# On preprosessing of protein sequences for neural network prediction of polyproline type II secondary structure

Markku Siermala, Martti Juhola and Mauno Vihinen

**DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES**
**UNIVERSITY OF TAMPERE**
**REPORT A-2000-13**

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
SERIES OF PUBLICATIONS A
A-2000-13, AUGUST 2000

# On preprosessing of protein sequences for neural network prediction of polyproline type II secondary structure

Markku Siermala, Martti Juhola and
Mauno Vihinen

Department of Computer and Information Sciences
P.O. Box 607
FIN-33014 University of Tampere, Finland

ISBN 951-44-4906-1
ISSN 0783-6910

# *On preprocessing of protein sequences for neural network prediction of polyproline type II secondary structures*

*Markku Siermala* [1], Martti Juhola[1] and Mauno Vihinen[2]*

*[1]Department of Computer and Information Sciences, 33014 University of Tampere, Finland*
*[2] Institute of Medical Technology, 33014 University of Tampere, Finland, and Tampere University Hospital, 20520 Tampere, Finland*

*The corresponding author.

Running title: Preprocessing for PPII predicting neural network

Key words: neural networks, proteins, prediction of polyproline type II secondary structures, polyproline type II structure, PPII

## Abstract

***Motivation:*** *Polyproline type II stretches are rather rare among proteins, and, therefore, it is a very challenging task to try to find them computationally. In the present study our aim was to consider especially the preprocessing phase, which is important for any machine learning method. Preprocessing includes selection of relevant data from Protein Data Bank and investigation of learnability properties. These properties show whether the material is suitable for neural network computing. In addition, algorithms in connection with data selection and other preprocessing steps were considered.*
***Results:*** *We found that feedforward perceptron neural networks were appropriate for the prediction of polyproline type II as well as relatively effective in this task. The problem is very difficult because of high similarity of the two classes present in the classification. Still neural networks were able to recognize and predict about 75 % of secondary structures.*
***Contact:*** *Markku.Siermala@uta.fi*

## Introduction

The present study considered preprocessing tasks necessary to predict secondary structures of polyproline type II (PPII) with neural networks. Proteins are one of the most essential compounds in nature. Polyproline II structures have quite rare occurrences and are therefore difficult to detect. As known, an amino acid sequence forms the primary structure, and a secondary structure as well also other higher order structures are composed of energetically and spatially favourable elements. Obviously, PPII secondary structures had not yet been tried to predict computationally. Neural networks are effective tools, which can recognize complicated relations. They have recently been applied also in bioinformatics, for example Cai and Chan (1995); Fariselli and Casadio (1996); Frishman and Argos (1997); Hanke and Reich (1996); Katz et al. (1992); Petersen et al. (1990); Rost (1997). In fact, preprocessing, especially in terms of neural networks, has to be executed very carefully and effectively so that the best possible results can be gained in the final prediction.

First, we briefly describe how the sequence data was chosen. For the alignment of sequences the algorithm of Needleman and Wunsch (1970) was employed in a conventional way. We analyzed the time complexity of this algorithm. Protein three-dimensional structures were taken from Protein Data Bank (PDB). Two windowing techniques of sequences were treated and tested to localize occurrences for the PPII class. These occurrences were used as elements in learning and test sets of neural networks. The distributions of the PPII class and non-PPII class were scrutinized, and their relatedness was calculated using Hamming distance. Also learnable relationships were studied based on the entropies according to Swingler (1996). Finally, results of our neural network predictions were surveyed and discussed in relation to the preprocessing.

## Selection of polyproline II data for preprocessing

The structural data was acquired from the Protein Data Bank (see ref. PDB), which included data of 8165 macromolecules at the moment of the beginning of the current research, in the autumn 1998. Cases of DNA and RNA were purified and after that there were 6821 macromolecules to be considered. Then cases with the resolution worse than 2.5 Å were deleted. Also theoretical models were discarded. These diminished the data down to 5568 macromolecules. The PDB contains lots of identical proteins. For the entries with identical protein sequences only the one with the highest resolution was chosen. Thereafter, there were 2937 macromolecules.

Then the data was divided into protein families to perform identity comparisons between them. Altogether, 506 protein families were found. Their sizes were very varying from those of a single protein to the largest of 400 proteins. The selection program as well as other software were implemented in C++.

## Identity comparison of proteins

Next, all the sequences within the families were compared to each other. Since a sequence length was about 1000 amino acids, two-dimensional matrices used in the alignment computation required approximately one million cells. Consequently, this computational step was somewhat time-consuming.

Needleman and Wunsch (1970) presented an algorithm for the sequence alignment to find the most similar subsequences. Although the method is familiar, we present it here so that the following complexity analysis is possible to understand. The method is formed on the basis of dynamic programming, which markedly reduces computation compared with a naïve technique of exhausting search. One protein sequence is represented by the rows of a matrix and the other by the columns. Each amino acid of either protein handles one row or column. When the sequences have been aligned, an identity value between two sequences can be computed. The algorithm presented in the following is an improved variation of Needleman and Wunsch, which takes advantage of similarity quantities from the PAM250 table of Mount (1996).

