J Mol Model (2001) 7:360–369
DOI 10.1007/s008940100038

ORIGINAL PAPER

**Jens Meiler · Michael Müller · Anita Zeidler
Felix Schmäschke**

Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks

Received: 10 January 2001 / Accepted: 21 May 2001 / Published online: 22 September 2001

Abstract In order to process data of proteins, a numerical representation for an amino acid is often necessary. Many suitable parameters can be derived from experiments or statistical analysis of databases. To ensure a fast and efficient use of these sources of information, a reduction and extraction of relevant information out of these parameters is a basic need. In this approach established methods like principal component analysis (PCA) are supplemented by a method based on symmetric neural networks. Two different parameter representations of amino acids are reduced from five and seven dimensions, respectively, to one, two, three, or four dimensions by using a symmetric neural network approach alternatively with one or three hidden layers. It is possible to create general reduced parameter representations for amino acids. To demonstrate the ability of this approach, these reduced sets of parameters are applied for the ab initio prediction of protein secondary structure from primary structure only. Artificial neural networks are implemented and trained with a diverse representation of 430 proteins out of the PDB. An essentially faster training and also prediction without a decrease in accuracy is obtained for the reduced parameter representations in comparison with the complete set of parameters. The method is transferable to other amino acids or even other molecular building blocks, like nucleic acids, and therefore represents a general approach.

Keywords Amino acid parameters · Neural networks · Quantitative structure–property relation · Secondary structure prediction

Electronic supplementary material to this paper can be obtained by using the Springer LINK server located at http://dx.doi.org/10.1007/s008940100038

J. Meiler (✉) · M. Müller · A. Zeidler · F. Schmäschke
Howard Hughes Medical Institute, University of Washington,
Box 357350, Seattle, Washington 98195-7350, USA
e-mail: jens@jens-meiler.de
Tel.: +1-206-5431295, Fax: +1-206-6851792

Introduction

Artificial neural networks are now a widespread and intensively discussed method for analyzing data and describing relationships in chemistry. [1, 2] Neural networks are often the preferred solution if the dependence cannot be expressed by a simple mathematical equation or this equation is unknown and also not important for solving the problem. Moreover, neural networks are particularly suitable for working with blurred information and are able to apply learned correlations to unknown examples. Many applications of these networks already exist in chemistry [3, 4, 5, 6, 7, 8, 9] and in particular also for secondary structure prediction of proteins. [10, 11, 12, 13, 14, 15, 16]

However, the focus of this paper is the introduction of a so far not intensively discussed possibility of using neural networks: reducing the number of dimensions for a parameter representation as traditionally performed by principal component or cluster analyses. The method was introduced by Livingstone et al. [17] and also used by Kocjancic and Zupan [18] as a mapping device. Here it is applied to reduce the dimension of property representations of amino acids. The potential of these reduced parameter representations is demonstrated by comparing them with the complete property representations in their ability to serve as input for another neural network that predicts the secondary structure of proteins from primary structure only.

System and methods

A dataset of *l* individuals containing *m* properties for each of these individuals is called a *m*-dimensional property representation of these *l* individuals. These *m* properties can be projected into *n* dimensions with *n*&lt;*m* by principal component analysis or cluster analysis. This is of special interest for obtaining linear dependencies between these properties and of course for visualizing relations between the *l* individuals. However, these purposes can also be obtained using artificial neural networks. As given in

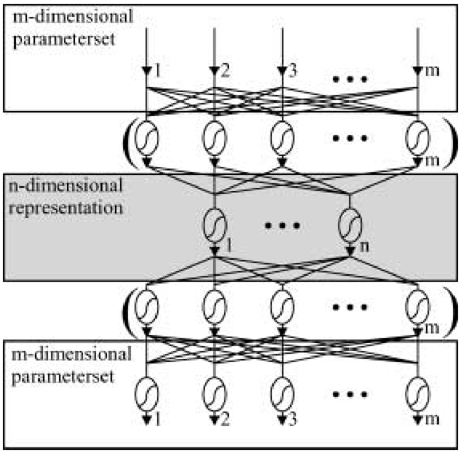
m-dimensional parameterset
1
2
3
• • •
m
m
n-dimensional representation
• • •
1
n
m
m-dimensional parameterset
1
2
3
• • •
m

Fig. 1 An *m*-dimensional parameter representation for an amino acids is presented to an artificial neural network with three or five hidden layers. The data are processed through a central hidden layer with $n \leq m$ neurons and recalculated by the output layer containing again *m* neurons. The network is trained with parameter representations for all 20 amino acids that occur in natural systems

Fig. 1, a three-layer neural network can be built with *m* inputs and *m* output neurons but only *n* hidden neurons. Due to their symmetric architecture (*m* inputs, *n* hidden neurons, *m* output neurons), these networks are called symmetric networks. This is in one sense a critical name since the nodes in the input layer act mainly as distributors. Therefore, they are also not regarded as neurons in Fig. 1. They process the non-weighted input data without summarizing and with a linear transfer function $y=x$ to pass the information to the hidden neurons. So the three basic features of an artificial neuron, the summation of previously weighted information to process it with a transfer function, are hardly recognizable in this case. Therefore, the nodes of the first layer have a different structure than the neurons in further layers. Thus the phrase “symmetric” just targets the symmetric distribution of all neurons including the nodes of the input layer relative to the central hidden layer. It does not reflect the fact that the nodes of the input layer have a significantly different structure than the neurons of the output layer. Moreover, the weights are also not restricted to be symmetric in their value with respect to the central hidden layer.

