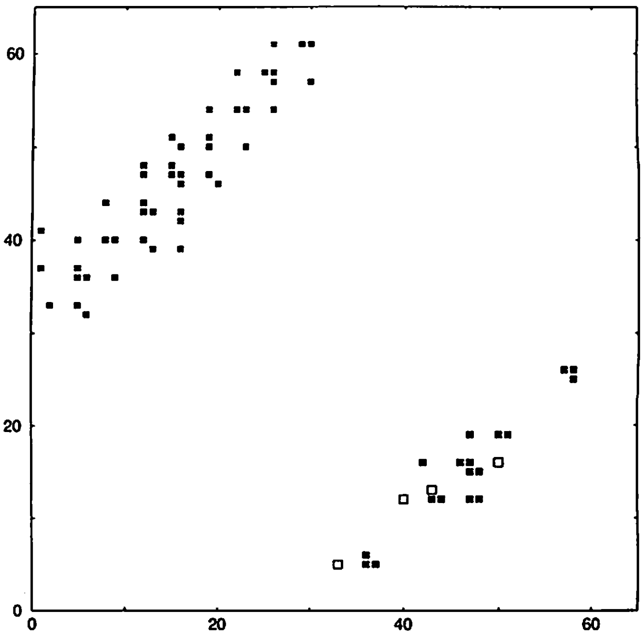


Fig. 16. Evaluation of the structure obtained by molecular dynamics minimization of the structure from Figure 15. The r.m.s. deviation between this structure and the native one equals 4.18 Å for all heavy atoms and 2.61 Å for the Cαs.

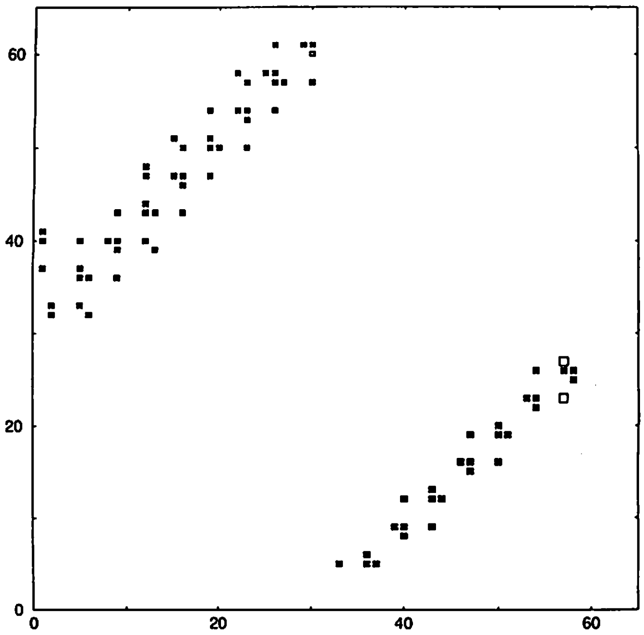


Fig. 17. Evaluation of the structure obtained by molecular dynamics minimization of the structure from Figure 16. The r.m.s. deviation between this structure and the native one equals 2.75 Å for all heavy atoms and 0.68 Å for the Cαs.

Table II. PEPs and r.m.s. deviation values calculated for different models of GCN4 dimer structure

| Model of GCN4 | All atoms r.m.s. | Cα r.m.s. | PEP |
|---|---|---|---|
| Version 0 | 4.52 | 3.05 | 0.000 |
| Version 1 | 4.18 | 2.61 | 0.240 |
| Version 2 | 2.75 | 0.68 | 0.380 |

The models were obtained by Vieth *et al.* (1994) using a combined Motel Carlo and molecular dynamics procedure.