Firstly, matrix $M$ is constructed, which includes $n$ rows and $m$ columns. It is filled with similarity values, which map the relatedness of amino acids. At the same time, path matrix $P$ is generated. Each element of $M$ will contain the cumulative sum of the preceding path. When the lowest row is reached, there is one or more maximal values $t$ at the lowest row or rightmost column. Path matrix $P$ shows how a route has been determined. It also shows jumps, i.e. deletions and insertions in the alignment of sequences. The maximal $t$ is searched for at the third step, and at the fourth one, an identity value is computed. Cost is calculated with punishment $a$+$bl$, in which $a$ is initial cost of a jump, $b$ cost of continuation, and $l$ its length. A jump emerged if in the subtraction of two successive positions ($p_1$, $p_2$) and ($r_1$, $r_2$) there will occur a value greater than 1 in either of the new positions. In the tests run cost $a$ was equal to 4 and $b$ equal to 2.

**Algorithm 1** Alignment of sequences (Needleman and Wunsch).

1. Matrix $M$ is initialized by similarity values from PAM table.
2. Rows $i$=1,..,$n$ are considered in order. At every row, columns $j$=1,…,$m$ are visited and value $M(i,j) = M(i,j) + \max(M(i\text{-}1,..j\text{-}1), M(..i\text{-}1,j\text{-}1))$ is computed, where the two last terms represent the maximum over columns from 1 to $j$-1 of row $i$-1 and the maximum over rows from 1 to $i$-1 of column $j$-1. If a jump exceeds one column or row, punishment value $a$+$lb$ is added to the preceding value. Value ($p_1$, $p_2$) is stored to element $P(i,j)$, where the maximum is located.
3. The maximal $t$ is searched for from the lowest row or rightmost column. Such a path is the best of all.
4. Identity value $I$ is equal to $q/s$, where $q$ is equal to a number of identical amino acid pairs of the path and $s$ is equal to a number of elements of the path.

Next, we present the analysis of time complexity of the algorithm. At the first step there are $m$×$n$ elementary operations for each element of the matrix. Let us assume that $m$ is less than $n$. Since each element is visited once, the time complexity of the first step is $O(n^2)$.

The second step incorporates the largest computational load of the algorithm. At every row $m$-1 operations are accomplished, and there exist $n$–1 rows. In order to compute the updating of $M(i,j)$, 1+2+…+($m$-1) operations, which is equal to ($m$-1)$m$/2, are accomplished, since there are $m$-1 columns at every row. Altogether, there are ($n$-1) ($m$-1)$m$/2 operations. Correspondingly, when the operations are counted column by column, there are ($m$-1)($n$-1)$n$/2 operations. By summing up these two results and by simplifying it a little, we obtain ($n$-1)($m$-1)($m$+$n$)/2. By assuming that $m$ is less than or equal to $n$, it is ultimately gained that

$$\frac{(n-1)(m-1)(m+n)}{2} \leq (n-1)^2 n.$$

From this inequality we obtain an upper bound of the time complexity to be equal to $O(n^3)$. The lower bound is correspondingly $\Omega(m^3)$ according to the assumption given above. There exist faster approximative implementations, which apply the Monte Carlo method, i.e. a solution is found at great probability, but not with entire certainty.

Large families caused problems, since their computing became so time-consuming. Consequently, two largest families were compelled to divide into parts. Otherwise, the largest family of over 400 members would have required a run time of more than 10 days for a PC. The average identity was 30 – 40 %. If an identity value was above 65 %, the resolutions of the sequence pair was compared, and the one with the better resolution was kept, but the other rejected. The high threshold of 65 % was applied to obtain data sufficiently. At the beginning there were 2937 proteins, and after the identity comparison 1847 proteins were accepted.

## Use of structure files

The DSSP method of Kabsch and Sander (1983), which defines secondary structures from atom coordinates, is based on pattern recognition. Since our final aim was to use machine learning methods, neural networks, we sampled 10 % of the ultimate material after the previous preprocessing to a test set and put the rest 90 % to the learning set. Thus, 10 disjoint test sets were obtained.

Two windowing techniques were tested to determine whether a part of a sequence has PPII structure or not. The purpose of the windowing was to choose sequences for neural networks. The location of a PPII structure is considered in terms of the middle position of the window (Figure 1). Ruggiero *et al.* (1993) have suggested a window length of 13 amino acids in secondary structure predictions. Since a number of input nodes in a neural network increases with 20 while increasing the window length with 1, long window lengths were not feasible. The nominal values of amino acids were encoded so that there were 20 input nodes for every element of the window. Exactly one of them is equal to 1, while the others are equal to 0. That one points out which amino acid is present at the current position. After various experiments, we tested exclusively window lengths of 7 and 13, which gave input vectors of either 140 or 260 bits. The first windowing technique (Figure 1) produced more than 8000 valid sequences for neural networks. Correspondingly, the second technique yielded more than 14000 items. When we used feedforward multilayer perceptron networks with backpropagation learning, we needed approximately

$$u = 10(20lp + 2p)$$

cases to the learning set, where *l* is the length of the window, *p* is a number of hidden nodes and there are two output nodes in the three-layer network employed. The number of output nodes is equal to the number of the classes. The principle of winner taking all was employed.