However, training this network with the *m* properties of *l* individuals to predict again the same *m* properties provides a network where in the hidden layer all the information is represented by *n* numbers. If now these *n* numbers for the *l* individuals are obtained, an *n*-dimensional representation of the *m* properties for each of the *l* individuals is found. Depending on the transfer function, the representing parameters lay between 0 and 1 (e.g.$\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$) or between –1 and 1 (e.g. $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$). The completeness of this representation can be obtained by investigating the deviation of the original properties with the properties back- calculated by the three-layer network.

Using a neural network with only one hidden layer, the dependence of the reduced parameter set p'_j ($1 \leq j \leq n$) on the original one p_i ($1 \leq i \leq m$) is linear except for the sigmoidal transfer function:

$$p'_j = \left(1 + e^{\left(- \sum_{i=1}^{m} w_{ij} p_i \right)} \right)^{-1}$$

Also the back transformation of the reduced parameters to the original uses the same “pseudolinear” equation. Thus, the similarity of this approach to a principal component analysis is obvious. Each of the derived parameters is a linear combination of the original parameters processed with the transfer function.

To analyze the data due to a more complex dependence, a five-layer network can be created, again with *m* input and output neurons and *n* neurons in the third layer as given in Fig. 1. Now an additional hidden layer allows a modification of the data before the reduced parameter representations are obtained in the central hidden layer and another hidden layer is inserted after the central hidden layer (Fig. 1). The model is much more complex than linear and therefore able to simulate even complicated polynomial functions because of the very flexible network model.

The number of neurons in these additional hidden layers could obviously be varied between *n* and *m*. Less than *n* neurons in these layers would force the network to reduce the dimension in the second and the fourth layer even further than just to *n* dimensions, whereas more than *m* neurons would distribute *m* input values in even more parameters in the second and the fourth layer. As expected, the lowest RMSD value in the optimization of these networks is obtained using *m* neurons in the second and in the fourth layer. In this case one working neuron in the additional layers is provided for every input and output neuron to process the information. The two latter points justify the use of *m* neurons in the second and the fourth layer. Since it is the largest possible value between *n* and *m* it is consequently the most complex model and therefore capable of achieving the biggest contrast to the three-layer case. Again the property information is represented by only *n* numbers in the third layer.

A more detailed description of the method as well as a comparison to other methods can be found in the literature. [17, 18, 19]

Algorithm and implementation

The generation and the testing of the reduced parameter representations is performed in two consecutive steps:

- Training of symmetric neural networks with property representations of all twenty naturally occurring amino acids and obtaining the reduced parameter representations from these networks.
- Testing the reduced parameter representations by comparison with the complete parameter representa-

Table 1 Amino acid parameter sets

Name	Ξ[a]	α[b]	υ_v[c]	π[d]	I[e]	α[f]	β[g]
ALA	1.28	0.05	1.00	0.31	6.11	0.42	0.23
GLY	0.00	0.00	0.00	0.00	6.07	0.13	0.15
VAL	3.67	0.14	3.00	1.22	6.02	0.27	0.49
LEU	2.59	0.19	4.00	1.70	6.04	0.39	0.31
ILE	4.19	0.19	4.00	1.80	6.04	0.30	0.45
PHE	2.94	0.29	5.89	1.79	5.67	0.30	0.38
TYR	2.94	0.30	6.47	0.96	5.66	0.25	0.41
TRP	3.21	0.41	8.08	2.25	5.94	0.32	0.42
THR	3.03	0.11	2.60	0.26	5.60	0.21	0.36
SER	1.31	0.06	1.60	-0.04	5.70	0.20	0.28
ARG	2.34	0.29	6.13	-1.01	10.74	0.36	0.25
LYS	1.89	0.22	4.77	-0.99	9.99	0.32	0.27
HIS	2.99	0.23	4.66	0.13	7.69	0.27	0.30
ASP	1.60	0.11	2.78	-0.77	2.95	0.25	0.20
GLU	1.56	0.15	3.78	-0.64	3.09	0.42	0.21
ASN	1.60	0.13	2.95	-0.60	6.52	0.21	0.22
GLN	1.56	0.18	3.95	-0.22	5.65	0.36	0.25
MET	2.35	0.22	4.43	1.23	5.71	0.38	0.32
PRO	2.67	0.00	2.72	0.72	6.80	0.13	0.34
CYS	1.77	0.13	2.43	1.54	6.35	0.17	0.41

[a] Steric parameter (graph shape index)
[b] Polarizability
[c] Volume (normalized van der Waals volume)
[d] Hydrophobicity
[e] Isoelectric point
[f] Helix probability
[g] Sheet probability

tions. Therefore both parameter representations are used to code protein sequences and their secondary structure is predicted by training another artificial neural network with these numbers.

For the calculation of the reduced parameter sets, five properties of amino acids are used: a steric parameter, hydrophobicity, volume, polarizability, [20] and isoelectric point (Table 1).

The steric parameter is the graph shape index Ξ. This parameter encodes complexity, branching, and symmetry of a group and can be calculated directly from the graph structure of the amino acid side chain.