**Table III.** Values of the PEP for the original PDB and deliberately misfolded structures of proteins

| Name of misfolded structure | Based on: | PEP for misfolded structure | PEP for native structure | Difference[a] | Percent[b] |
|---|---|---|---|---|---|
| 1bp2on2paz.brk | 2paz.pdb | 0.425 | 0.688 | 0.263 | 36.23 |
| 1cbhon1ppt.brk | 1ppt.pdb | 0.181 | 0.197 | 0.016 | 8.12 |
| 1fdxon5rxn.brk | 5rxn.pdb | 0.291 | 0.305 | 0.014 | 4.59 |
| 1hipon2b5c.pdb | 3b5c.pdb | 0.183 | 0.259 | 0.076 | 29.34 |
| 1lh1on2i1b.brk | 2i1b.pdb | 0.415 | 0.706 | 0.291 | 41.22 |
| 1p2pon1rn3.brk | 3rn3.pdb | 0.358 | 0.446 | 0.088 | 19.73 |
| 1ppton1cbh.brk | 1cbh.pdb | 0.037 | 0.041 | 0.004 | 9.76 |
| 1reion5pad.brk | 5pad.pdb | 0.254 | 0.400 | 0.146 | 36.50 |
| 1rhdon2cyp.brk | 2cyp.pdb | 0.313 | 0.506 | 0.193 | 38.14 |
| 1rn3on1p2p.brk | 1p2p.pdb | 0.287 | 0.341 | 0.054 | 15.84 |
| 1sn3on2ci2.brk | 2ci2.pdb | 0.186 | 0.276 | 0.090 | 32.61 |
| 1sn3on2cro.brk | 2cro.pdb | 0.077 | 0.256 | 0.179 | 69.92 |
| 2b5con1hip.brk | 1hip.pdb | 0.146 | 0.252 | 0.106 | 42.06 |
| 2cdvon2ssi.brk | 2ssi.pdb | 0.238 | 0.221 | –0.017 | –7.69 |
| 2ci2on1sn3.brk | 1sn3.pdb | 0.202 | 0.258 | 0.056 | 21.70 |
| 2ci2on2cro.brk | 2cro.pdb | 0.160 | 0.256 | 0.096 | 37.50 |
| 2croon1sn3.brk | 1sn3.pdb | 0.212 | 0.258 | 0.046 | 17.83 |
| 2croon2ci2.brk | 2ci2.pdb | 0.278 | 0.276 | –0.002 | –0.72 |
| 2cypon1rhd.brk | 1rhd.pdb | 0.307 | 0.305 | –0.002 | –0.66 |
| 2i1bon1lh1.brk | 1lh1.pdb | 0.229 | 0.358 | 0.129 | 36.03 |
| 2pazon1bp2.brk | 1bp2.pdb | 0.201 | 0.345 | 0.144 | 41.74 |
| 2ssion2cdv.brk | 2cdv.pdb | 0.082 | 0.181 | 0.099 | 54.70 |
| 2tmnon2ts1.brk | 2ts1.pdb | 0.165 | 0.344 | 0.179 | 52.03 |
| 2ts1on2tmn.brk | 2tmn.pdb | 0.364 | 0.514 | 0.150 | 29.18 |
| 5rxnon1fdx.brk | 1fdx.pdb | 0.249 | 0.262 | 0.013 | 4.96 |
| 5padon1rei.brk | 1rei.pdb | 0.396 | 0.602 | 0.206 | 34.22 |

[a] The difference between the PEP value for the native and misfolded structures. Difference = PEP(native) – PEP(misfolded).

[b] The relative values of change of the PEP parameter between the native and misfolded structures. Percent = difference/PEP(native).

### *Application to native and near native structures*

Figure 9 shows the result of using our program to evaluate the native structure of plastocyanin (1pcy.pdb). This structure was not included in the training set. The results of evaluating protein contact maps will be presented here as asymmetric contact maps. The input contact map is presented above the diagonal, and the results of the evaluation lie below the diagonal. Contacts of patterns predicted to be positive (protein-like, i.e. with an ANN output >0.5) are represented as solid black squares; those contacts which are in the patterns predicted to be negative (non-protein-like, i.e. with an ANN output <0.5) are represented by open squares. For a clearer picture, only the central contacts of the patterns are represented. Most of the patterns in the PDB structure of 1pcy are classified as being strongly protein-like. Only a few contacts in the output are strongly negative (white squares), and a few contacts are unclassified because they belong to patterns which are too sparse.

The pattern evaluation program can also provide additional information about the entire protein structure as a single number. This is the summed answer from the ANN for all patterns divided by the number of residues in the protein; we define this quantity as being the pattern evaluation parameter (PEP). We decided to normalize the summed output from the ANN by the number of residues rather than by the number of contacts so as to penalize additionally structures with low packing densities. Typical structures of globular proteins from the PDB have the number of contacts proportional to the number of residues. Figure 10 plots the number of contacts as a function of the number of residues for the set of proteins used in our study. The linear correlation coefficient is equal to 0.98, which is in good agreement with the above statement.