In order to search for PPII structures we applied methods of Adzhubei and Sternberg (1993). The first condition for structures is set to virtual angle α, which is like a sieve with the task to prune such candidates, whose angles ϕ and ψ are outside a predefined area or which are not left-handed. Angle α was computed as follows, where *i* is the index of an amino acid.

$$\mathrm{a} = 180° + ?_{i} + \mathrm{f}_{i+1} + 20°(\sin \mathrm{f}_{i} + \sin ?_{i+1})$$

PPII structures appeared round the point α=-110°, ϕ=-75° and ψ=145° in particular. Geometrically speaking, these angles define a structure, where there are three amino acids per cycle. This forms a structure triangularly repeating in the space. Regularity of a structure was computed with ϕ and ψ angles using the subsequent equation when *n* is the number of amino acids in a structure.

where

$$d_{k-1,k} = \sqrt{(?_{\mathrm{i\text{-}1}} - ?_{\mathrm{i}})^2 + (\mathrm{f}_{i} - \mathrm{f}_{i+1})^2}\,.$$

$$D = \frac{\sum_{k=1}^{n-1} d_{k,k+1}}{n},$$

Regularity is an average distance formed with successive angles ϕ and ψ. Structures were searched for with the following algorithm.

**Algorithm 2** Searching for structures (parameter: size)

```
while there exist protein molecules do
  while there exist amino acids do
    if lower bound < $\alpha$ < upper bound do
      the next amino acid is taken under consideration
      while lower bound < $\alpha$ < upper bound do
        the next amino acid is considered
      end while
      if there are more successive structures than value of size then
        regularity computing of the chain ($D < 50$) with above-mentioned formula
        if structure regular and there are more amino acids in the structure than value of
        size then
          regularity of every part is checked
          if regularity $D$ of every part < 50 then
            amino acids of part are marked PPII active
          end if
        end if
      end if
    end if
    the next amino acid is considered
  end while
  windowing of protein (Algorithm 3)
  the next protein is considered
end while
```

The parameter ‘size’ defines how many suitable (in terms of the angles) amino acids are required for the regularity condition. Sizes of 2 and 3 were tested. The latter was suggested by Azhubei and Sternberg. The time complexity is $O(n^2)$, but in practice its running time is short, because PPII structures are rare comprising only 1.3 % of the studied proteins.

The next algorithm was used to store every case that belonged to class PPII and every *k*th case that belonged to class non-PPII. All non-PPII cases were not included, since there were much more non-PPII cases than PPII cases. The time complexity of this algorithm is $O(n)$.

**Algorithm 3** Windowing of protein

```
window is set at the beginning of protein so that it starts from the position of the first
amino acid of the protein
while end of protein is not encountered do
  if there appears PPII structure in the middle of window then
    sequence encountered within window is stored to PPII file
  else
    if this is the next kth non-PPII then
      sequence within window is stored to non-PPII file
    end if
  end if
  window is moved one amino acid forward
end while
```

## Properties of the material

The beginning of a structure was determined to be at the position of the amino acid $i$ that had angle $\psi$ with satisfied regularity conditions at position $i$-1. The end of the structure was at the position of amino acid $k$ that had angle $\phi$ with satisfied regularity condition at the position $k$+1 (see Figure 2).

As mentioned, window lengths 7 and 13 amino acids were widely tested (see also Siermala *et al.* 2000), and the latter was found to be better choice. For the former parameter ‘size’ there was only 1.24 % of the PPII frequency in proteins on average for the size equal to 3 and about 3 % for the size equal to 2. Out of 1849 proteins 862 contained PPII structures. Since the size of 3 produced better results, only these results are reported in the subsequent text. The first windowing technique (Figure 1) gave 6950 PPII structures and the second technique 11000 PPII structures. The first windowing technique generated better results than the second one. Correspondingly, the window length of 13 evolved better results than 7. Therefore, only the combination of the best choices is considered in the following. Next, we scrutinized frequencies of different amino acids in our data selected. Amino acids G, H, L, N, P, S, V, and Y interacted with PPII, e.g. P occurred very frequently in it, but G rather infrequently.

In Table 1 there are frequency ratios between the PPI class and the non-PPII class when the latter was decreased to the size of the former. Decreasing was accomplished by sampling (as described later) so that the distribution of amino acids within the non-PPI class was kept unchanged. The decreasing was, however, necessary, since the great majority of cases were in the latter class. The column of proline P in Table 1 is essential for PPII, and thus its ratios are high. Especially the 7th row is important, because it is the middle of the window.