The hydrophobicity π is defined as a side chain parameter as π(side chain)=log *P*(amino acid)–log *P*(glycine) in which *P* is the partition coefficient of the amino acid in octanol/water.

The normalized van der Waals volume υ_v is defined by υ_v(side chain)=[*V*(side chain)–*V*(H)]/*V*(CH_2). The measure is therefore 0 for glycine and 1 for alanine.

The polarizability α is related to the molar refractivity. It is given by:

$$\alpha = \frac{3}{4\pi N} \cdot \frac{M}{d} \cdot \frac{n^2 - 1}{n^2 + 2}$$

(*n*: index of refraction, *M*: molecular weight, *d*: density, *N*: number of atoms).

These values are used alone (five-parameter set) as well as in combination with two statistical parameters: helix and sheet probability (seven-parameter set, Table 1). The secondary structure probabilities are extracted from a subspace of the PDB [21] containing 430 proteins with over 60,000 residues. Thus, every one of the 20 naturally occurring amino acids is represented by five or seven properties, respectively.

Overall, 24 small neural networks were trained with different number of hidden neurons to obtain the reduced parameter representations:

Number of hidden neurons	Three layer network	Five layer network
Five parameter set	1, 2, 3, 4, 5	1, 2, 3, 4, 5
Seven parameter set	1, 2, 3, 4, 5, 6, 7	1, 2, 3, 4, 5, 6, 7

For the mapping procedure, no testing set of data is required. All 20 amino acids are therefore a part of the training set of data. The training is continued until the root mean square deviation of the recalculated property values is minimized. After the training process the networks are “cut” after the second (three-layer network) or third layer (five-layer network) to obtain the reduced parameter sets. The central hidden layer becomes the output layer and the values detected by applying the property values at the inputs for all 20 amino acids provide the parameter representations.

In order to test the information conserved in these reduced parameter representations, neural networks were trained to predict secondary structure of proteins from primary structure only: basically the secondary structure is predicted for every amino acid in one run. The sequence information is provided as input for a symmetric window around this amino acid of interest. The secondary structure of a protein is calculated by moving this window over the sequence and calculating the secondary structure for every amino acid individually. This concept of a moving window has been widely used for this purpose (e.g. [10]).

The input values for every amino acid are the *m* properties of the amino acid or their reduced *n*-dimensional parameter representations. The number of values necessary to describe one amino acid is therefore not fixed but varies between one and seven. In our example the size of the window was optimized to contain the central amino acid as well as 15 amino acids before and after it (Fig. 2). Thus, this initial window has a size of 31 amino acids. This is a compromise between a window as large as possible to provide the most possible complete sequence information and a preferably small input layer to minimize the number of connections. Since the number of connections is equal to the degrees of freedom in the artificial neural network, this parameter determines the necessary training information (number of sequences with known secondary structure) as well as the time for the training procedure.

Although the next neighbors of the amino acid of interest have a larger impact on the formation of secondary structure fragments, the part of the sequence not represented by the 31 amino acid window also influences the formation of secondary structure. This is because the

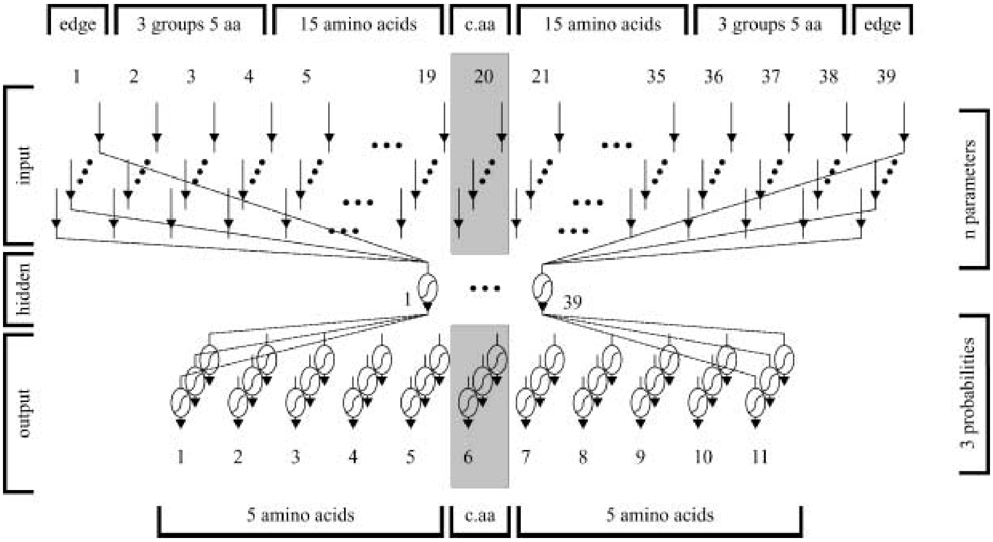

Fig. 2 Three-layer network for predicting secondary structure of amino acids is visualized. 39 *n*-dimensional parameter or *m*-dimensional property representations of amino acids are presented to the network. The number of input neurons is therefore 39*n* or 39*m*, respectively. The *gray shaded* central amino acid is surrounded by 15 amino acids represented by individual parameters as well as 15 additional amino acids represented by average parameters computed for three groups of five amino acids, respectively. All other amino acids surrounding this window of 61 amino acids are represented by the average value of their parameters ("edge"). The data pass a hidden layer containing 39 hidden neurons and the network is trained to predict probabilities for α-helix, β-sheet, and unknown secondary structure for the central amino acid as well as the next five amino acids on both sides of the central amino acid

secondary structure is influenced not only by short primary structure sequences but also by long-range interactions that occur in the tertiary structure and can therefore contain interactions between amino acids that are far apart in the primary sequence.