In the case of partially misfolded structures, the number of contacts drops (with no change in the number of residues) and the PEP will be lower than in the case of a regularly packed structure. In the case of the 1pcy structure, the PEP has the value of 0.398. This is a low value for β-protein structures in the PDB (see the analysis below).

In the next step we have prepared distorted versions of the 1pcy structure. The positions of backbone atoms of the original structure were randomly changed by a small vector, and then the side-chain atoms were added to these distorted backbones using the procedure found in the SYBYL modeling package (version 6.0; Tripos Associates Inc., St Louis, MO). Overlaps and distorted bond lengths and angles were removed by the standard minimization procedure in SYBYL. In this way we have obtained two structures close to the starting 1pcy.pdb structure (as measured by r.m.s. distance), but with randomized contact patterns. The first structure we prepared had a 0.6 Å r.m.s. deviation from native for the Cα backbone atoms, and 2.0 Å r.m.s. deviation on all atoms.

Figure 11 shows a stereoview of the distorted 1pcy structure (black) superimposed on the native 1pcy structure (grey). As one can see, the overall size of the hydrophobic core and backbone trace is preserved in the distorted structure, but the side-chain packing looks different. This observation is confirmed by contact maps of the distorted structure, presented in Figure 12. Compared with the native molecule, this contact map is less organized.

Analysis by our program gives the value 0.273 for the pattern evaluation parameter. Many of the contacts are omitted by the program, but two patterns are shown as being very protein-like. A second distorted structure has an r.m.s. deviation on the Cαs equal to 1.2 Å and a full atom r.m.s. deviation of 2.98 Å. The pattern evaluation parameter is significantly smaller (0.223), and whole fragments of protein are classified as being non-protein-like (Figure 13). Both structures discussed above were prepared only to illustrate that our method can

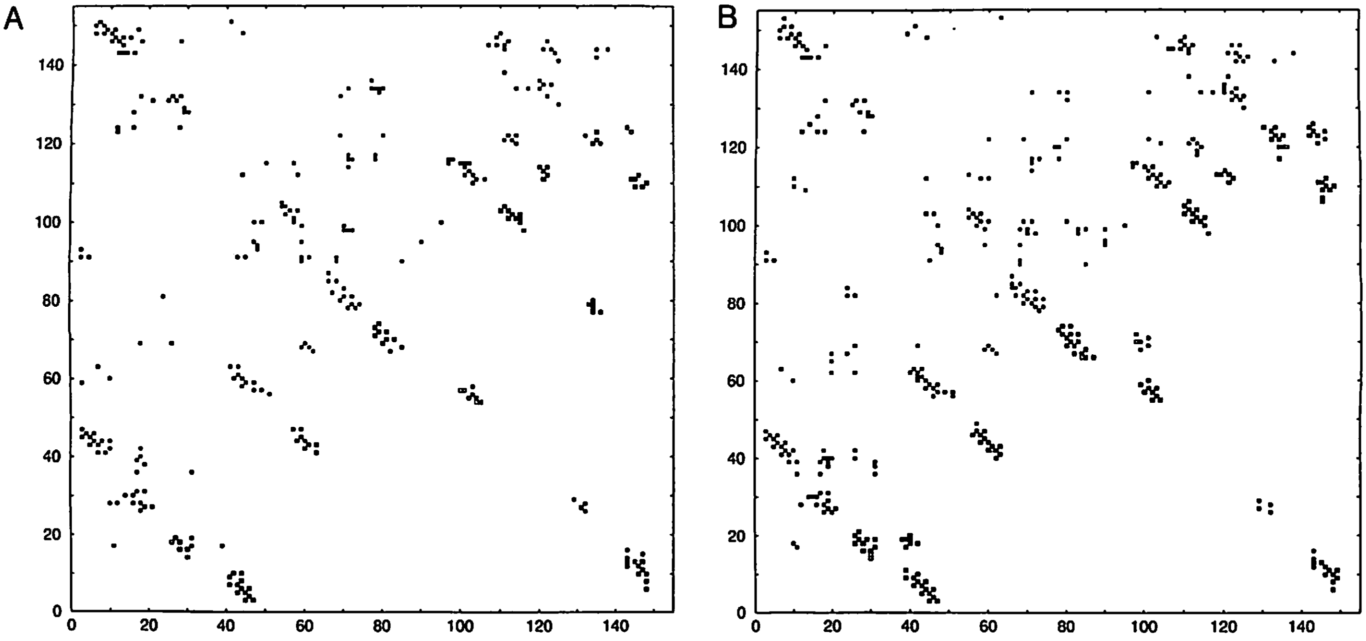


Fig. 18. (A) Evaluation of the quality of a deliberately misfolded structure of 2ilb.pdb [1lhlon2ilb.brk from Holm and Sander (1992)]. The representation is similar to that described in the legend to Figure 8. (B) Evaluation of the quality of the structure of 2ilb.pdb from the PDB. The representation is described in the legend to Figure 9.