The classes of PPII and non-PPII structures were found to be rather similar and therefore we computed Hamming distances between and within the classes to describe the similarity property. Hamming metric was computed with the equation

$$h = \sum_{i=1}^{N} (x_i(1-y_i) + y_i(1-x_i)),$$

for bit vectors, where $x_i$ and $y_i$ are the *i*th variables from the opposite classes. We computed Hamming distances for window lengths 5, 7, and 13, and results of the last case are presented in Table 2. It is difficult to separate between the two classes, because the mode (distance 8) of the PPII class is as far as in the non-PPII class. On the other hand, there are more cases within the PPII class in connection with small (less than 8) distances than within the non-PPII class. Corresponding results were obtained when we applied the PAM250 comparison table and its relatedness values. Relatedness between two sequences was computed by comparing amino acids at the same positions of the two sequences. According to such pairs their relatedness values were summed up as the similarity value of the two sequences. The relatedness values of our data are in Figure 3. Again, it is seen that the situation does not unfortunately differ essentially between the classes and within the PPII class.

Swingler (1996) has presented how learnability of data can be investigated with entropies. The method is based on Shannon's information theory. In our data there were 8500 cases, which were randomly distributed to learning and test sets for neural networks. There were no identical cases within either class, but between the classes there were 20 cases in both classes. Thus, there were 8460 sequences only in either one or the other class. Let *X* be the input cases and *Y* the output cases. Their entropy values are calculated according to

$$H = \sum_{i=1}^{n} p_i \log \frac{1}{p_i},$$

in which $p_i$ is the probability of case *i* and log is the natural logarithm. For the input vectors of a neural network the probability is equal to 1/8500 that a case occurs only in one of the classes, and correspondingly 2/8500 that it occurs in both. Thus, we obtained an entropy value $H(X)$ of *X* which was equal to approximately 9.01. There are two output classes (PPII and non-PPII) and their probabilities were 1/2. Consequently, an entropy value $H(Y)$ was obtained for both classes as log 2, which is approximately 0.69.

Conditional entropy deals with entropy of class $u_i$, when it belongs to input $v_j$. This is defined by

$$H(Y|X) = \sum_{i=1}^{n}\sum_{j=1}^{m} p(u_i, v_j) \log \frac{p(v_j)}{p(u_i, v_j)},$$

where $p(u_i,v_j)$ means the probability that both class $u_i$ and case $v_j$ occur. If an element appears only in one of the classes, $p(u_i,v_j)$ is equal to $p(v_j)$, and the value of log expression is then equal to 0. When an element occured in both classes (20 pieces) and there were always double such elements, we got

$$p(u_i, v_j) = \frac{1}{2}\frac{2}{8500}$$

and

$$p(v_j) = \frac{2}{8500}.$$

All over, there were 40 such cases. Thus, a conditional entropy value of approximately 0.0033 was obtained. Further, we achieved

$$\frac{H(Y|X)}{H(Y)} \approx 0.0048,$$

which is near 0 at the interval of (0,1). This means highly learnable data by Swingler (1996). Mutual information is defined with the subsequent formula.

$$I(X;Y) = H(X) - H(X|Y)$$

This is equivalent to

$$I(X;Y) = I(Y;X) = H(Y) - H(Y|X).$$

Thus we obtained

$$I(Y;X) = \log 2 - 0.003 \approx 0.69.$$

Ultimately, according to Swingler (1996) when a ratio of

$$\frac{I(Y;X)}{H(Y)} \approx 0.95$$

is near the upper bound 1 of the interval (0,1), the material is highly learnable.

Scarcity of PPII cases in the material was a hard problem for neural networks and obviously also for any detection and prediction technique. We chose the most “secure” means, in the sense of neural network computation, to exceed this difficulty by decreasing the non-PPII class remarkably. The non-PPII class was decreased with systematic and random sampling to modify it along with the uniform distribution jointly with the PPII class. Nevertheless, our sampling technique guarantees that the similar distribution inside the non-PPII class was preserved in spite of decreasing.

## Neural network tests and their results

We accomplished wide test series by varying several parameters (Siermala *et al.* 2000), part of which has already been described above. However, in this context it is reasonable and sufficient to present the most successful parameter combinations. On the other hand, differences between results of several combinations were often rather small. Altogether we tested 32 networks of different topologies. We implemented three-layer preceptrons with the backpropagation learning algorithm in Matlab (MathWorks Inc.) programming environment. Two windowing techniques and three sequence lengths were perused with 4, 8, 15, or 25 hidden nodes in networks. Validation sets were applied to prevent overlearning. The data coding was depicted earlier. Crossvalidation was used in the tests.

The results were compared with t test. The window length 13 was significantly better than 5 or 7. Moreover, the first windowing technique was also significantly better than the second one. The numbers 2 and 4 of the hidden nodes were significantly better than the others. In principle more hidden nodes in a perceptron neural network allows more complicated decision surfaces. On the other hand, larger numbers of hidden nodes require larger learning sets and there were only a limited number of cases.