To enable the network to use at least some information about the parts of the sequence not considered, all amino acids before and after this window of 31 amino acids were incorporated into average property and parameter values. These averages somehow represent the character of the remaining part of the protein that surrounds the 31-amino acid window (e. g. mainly aliphatic amino acids, pH character,...). The averages are computed for four groups of amino acids before and after the 31-amino acid window, respectively. The first three groups contain the average values for five amino acids each. They therefore hold information for another 30 amino acids, 15 on each side of the window. The fourth group on both sides contains the average values of all remaining amino acids before or beyond this window of now 61 amino acids to the start or to the end of the sequence. As visualized in Fig. 2 this leads to 31+8=39 groups of input properties (or parameters). For every one of these 39 groups either *m* properties or *n* parameters are used as description, which leads to 39·*m* and 39·*n* input neurons, respectively. The use of these additional four groups with averaged parameters on both sides leads to an overall improvement of about 2% in the prediction (compare with the Q_3 values discussed later).

All three-layer neural networks contain 39 hidden neurons. This number is optimized for the network trained with the seven-property representation and remains constant to ensure comparable conditions for all experiments.

Probabilities for being a part of an α-helix, β-sheet, or coil for every amino acid (in the range of 0–1) are obtained at the output layer. "Coil" covers in our case all other secondary structure elements, loops, and turns except α-helix and β-sheet. For the training of these networks these probabilities are set to be "1 0 0", "0 1 0", or "0 0 1", respectively.

Optimizing these networks shows that an increase of correctly predicted secondary structure is obtained by predicting the secondary structure of more than one amino acid in one run. The optimum is found by calculating probabilities for five amino acids before and after the central amino acid, respectively. Therefore, 11 amino acid probability sets are predicted parallel in one network run, which leads to 33 output neurons for all neural networks (Fig. 2). By moving the window over the amino acid sequence, every single amino acid is part of the output window exactly eleven times. These 11 predicted probabilities for one amino acid are combined by a triangular weighted average. The prediction is weighted with 1 if the amino acid is the central one and the weight is reduced as the amino acid moves to the edges of this window of 11 amino acids. The vector of the 11 weights is consequently *w*=(0.166, 0.333, 0.500, 0.666, 0.833, 1.000, 0.833, 0.666, 0.500, 0.333, 0.166). The three probabilities are computed by

$$p^{\mathrm{H,S,C}} = \sum_{i=1}^{11} p_i^{\mathrm{H,S,C}} w_i \Big/ \sum_{i=1}^{11} w_i$$

where $p_i^{\mathrm{H,S,C}}$ are the eleven predicted probabilities for an amino acid to be part of a helix, a sheet, or a coil region.

This procedure allows us to correct a possible wrong judgment for the central amino acid alone. In about 3% of all cases this correction takes place and the overall accuracy is therefore improved by 3% using this procedure (compare with the Q_3 values discussed later).

All in all 16 neural nets using different property and parameter representations as input are trained. Two networks with the complete five- and seven-parameter representations as well as 14 networks using one-, two-, three-, and four- (only for the seven-property representation) dimensional parameter representations are obtained from three- and five-layer networks with five and seven properties used for training. To ensure the use of all known folds in this procedure, the FSSP database (http://www.ebi.ac.uk/dali/fssp/) introduced by Holm and Sander [21] is used.

For the training of these networks, 430 peptides derived from this database are separated into two sets of data: first with 95% of the peptides for training the networks and second with the remaining 5% for testing. The networks are trained until the root mean square deviation of the testing data set is minimized. Probabilities between 0 and 1 are obtained using a sigmoidal transfer function and back-propagation of errors as the training method. All neural networks are trained and analyzed using the program “Smart”. [22]

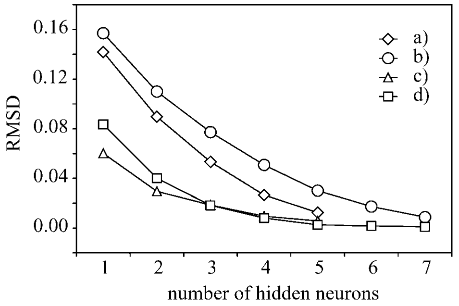

Fig. 3 Final RMSD values for neural network back-calculation of an *m*-dimensional parameter set drawn on the *y*-axes. The number of hidden neurons used is given on the *x*-axes. The number of layers and the parameter set used are varied according to: *a*) three-layer networks trained with five-parameter set; *b*) three-layer networks trained with seven-parameter set; *c*) five-layer networks trained with five-parameter set; *d*) five-layer networks trained with seven-parameter set. The assigned letters correspond to the letters used in Tables 2, 3, and 4