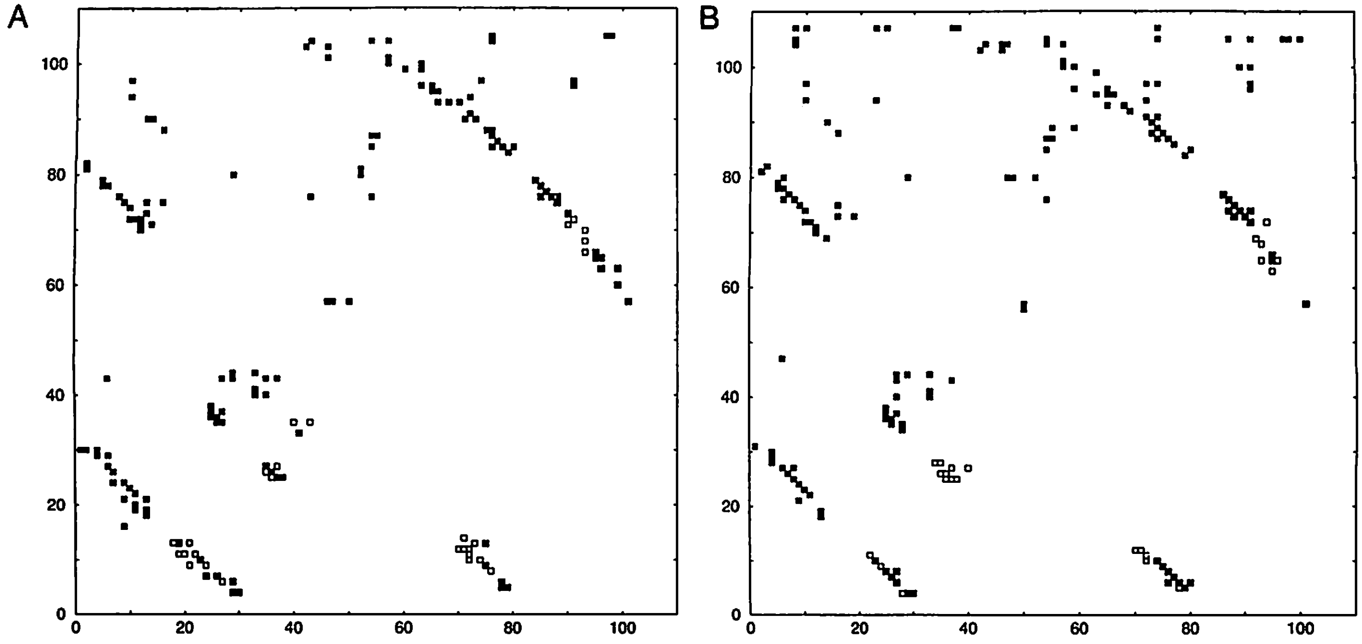


Fig. 19. (A) Evaluation of the deliberately misfolded structure of 2ssi.pdb [2cdvon2ssi.brk from Holm and Sander (1992)]. The representation is described in the legend to Figure 9. (B) Evaluation of the quality of the original structure of 2ssi.pdb.

distinguish between native proteins and structures with randomized side chains.

An interesting question is: How does the pattern evaluation parameter depend on the secondary structure class of the protein? To answer this, we have selected from our subset of PDB protein structures those proteins with prevailing α, β and mixed α/β structures. 'Helical' proteins are those which are at least 40% helical and contain not >10% β structure. 'β'-proteins are those which contain at least 40% β structure and are <10% helical. Mixed α/β-proteins constitute the remainder. For these three sets of proteins we calculated the pattern evaluation parameter. The results are presented as a histogram in Figure 14.

It is evident from this plot that the PEP depends strongly on the type of secondary structure. 'β'-type proteins are, on average, better recognized than 'α' types, with mixed motif proteins intermediate in magnitude. One of the reasons for this effect may be the fact that the contact patterns for β structures are very similar to each other, and it was easier for our ANN to find the characteristic features of these patterns.