Lastly, the optimal results were obtained with the network with 4 hidden nodes and window length 13. The recognition accuracy of PPII structures is defined as usual

$$r = \frac{tp}{tp + fn} 100\,\%$$

and the prediction accuracy is

$$p = \frac{tp}{tp + fp} 100\,\%,$$

where *tp* is true positive, *fn* false negative and *fp* false positive PPII cases according to decisions made by the neural network. The method recognized 72.6 % and predicted 74.1 % of all PPII cases on average when eight disjoint test sets and eight partially different (dependent on the sampling of the corresponding test set) learning sets were used for crossvalidation. From the non-PPII class the neural network was able to recognize 74.7 % and to predict 73.3 % cases on average. The total accuracy is defined to be

$$t = \frac{tp + tn}{tp + tn + fp + fn} 100\,\%,$$

where *tn* is true negative cases. A total accuracy of 73.7 % was obtained for PPII. PPII structures have not been predicted computationally previously, but a number of neural networks have been trained to predict $\alpha$ helices and $\beta$ strands, e.g. by Ruggiero *et al.* (1993), who obtained accuracy of 72.5 %. Using statistical methods lower values, such as 49 %, 50 – 60 %, and 68.5 %, were obtained in various articles, e.g. by Rost and Sander (1998).

Ultimately, we tested naturally distributed (non-uniformally) test sets from the material, when the non-PPII class was not yet reduced to the similar size as the PPII class. The PPII cases accounted only for 1.3 % of the length of the tested sequences. Nevertheless, the neural network succeeded in recognizing 72.0 % of PPII and 74.5 % of non-PPII on average. However, there still remain a large number of false positive PPII findings, which is a problem, which is not due to the neural network method, but because of the very skewed distribution.

## Conclusions

Feedforward perceptron neural networks were quite efficient to solve the classification problem of PPII, although the similarity between the two classes was high as indicated by Hamming distances. The conditional entropies computed showed that material was well learnable for neural networks. Still the learnability property seems to be at a high level and is not able to take into account all important issues that may be present in the material. The windowing technique was very effective with the window length of 13 amino acids. The best three-layer (one hidden layer) neural network included 4 hidden nodes.

The neural network correctly predicted about three fourths of all cases. The natural, very skewed distribution was still difficult because of a large number of false positive findings. Such a difficulty has not been discussed elsewhere, since other studies have concerned much more common secondary structures like α helices and β strands. We will try to increase the effectiveness of our approach also in this respect. To conclude, the significance of careful preprocessing of the material for neural network approach was well seen in this study. Secondly, neural networks are efficient for solving these complicated prediction problems related to protein structures.

## Acknowledgments

The research of the first author was financially supported by the Academy of Finland and with a research grant of the University of Tampere. The author is grateful to them for their support.

## References

Adzhubei, A. and Sternberg, M. (1993) Left-handed polyproline II helices commonly occur in globular proteins. *J. Mol. Biol.*, **229,** 472-493.

Cai, Y. and Changqing, C. (1995) Artificial neural network method for discriminating coding regions of eukaryotic genes. *CABIOS*, **11,** 497-501.

Fariselli, P. and Casadio, R. (1996) HTTP: a neural network-based method for predicting the topology of helical transmembrane domains in proteins. *CABIOS*, **12,** 41-48.

Frishman, D. and Argos, P. (1997) A neural network for recognizing distantly related protein sequences. In Fiesler, E. and Beale, R. (eds.), *Handbook of Neural Computation*. IOP Publishing and Oxford University Press, pp. G4.4:1-8.

Henk, J. and Reich, J.G. (1996) Kohonen map as a visualization tool for the analysis of protein sequences: multiple alignments, domains and segments of secondary structures. *CABIOS*, **12,** 447-454.

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

Mount, D. (1996) Dayhoff scoring matrices (percent accepted mutation or pam matrix) for sequence comparisons. *HTTP://www.blc.arizona.edu/courses/bioinformatics/dayhoff.html* (August 2, 1999)

Katz, W, Snell J. and Mericel M. (1992) Artificial neural networks. *Meth. Enzymology*, **210,** 610-632.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to search for similarities in amino-acid sequence of two proteins. *J. Mol. Biol*., **48,** 443-453.

PDB: http://www.rcsb.og/pdb/

Petersen S., Bohr, H., Bohr, J., Brunak S., Cotteril, R., Fredholm, H. and Lautrup, B. (1990) Training neural networks to analyse biological sequences. *Tibtech*, **8,** 304-308.

Rost, B. (1997) A neural network for prediction of protein secondary structure. In Fiesler, E. and Beale, R. (eds.), *Handbook of Neural Computation*. IOP Publishing and Oxford University Press, pp. G4.1:1-9.

Rost, B. and Sander, C. (1998) *3rd Generation Prediction of Secondary Structure.* Humana Press.

Ruggiero, C., Sacile, R. and Rauch, G. (1993) Peptides secondary structure prediction with neural networks: a criterion for building appropriate learning sets. *Trans. Biomed. Eng.*, **40,** 1114-1121.

Siermala, M., Juhola, M. and Vihinen, M. Neural network prediction of polyproline type II secondary structures. Accepted to Proc. of Medical Informatics Europe 2000, Hannover, Germany, 27 August – 1 September, 2000.

Swingler, K. (1996) *Applying Neural Networks*. Academic Press, London.