Discussion

Figure 3 gives the root mean square deviation of the recalculated and normalized property values depending on the number of layers, number of hidden neurons and number of properties. As expected, a network with *m* hidden neurons is able to recalculate the *m* given properties totally. The small deviation obtained for the networks with five or seven hidden neurons, respectively, is due to the use of an optimization instead of a direct calculation of the weights. However, differences are observed for networks with a reduced number of hidden neurons. First of all, networks with five layers are able to reproduce the data with smaller deviation than nets with only three layers using the same number of hidden neurons. This is in line with expectations because of the ability of these nets to simulate more complicated dependencies and proves that in these cases more information can be projected into an *n*-parameter representation. In this case for both the five- and the seven-property representation, a nearly complete description of the properties is given by only three numbers. The RMSD is about 0.020 in both cases. The RMSD for the three-layer networks is significantly larger with about 0.053 and 0.080 for the five-property and for the seven-property fit, respectively. These values are in the order of the according networks with only one hidden neuron but five layers. However, one has to keep in mind that in the networks with five layers, two layers are used to recalculate the property values from the parameter representation instead of only one layer in the three-layer network. Thus, the recalculation again uses a much more complicated model and of course more weights. The improvement of the prediction going from five to seven properties and especially for going from three- to five-layer networks is clearly observable.

All reduced parameter representations derived are given in Tables 2 and 3, and some examples are visualized in Fig. 4. Groups of amino acids with similar properties are plotted closer together and become better separated by increasing the number of layers from three to seven and also by using seven instead of five properties, even for the one-dimensional representations.

For example, the aromatic amino acids are all plotted together in a range of 0.20 in the three-layer case but in a range of 0.05 in the five-layer case. Also, the basic ami-

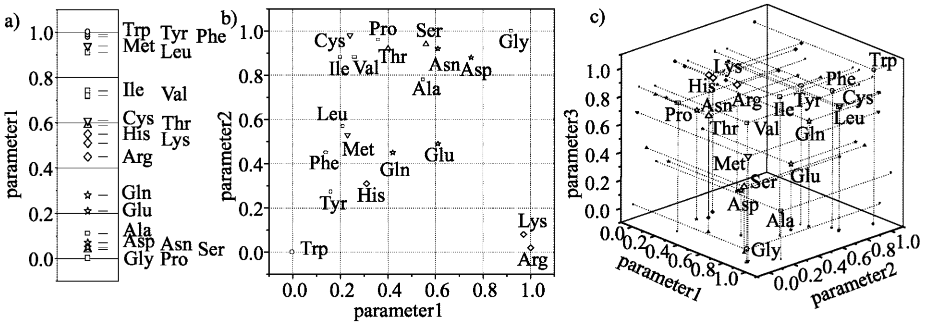

Fig. 4 Obtained reduced parameters for the five-layer neural network mapping the seven-parameter set in one (**a**), two (**b**), or three (**c**) dimensions

Table 2 Reduced parameter representation obtained from the three-layer network

Name	(a) Three-layer networks trained with five-parameter set						(b) Three-layer networks trained with seven-parameter set									
	1D	2D		3D			1D	2D		3D			4D			
ALA	0.13	0.15	0.04	0.20	0.84	0.27	0.28	1.00	0.52	0.84	0.50	0.00	1.00	1.00	0.98	0.56
GLY	0.00	0.00	0.03	0.03	1.00	0.32	0.00	0.99	1.00	0.04	0.05	0.20	0.29	0.76	0.18	0.07
VAL	0.55	0.57	0.04	0.69	0.59	0.18	0.76	0.19	0.68	0.68	0.20	0.87	0.38	0.73	0.31	0.93
LEU	0.61	0.62	0.03	0.71	0.53	0.20	0.69	0.45	0.44	1.00	0.43	0.49	0.79	0.82	0.73	0.87
ILE	0.75	0.73	0.03	0.91	0.50	0.16	0.84	0.13	0.60	0.88	0.25	0.88	0.47	0.73	0.38	1.00
PHE	0.81	0.80	0.11	0.83	0.25	0.32	0.83	0.17	0.52	0.88	0.39	0.83	0.43	0.33	0.32	0.78
TYR	0.76	0.75	0.28	0.66	0.09	0.45	0.81	0.18	0.52	0.56	0.44	0.92	0.18	0.03	0.12	0.54
TRP	1.00	1.00	0.20	1.00	0.03	0.32	1.00	0.00	0.36	0.96	0.51	1.00	0.35	0.09	0.31	0.86
THR	0.36	0.37	0.11	0.40	0.56	0.36	0.50	0.47	0.76	0.40	0.22	0.68	0.26	0.42	0.14	0.48
SER	0.15	0.15	0.09	0.14	0.72	0.41	0.26	0.77	0.84	0.24	0.21	0.39	0.32	0.55	0.22	0.26
ARG	0.46	0.40	1.00	0.06	0.00	0.00	0.51	0.92	0.00	0.00	1.00	0.52	0.05	0.21	1.00	0.13
LYS	0.33	0.27	0.82	0.00	0.16	0.05	0.42	0.93	0.20	0.00	0.84	0.47	0.08	0.36	0.87	0.14
HIS	0.52	0.50	0.43	0.37	0.25	0.18	0.57	0.56	0.44	0.32	0.56	0.68	0.18	0.30	0.47	0.38
ASP	0.16	0.20	0.19	0.09	0.44	0.98	0.23	0.87	0.76	0.48	0.40	0.18	0.53	0.00	0.04	0.00
GLU	0.24	0.28	0.24	0.14	0.31	1.00	0.36	0.95	0.40	0.88	0.74	0.00	0.94	0.09	0.46	0.14
ASN	0.21	0.20	0.34	0.09	0.44	0.41	0.25	0.86	0.68	0.12	0.43	0.40	0.19	0.24	0.27	0.06
GLN	0.31	0.32	0.30	0.20	0.31	0.55	0.43	0.84	0.40	0.64	0.67	0.24	0.59	0.30	0.59	0.26
MET	0.58	0.58	0.09	0.60	0.41	0.34	0.67	0.48	0.40	0.96	0.52	0.48	0.74	0.58	0.66	0.73
PRO	0.34	0.33	0.01	0.43	0.72	0.14	0.45	0.44	0.92	0.08	0.00	0.82	0.00	0.58	0.00	0.46
CYS	0.42	0.42	0.00	0.57	0.75	0.16	0.57	0.29	0.88	0.44	0.04	0.81	0.27	0.82	0.20	0.75