### *Application to protein structures at different stages of crystallographic refinement*

Four structures of the variable domain of anti-progesterone antibody DB3 (Arevalo *et al.*, 1993), at different stages of refinement from crystallographic data, were obtained from Dr Ian Wilson's laboratory. These structures were used to test the use of our algorithm for evaluating the quality of protein structures (more precisely, their side-chain packing) at various stages of refinement. The first (starting) structure is a model of the variable domain structure of DB3 built by Drs Arthur Lesk and Cyrus Chothia (MRC Laboratory, Cambridge, UK) according to the methods of Chothia *et al.* (1989). The second structure was obtained by superimposing the domains from the first model onto the structure of McPC603 (Satow *et al.*, 1986). In the third structure, the original coordinates of McPC603 domains were rotated to agree with the experimental data using a correlation coefficient in Patterson space (J.Arevalo, personal communication). The fourth structure is the fragment containing the variable domain from the final version of the DB3 structure (deposited as 1dba.pdb in the PDB), whose refinement is described in Arevalo *et al.* (1993). The PEP values calculated by our algorithm for these structures are 0.562 for the first structure, 0.607 for the second, 0.638 for the third and 0.602 for the fourth. All the PEP values are close to the average values for 'β'-proteins (Figure 14). The value of PEP increases during the first stage of the refinement process, and the difference between the first (model) structure and the third minimized structure is ~12%. This example shows that our method can be useful during the first stage of protein structure refinement in the preparation of the initial model. The difference in PEP between the second (non-minimized) and the last two structures is ~5%. This shows us that the differences occurring during the later stages of refinement of protein structures are at the level of sensitivity of our method. Our method is based on the recognition of characteristic interaction patterns between secondary structure fragments. Protein-like patterns (as assessed by the algorithm) occur before the last minimization stage and change only slightly thereafter. Additionally, our method uses a very simplified representation of protein structure: a simple binary contact/no contact alternative without any side-chain energetic terms. Further development of the model and the inclusion of additional information about local structure should improve the performance of the method for 'almost' well-refined structures.

### *Application to predicted tertiary structures*

In another test we have used structures of the GCN4 leucine zipper dimer, generated by Vieth *et al.* (1994), using a combined Monte Carlo and molecular dynamics procedure. In this procedure, the initial conformations of the proposed structure were generated using a fast Monte Carlo–reduced representation program. These structures were then carefully minimized using the CHARMM molecular dynamics package (Brooks *et al.*, 1983).

Figure 15 shows a contact map for the structure which was obtained by a full atom rebuilding procedure on the basis of a lattice reduced representation structure. The contact map looks random, and no protein-like pattern was found by the program. The value of the pattern analysis parameter is equal to 0.0, and the r.m.s. deviation between this structure and the native conformation 2zta.pdb, measured for the Cαs, is equal to 3.05 Å. This structure was then used as a starting point for the minimization procedure described in greater detail elsewhere (Vieth *et al.*, 1994). Figure 16 presents the map from a structure obtained after the first stage of molecular dynamics refinement. Figure 17 presents the map from the structure at the end of the molecular dynamics minimization procedure.

The r.m.s. deviations from the PDB values and pattern evaluation parameters for these structures are presented in Table II. For these examples, the pattern analysis parameters in Table II correlate with the r.m.s. deviations from native.

### *Application to deliberately misfolded structures*

Holm and Sander (1992) have published a set of deliberately misfolded proteins generated by swapping sequences and structures between proteins with an equal number of residues. The resulting positions of the side chains were then optimized using a fast Monte Carlo algorithm, and the structures were minimized using the GROMOS program (van Gunsteren and Berendsen, 1987).

We used these misfolded structures as an additional test for our ANN method. The misfolded and original PDB structures were evaluated using our algorithm. The results are presented in Table III. In 23 out of 26 cases the average output of our ANN is greater for the native than for the misfolded structure. In 19 cases, this difference is >10%, which means that these structures could be easily distinguished using our method. This is a good test of the sensitivity of our method because no explicit information about sequence is used in the evaluation of structure quality. Using only information about patterns of interaction between supersecondary fragments in these structures, our algorithm can distinguish between two very similar structures with protein-like packing and almost indistinguishable potential energies (Holm and Sander, 1992).