Table 1. Ratios between the frequencies of the PPII class and those of the non-PPII class. Columns are the 20 different amino acids, and rows are positions of the window, where the middle (7th) one is essential. The ratios of proline P are naturally high, i.e. important for PPII.

| | A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0,83 | 0,93 | 0,79 | 0,93 | 0,86 | 1,09 | 0,95 | 1,11 | 1,00 | 0,96 | 0,65 | 1,10 | 1,47 | 1,05 | 1,22 | 1,09 | 0,98 | 0,92 | 0,88 | 1,03 |
| **2** | 0,81 | 1,05 | 0,87 | 0,94 | 0,97 | 1,19 | 0,96 | 0,91 | 1,06 | 0,97 | 0,82 | 1,01 | 1,26 | 1,02 | 1,10 | 1,08 | 0,92 | 0,87 | 0,88 | 1,08 |
| **3** | 0,72 | 1,23 | 0,73 | 0,91 | 0,99 | 1,10 | 0,97 | 1,21 | 1,10 | 0,89 | 0,87 | 0,93 | 1,50 | 1,01 | 1,37 | 1,07 | 1,03 | 0,89 | 0,75 | 1,06 |
| **4** | 0,84 | 1,09 | 0,75 | 0,92 | 1,17 | 0,93 | 0,97 | 1,02 | 1,14 | 0,99 | 0,83 | 0,73 | 1,73 | 0,98 | 1,20 | 0,94 | 0,95 | 1,06 | 0,77 | 1,00 |
| **5** | 0,83 | 1,24 | 0,65 | 0,87 | 1,23 | 0,78 | 0,90 | 1,15 | 1,07 | 1,07 | 0,77 | 0,68 | 2,28 | 0,98 | 1,20 | 0,82 | 1,04 | 1,15 | 0,62 | 0,88 |
| **6** | 0,85 | 1,06 | 0,67 | 0,87 | 0,99 | 0,63 | 0,77 | 1,05 | 0,98 | 1,14 | 0,70 | 0,63 | 2,68 | 0,99 | 1,16 | 1,10 | 1,10 | 1,06 | 0,58 | 0,77 |
| **7** | 0,93 | 0,73 | 0,81 | 1,03 | 0,87 | 0,30 | 0,83 | 0,98 | 1,07 | 1,09 | 0,65 | 0,62 | 4,08 | 1,06 | 1,22 | 0,96 | 0,93 | 1,07 | 0,60 | 0,59 |
| **8** | 0,98 | 0,66 | 0,92 | 1,08 | 0,80 | 0,49 | 0,72 | 0,94 | 0,96 | 1,01 | 0,63 | 0,78 | 3,52 | 0,89 | 1,01 | 1,10 | 0,97 | 0,95 | 0,63 | 0,58 |
| **9** | 0,89 | 0,73 | 1,03 | 1,26 | 0,63 | 0,62 | 0,71 | 0,86 | 0,98 | 0,94 | 0,55 | 0,88 | 3,12 | 1,02 | 1,01 | 1,28 | 1,05 | 0,85 | 0,50 | 0,59 |
| **10** | 0,84 | 0,78 | 1,28 | 1,21 | 0,70 | 0,76 | 0,87 | 0,68 | 0,85 | 0,91 | 0,71 | 0,91 | 2,39 | 1,07 | 1,01 | 1,30 | 1,00 | 0,84 | 0,63 | 0,76 |
| **11** | 0,85 | 1,06 | 1,15 | 1,22 | 0,82 | 0,86 | 0,97 | 0,83 | 0,85 | 0,88 | 0,71 | 1,01 | 2,12 | 1,06 | 1,05 | 1,24 | 0,98 | 0,82 | 0,73 | 0,74 |
| **12** | 0,82 | 0,81 | 1,19 | 1,23 | 0,82 | 0,99 | 0,96 | 0,79 | 0,96 | 0,83 | 0,72 | 0,95 | 1,41 | 1,12 | 1,16 | 1,27 | 1,01 | 0,85 | 0,88 | 0,96 |
| **13** | 0,92 | 1,24 | 0,98 | 1,36 | 0,82 | 0,94 | 1,23 | 0,95 | 0,91 | 0,90 | 0,75 | 0,98 | 1,34 | 1,24 | 0,87 | 1,17 | 1,00 | 0,84 | 0,86 | 1,03 |

Table 2 Hamming distances in percents computed in the case of the window length of 13 amino acids between the classes of PPII and non-PPII, and within PPII.

| type | | | | | distance | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| between classes | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 15 | 67 | 15 |
| within PPII | 7 | 4 | 4 | 3 | 4 | 0 | 6 | 22 | 45 | 5 |

Figure captions

Figure 1. Two windowing techniques are presented in connection with the window length of 13 amino acids. The grey positions indicate localized PPII structures. The first technique accepts a string of the exact window length to the PPII class if the structure is located at the position of the middlemost amino acid. The second technique accepts a string to the PPII class if the structure is within the three middlemost amino acids. Otherwise, the content of the window is determined to the non-PPII class.