Table 3 Reduced parameter representation obtained from the five-layer network

Name	(c) Five-layer networks trained with five-parameter set						(d) Five-layer networks trained with seven-parameter set									
	1D	2D		3D			1D	2D		3D			4D			
ALA	0.01	0.13	0.06	0.23	0.19	0.19	0.11	0.55	0.78	1.00	0.19	0.25	0.56	1.00	0.86	0.19
GLY	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.92	1.00	0.90	0.00	0.00	0.05	0.70	0.59	0.17
VAL	0.93	0.26	0.11	0.89	0.29	0.84	0.72	0.26	0.88	0.71	0.18	0.79	0.48	0.77	0.30	0.99
LEU	0.94	0.78	0.23	0.98	0.39	0.48	0.91	0.21	0.57	0.90	0.76	0.79	0.70	0.78	0.89	0.49
ILE	0.94	0.44	0.15	1.00	0.29	1.00	0.74	0.20	0.88	0.71	0.45	0.89	0.94	0.83	0.35	1.00
PHE	0.96	0.92	0.23	0.93	0.59	0.36	0.98	0.14	0.45	0.83	0.77	0.88	0.69	0.50	0.81	0.58
TYR	0.96	0.94	0.30	0.50	0.75	0.16	0.99	0.16	0.27	0.66	0.67	0.89	0.58	0.23	0.70	0.52
TRP	1.00	0.98	0.15	0.95	1.00	0.11	1.00	0.00	0.00	0.95	1.00	1.00	1.00	0.37	0.89	0.66
THR	0.74	0.18	0.13	0.41	0.32	0.52	0.59	0.40	0.92	0.37	0.18	0.73	0.23	0.55	0.38	0.67
SER	0.03	0.12	0.09	0.16	0.30	0.10	0.04	0.56	0.94	0.76	0.11	0.37	0.19	0.55	0.57	0.41
ARG	0.50	1.00	1.00	0.05	0.96	0.00	0.45	1.00	0.02	0.00	0.76	0.66	0.34	0.00	1.00	0.01
LYS	0.47	0.97	1.00	0.02	0.75	0.00	0.51	0.97	0.08	0.15	0.42	0.86	0.27	0.11	0.86	0.08
HIS	0.69	0.90	0.40	0.30	0.65	0.05	0.55	0.31	0.31	0.20	0.34	0.92	0.39	0.25	0.76	0.39
ASP	0.09	0.07	0.40	0.23	0.27	1.00	0.07	0.75	0.88	0.54	0.29	0.22	0.51	0.41	0.11	0.00
GLU	0.12	0.13	0.45	0.27	0.40	0.76	0.21	0.61	0.49	0.68	0.57	0.37	0.90	0.87	0.32	0.01
ASN	0.37	0.59	0.55	0.11	0.49	0.04	0.05	0.61	0.92	0.17	0.27	0.68	0.19	0.18	0.68	0.18
GLN	0.34	0.67	0.45	0.20	0.55	0.09	0.28	0.42	0.45	0.71	0.69	0.65	0.46	0.49	0.84	0.21
MET	0.94	0.84	0.30	0.73	0.52	0.21	0.94	0.23	0.53	0.88	0.78	0.79	0.66	0.67	0.89	0.42
PRO	0.77	0.18	0.09	0.70	0.17	0.71	0.00	0.36	0.96	0.24	0.05	0.82	0.00	0.82	0.00	0.99
CYS	0.92	0.21	0.02	0.93	0.30	0.44	0.61	0.24	0.98	0.88	0.03	0.65	0.09	0.54	0.65	0.74

no acids are plotted closer together and separated clearly from the rest. Only histidine is found between the aromatic and the basic amino acids because of its ambiguous character. Glutamine and asparagine are plotted closer together and become clearly separated from glutamic and asparagic acid in going from three- to five-layer networks. While methionine and cysteine are also plotted close together, aliphatic amino acids are relatively widespread but sorted by the size of the side chains. Also serine and threonine are projected with a large difference. This marks the incomplete representation given in this one dimension and is improved by introducing further parameters. This incompleteness also causes the ambiguous position of proline in the one-dimensional representations (0.34, 0.45, 0.77, 0.00). The network obviously learns the large and easily obtainable differences first. The single neuron in the hidden layer reaches its capacity fast, and the proline parameter becomes ambiguous. The RMSD values of the back-calculated properties (Fig. 3) prove that only a part of the information can be saved by the network. However, an increase in the number of dimensions will overcome this problem and the parameters for proline will also become rather well defined. The still observable usually relatively small differences between the individual parameter representations are then only based on differences in the applied property sets (five- versus seven-property representation) or on different models (three- versus five-layer network).