As an illustration, in Figure 18A and B we present an example of contact maps where the difference in the pattern evaluation parameter between the misfolded and native structures is maximal (0.291). The contact maps for both structures look very similar (see upper left quadrants), and there is no obvious difference between the misfolded structure and the structure taken from PDB. It is much easier to classify the structures using the PEP output generated by our algorithm (lower right quadrants). As before, the filled squares represent 'protein-like' patterns and the open squares are the patterns classified by our network as being atypical of proteins. Only the central contacts of the scanned patterns are shown. For the original protein structure, all characteristic β–β interaction patterns are recognized by the ANN as being very popular. There are only a few examples of atypical patterns, and they are located outside the main β structure. In the case of the misfolded structure, the β–β patterns are less characteristic and are often spoiled by the inclusion of non-typical subpatterns. The difference between the evaluation parameters shows distinct variations in the two contact maps. Thus, even for very similar structures (based on an r.m.s. definition) with very similar potential energies, the structures can be distinguished by our method.

This picture is similar for most of the structures from the Holm and Sander (1992) set. However, for five cases (1fdxon5xn, 1pptonlcbh, 2croon2ci2, 2cyponlrhd and 5rxnonlfdx) the variation is too small to allow us to differentiate between the two structures based on the PEP of the contact maps. In the case of 2cdvon2ssi, our evaluation is better for the misfolded than for the parent PDB structure.

As can be seen in Figure 19A and B, the numbers of positive and negative predictions are very similar for both 2cdvon2ssi.brk and 2ssi.pdb structures, but some regions seem to be better ordered in the misfolded than in the native structure. To explain this effect, we examined more closely the PDB file of the original structure of 2ssi.pdb. In their remarks (Bernstein *et al.*, 1977; Hirono *et al.*, 1979), the authors wrote that in the region ALA62 to MET70 the positions of the side-chain atoms are approximate and are given only for reference. When these residues were excluded from consideration, the prediction becomes slightly (a difference of 0.005) better for the native 2ssi.pdb structure. It is possible in this case that the side chains in this loop were better minimized for the misfolded than for the native structure.

## Discussion

The examples presented above show that the ANN system for evaluating patterns in protein contact maps can provide some objective information about the local packing of supersecondary structure fragments in model structures. Using the present algorithm, the process of evaluation of a single protein side-chain contact map is extremely fast and is very easy to vectorize or parallelize. This information may be very useful when one needs to evaluate a large number of possible variants of the protein structure generated, for example, by a Monte Carlo reduced representation program (Skolnick *et al.*, 1993). The algorithm is completely automatic and may be used as an internal procedure in programs used for the prediction of protein structure. The pattern evaluation ANN system has been used, for example, for the restriction of the conformational space to states where the local side-chain packing is similar to a typical pattern in the PDB protein structural database. Additionally, this procedure may detect regions where supersecondary structure packing is very non-protein-like. The present algorithm may be understood as a next step in the implementation of artificial intelligence methods in the protein folding problem. The authors are working on extending this method to make it more general and sensitive by adding information about protein sequences to the pattern information.

## Acknowledgements

We gratefully acknowledge Dr Adam Godzik and Michal Vieth for their stimulating and useful discussions. Thanks are also given to Drs Ian Wilson and J.Arevalo for graciously providing us with structures of the variable domain of the anti-progesterone antibody DB3 at various stages of refinement. This research was supplemented in part by NIH grant number GM-37408.

## References

Arevalo,J.H., Stura,E.A., Taussig,M.J. and Wilson,I.A. (1993) *J. Mol. Biol.*, **231**, 103–118.

Bernstein,F.C., Koetzle,T.F., Williams,G.J.B., Meyer,E.F.,Jr, Brice,M.D., Rodgers,J.R., Kennard,O., Shimanouhi,T. and Tasumi,M. (1977) *J. Mol. Biol.*, **112**, 535–542.

Bohr,H., Bohr,J., Brunak,S., Cotterill,R.M.J., Lautrup,B., Norskov,O.H. and Petersen,S.B. (1988) *FEBS Lett.*, **214**, 223–228.

Brooks,B.R., Bruccoleri,R., Olafson,B., States,D., Swaminathan,S. and Karplus,M. (1983) *J. Comput. Chem.*, **4**, 187–217.

Burns,J.A. and Whitesides,G.M. (1993) *Chem. Rev.*, **93**, 2583.