Figure 2. A sequence is encoded to the form (bit vector) “understood” by the neural network. The angles of the structure file in the polygon implies a PPII structure. Amino acids K, A, and P are set to PPII active. Sequences are assigned to the PPII and non-PPII classes by applying the window. From each window a long bit vector of ones and zeros are input to the neural network.

Figure 3. Relatedness frequencies of the test material. Relatedness of amino acids was computed between the cases of the PPII and non-PPII classes in the upper part and within the PPII class in the lower part. Relatedness between two sequences was computed by comparing amino acids at the same positions of the two sequences, and these values were summed up to be the similarity value.

Figure 1

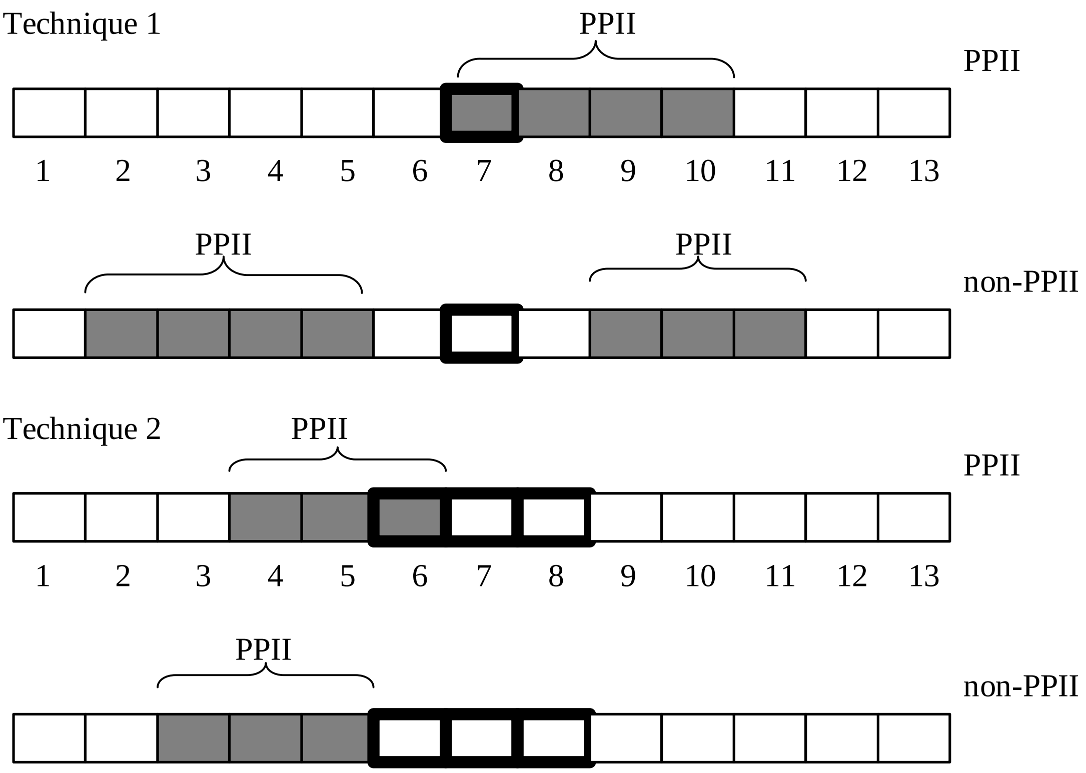
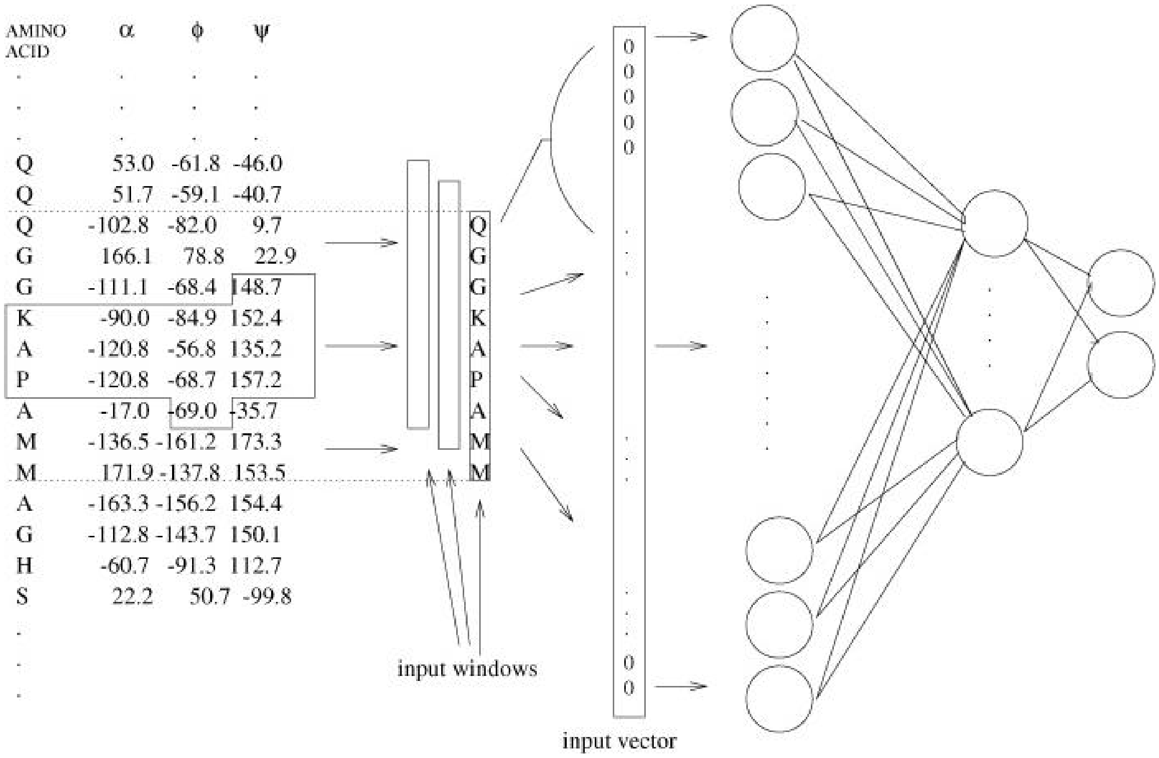
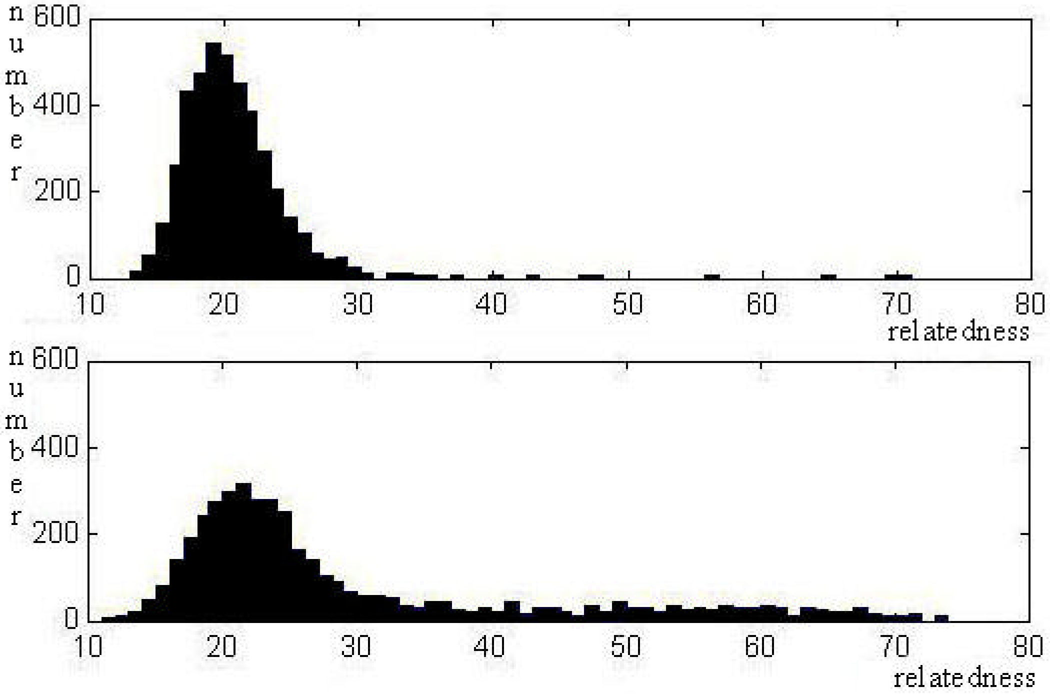