For the two- and three-dimensional parameter representations, an improvement can again be obtained using two additional hidden layers. Moreover, the representation changes significantly going from five to seven properties. A better use of space while using the seven property values and a clearer separation of the amino acid groups is obtained by increasing the number of layers in the network.

Beside the information obtainable by this projection method about similarities in a data set, these reduced parameter representations given in Tables 2 and 3 can be used as general reduced parameter sets for representing amino acids. Using the trained and cut neural networks, the same parameter representation can also be calculated for other amino acids. Of course, the parameters represent a combination of the properties used and therefore they cannot be directly interpreted as easily understandable properties. Their advantage is the reduction in number. Visualization becomes possible and allows a graphical analysis of the parameter space. Calculation time can be saved by using, for example, three instead of seven parameters. For a 200-amino acid protein the representing code can thus be reduced from 1,400 using all seven properties to only 600 numbers using a three-dimensional parameter representation. In our special case, the calculation of all secondary structure probabilities for the database with 430 proteins can be reduced from 120 s for the seven property values to 50 s for the three-dimensional parameter representation. However, this is only a relative small improvement considering the high speed of computers today. Much more impressive is the gain of time in the training process. The necessary time for stabilizing the weights does not depend linearly on the number of weights, but increases much faster with higher number of weights. This is difficult to calculate in general, but for the same example the complete training process lasts about 24 h for the seven property values but less than 4 h in the case of the three-dimensional parameter representation. (All CPU times are obtained from a 450 MHz Pentium II processor equipped with 512 MB RAM.)

The method could also become more effective if a higher number of parameters can be projected in two or three dimensions. Considering the enormous number of amino acid and nucleic acid sequences, the potential of this method to provide general parameter representations combining a large number of relevant properties is remarkable and can lead to better mapping pictures as well as substantial gain in processing time.

Moreover, the influence of each property on a parameter can be extracted by a “sensitivity analysis” of the neural network. In order to do this, the value for each input is varied within the experimental input range, while all other input are set to be zero. The covered range of the output neurons gives a sensitivity value between 0 and 1. If the influence of a particular input on a particular output value is high, the output range covered is large and the sensitivity becomes 1. However, if the input has no influence at all, the output value will remain constant and the sensitivity stays 0.

It must be mentioned that this method gives only a qualitative picture for several reasons: the unchanged input values are set to be zero, which is just one realistic input signal. However, there are usually a lot more realistic input values, which are not used for this analysis. The use of another static realistic input signal might influence the sensitivities obtained. Moreover, due to the constant signals chosen for all other neurons, the method is unable to detect cross-correlation effects between different input values. However, the method works reliably in our hands. The sensitivity values change below 10% by setting the unchanged input data to other realistic input values. The relations between the different sensitivities, which change by less than 5%, are preserved even better.

The one-dimensional parameter representation is found to be dominated by the volume, while, for example, the isoelectric point has no influence at all. In the two-parameter representation the first parameter is still dominated by the volume but the isoelectric point is projected into the second parameter together with the polarizability, which is also only badly represented in the one-dimensional parameter. The three-parameter representation takes all the five parameters into consideration. The possibility of projecting a part of the steric information together with hydrophobicity in parameter 1, and polarizability, volume, and again hydrophobicity in parameter 2 is in line with the empirical understanding of these parameters.

The secondary structure prediction of proteins was chosen as an example problem in order to test the reduced parameter representation. Methods for predicting the secondary structure of proteins have been discussed widely and are therefore optimal for testing the derived amino acid parameters. However, this prediction is carried out to compare the results for the full and reduced parameter representations only and not to achieve an optimal secondary structure prediction. Therefore, the setup was not optimized to give the best possible results.

The Q_3 values (as introduced in the literature [23]) for the 16 trained neural networks obtained for the test data set are given in Table 4. The results for the training data set are slightly better than those obtained for the testing data as expected, and therefore not reported here. The fraction of correct predictions achievable with this straightforward use of the primary sequence of one protein only is 67% for the total seven-parameter representation and 63% for the five-parameter representation. However, the three- and four-dimensional reduced parameter representations give results of the same quality.

Table 4 Results for secondary structure prediction of proteins from test data set using the reduced parameter representation in comparison with the complete parameter set