Chiche,L., Gregoret,L.M., Cohen,F.E. and Kollman,P.A. (1990) *Proc. Natl Acad. Sci. USA*, **87**, 3240–3243.

Chothia,C. *et al.* (1989) *Nature*, **342**, 877–833.

Chothia,C., Levitt,M. and Richardson,D. (1981) *J. Mol. Biol.*, **145**, 215–250.

Eberhart,R.C. and Dobbins,R.W. (1990) *Neural Network PC Tools: A Practical Guide*. Academic Press, San Diego, CA.

Eisenberg,D. and McLachlan,A.D. (1986) *Nature*, **319**, 199.

Godzik,A. and Sander,C. (1989) *Protein Engng*, **2**, 589–596.

Godzik,A., Skolnick,J. and Kolinski,A. (1993) *Protein Engng*, **6**, 801–810.

Hilbert,M., Bohm,G. and Jaenicke,R. (1993) *Proteins*, **17**, 138–151.

Hinton,G.E. and Anderson,J.A. (1981) *Parallel Models of Associative Memory*. Erlbaum, Hillsdale, NJ.

Hirono,S., Nakamura,K.T., Iitaka,Y. and Mitsui,Y. (1979) *J. Mol. Biol.*, **131**, 855.

Hobohm,U. and Sander,C. (1994) *Protein Sci.*, **3**, 522–524.

Holley,L.H. and Karplus,M. (1989) *Proc. Natl Acad. Sci. USA*, **86**, 152.

Holm,L. and Sander,C. (1992) *J. Mol. Biol.*, **225**, 93.

Holm,L. and Sander,C. (1993) *J. Mol. Biol.*, **233**, 123–138.

Jernigan,L.J. (1992) *Curr. Opin. Struct. Biol.*, **2**, 248.

Kabsch,W. and Sander,C. (1983) *Biopolymers*, **22**, 2577–2637.

Kneller,D.G., Cohen,F.E. and Langridge,R. (1990) *J. Mol. Biol.*, **214**, 171–182.

Kolinski,A., Godzik,A. and Skolnick,J. (1993) *J. Chem. Phys.*, **98**, 7420.

Levitt,M. (1992) *J. Mol. Biol.*, **226**, 506–533.

Lim,W.A. and Sauer,R.T. (1991) *J. Mol. Biol.*, **219**, 359.

Maggiora,G.M., Mao,B., Chou,K.C. and Narasimhan,S.L. (1991) *Methods Biochem. Anal.*, **35**, 2–86.

Maiorov,V.N. and Crippen,G.M. (1992) *J. Mol. Biol.*, **227**, 876–888.

Masters,T. (1993) *Practical Neural Network Recipes in C++*. Academic Press, San Diego, CA.

Matthews,B.W. (1975) *Biochim. Biophys. Acta*, **405**, 442–451.

McClelland,J.L. and Rumelhart,D.E. (1990) *Explorations in Parallel Distributed Processing*. MIT Press, Cambridge, MA.

McGregor,M.J., Flores,T.P. and Sternberg,M.J.E. (1989) *Protein Engng*, **2**, 521–526.

Presnell,S.R. and Cohen,F.E. (1989) *Proc. Natl Acad. Sci. USA*, **68**, 6592.

Qian,N. and Sejnowski,T.J. (1989) *J. Biol. Mol.*, **202**, 865.

Rumelhart,D.E., McClelland,J.L. and the PDP Research Group (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA, Vol. 1.

Satow,Y., Cohen,G.H., Padlan,E.A. and Davies,D.R. (1986) *J. Mol. Biol.*, **190**, 593–604.

Skolnick,J., Kolinski,A., Brooks,C.L.,III, Godzik,A. and Rey,A. (1993) *Curr. Biol.*, **3**, 414–423.

van Gunsteren,W.F. and Berendsen,H.J.C. (1987) *GROMOS: Groningen Molecular Simulation Computer Program Package*. University of Groningen, The Netherlands.

Vieth,M., Kolinski,A., Brooks,C.L.,III and Skolnick,J. (1994) *J. Mol. Biol.*, **237**, 361–167.

Vriend,G. and Sander,C. (1991) *Proteins*, **11**, 52–58.

Yun-yu,S., Mark,A.E., Cun-xin,W., Fuhua,H., Berendsen,H.J.C. and van Gunsteren,W.F. (1993) *Protein Engng*, **6**, 289–295.

*Received June 29, 1994; revised August 26, 1994; accepted November 30, 1994*