Technique 1
PPII
PPII
1
2
3
4
5
6
7
8
9
10
11
12
13
PPII
PPII
non-PPII
Technique 2
PPII
PPII
1
2
3
4
5
6
7
8
9
10
11
12
13
PPII
non-PPII

Figure 2



| AMINO ACID | $\alpha$ | $\phi$ | $\psi$ |
|---|---|---|---|
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| Q | 53.0 | -61.8 | -46.0 |
| Q | 51.7 | -59.1 | -40.7 |
| Q | -102.8 | -82.0 | 9.7 |
| G | 166.1 | 78.8 | 22.9 |
| G | -111.1 | -68.4 | 148.7 |
| K | -90.0 | -84.9 | 152.4 |
| A | -120.8 | -56.8 | 135.2 |
| P | -120.8 | -68.7 | 157.2 |
| A | -17.0 | -69.0 | -35.7 |
| M | -136.5 | -161.2 | 173.3 |
| M | 171.9 | -137.8 | 153.5 |
| A | -163.3 | -156.2 | 154.4 |
| G | -112.8 | -143.7 | 150.1 |
| H | -60.7 | -91.3 | 112.7 |
| S | 22.2 | 50.7 | -99.8 |
| . | | | |
| . | | | |
| . | | | |

Figure 3



number
600
400
200
0
10
20
30
40
50
60
70
80
relatedness
number
600
400
200
0
10
20
30
40
50
60
70
80
relatedness