Dimension of parameter representations		Prediction of artificial neural networks (%)									
		α-helix predicted as			β-sheet predicted as			Coil predicted as			Q_3 (Σ)
		Helix	Sheet	Coil	Helix	Sheet	Coil	Helix	Sheet	Coil	Correct
(a)	1	9.4	3.6	16.8	3.6	8.8	14.7	3.1	4.4	35.5	53.8
	2	10.8	3.4	15.7	2.4	10.9	13.7	3.1	3.7	36.3	58.0
	3	14.3	3.0	12.6	2.2	12.0	12.9	2.8	3.9	36.4	62.7
(b)	1	9.7	4.0	16.2	3.9	10.0	13.2	3.4	4.8	34.8	54.5
	2	16.1	2.9	10.9	1.9	10.2	15.0	4.6	5.6	32.9	59.1
	3	19.6	2.2	8.1	1.3	11.7	14.0	4.0	4.6	34.4	65.7
	4	19.5	2.8	7.6	2.3	12.2	12.5	4.4	4.7	34.0	65.7
(c)	1	7.7	3.6	18.6	1.9	10.6	14.5	3.5	4.4	35.2	53.5
	2	13.8	1.9	14.1	1.9	7.3	17.9	3.7	3.1	36.2	57.3
	3	16.4	2.3	11.1	1.8	12.2	13.1	4.5	3.7	34.8	63.4
(d)	1	11.8	2.7	15.4	3.3	8.7	15.1	3.7	4.0	35.3	55.8
	2	15.4	2.2	12.3	2.8	9.8	14.4	5.2	4.0	33.8	59.0
	3	20.3	1.7	7.9	3.5	9.8	13.8	4.5	3.2	35.3	65.3
	4	20.0	1.8	8.0	1.8	12.0	13.2	4.4	4.9	33.8	65.8
Five-parameter set		16.1	2.0	11.8	2.6	11.5	13.0	3.6	4.2	35.3	62.8
Seven-parameter set		20.1	2.2	7.6	3.3	12.4	11.3	4.1	4.4	34.6	67.1

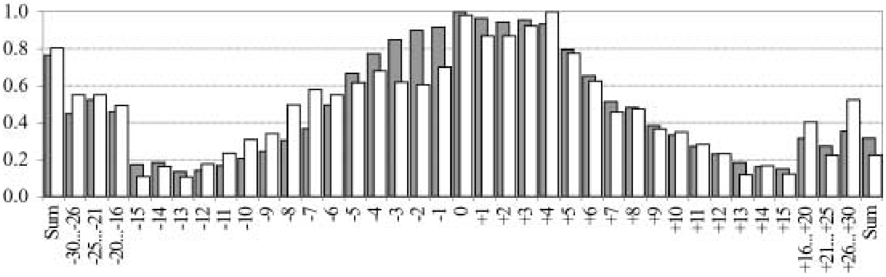

Fig. 5 Input sensitivity for the 39 input blocks of the neural network predicting secondary structure probabilities from the seven-parameter set of amino acids. The sensitivity is normalized to give 1 for the central amino acid. Summarized is over all output values (*gray bars*) and only over the helix probability (*white bars*)

The prediction accuracy remains in a range of ±2% constant in all cases. In the case of the five-layer three-dimensional representation it even becomes slightly better than the complete parameter representation.

The prediction accuracy increases by about 5% with every new dimension introduced for the first three dimensions. No further increase is obtained on introducing further dimensions. Since the prediction accuracy reaches the level obtained for the full parameter representations, the optimal prediction using this method of coding is reached by introducing the third parameter. The completeness of the description with these three parameters is shown by analyzing the RMSD values in Fig. 3 and now also becomes visible in these results.

It is not surprising that the reduced parameter set obtained from the five-layer networks gives only slightly better results than that obtained from the three-layer networks. The step of data interpretation not performed in the latter case can be completed by the three-layer neural network used for deriving secondary structure information. Thus, a more significant improvement for parameter sets derived with symmetric five-layer networks compared with three-layer networks might become obtainable if linear methods instead of neural networks were used for the further data processing.

Figure 5 provides an analysis of the input sensitivities for the calculation of secondary structure with the seven-property representation. This analysis leads to comparable results for all networks with a prediction accuracy better than 60%, so that the network using the complete seven-parameter representation is chosen as an example. All sensitivities for one of the 39 input data blocks are summarized in order to derive information about the influence of each of these input data blocks. The values are normalized to give 1 for the highest sensitivity. As expected, the sensitivity for the central amino acid is the highest. Moreover, a comparably high sensitivity is obtained for the four bordering input blocks on both sides containing averaged information for the surrounding amino acids. Of special interest is the higher influence of amino acids located after the central amino acid in the PDB file. These are the amino acids synthesized after this central amino acid in nature and would therefore suggest that these amino acids have a higher influence on the folding behavior of the central amino acid. A plot of the sensitivities obtained only for the helix probability provides an increased sensitivity for the –8th, –4th, +4th, and +8th amino acids. This result shows the influence of the local primary structure on the formation of hydrogen bonds to form an α-helix, which is also the reason for the higher accuracy in the prediction of α-helices.

Figure 6 gives the sensitivity summarized for the seven- and five-parameter representations, respectively, obtained for the two networks using the complete parameter representations. The dominating influence of the helix and sheet probability in the seven-parameter representation is easy to obtain. The improvement in the prediction obtainable on going from five to seven properties is caused by the two statistically derived values and is therefore also the reason for their high sensitivity. In this case the neural network extracts an essential part of the information not on the basis of amino acid parameters but rather on the basis of the database knowledge. While the amino acid properties only reflect primary structure information, the database averages already contain information about the preferred secondary and tertiary structures of proteins. This additional relevant information has consequently to improve the prediction of secondary structure.

A relative increase of the sensitivity for the volume and the steric parameters is obtained for the five-parameter representation with respect to the seven-parameter representation as compensation for secondary structure probabilities not provided in this case. Both parameters have a high influence on the formation of secondary structure, as is well known. However, the better results obtained for the seven-parameter representation prove that this information cannot be replaced completely.

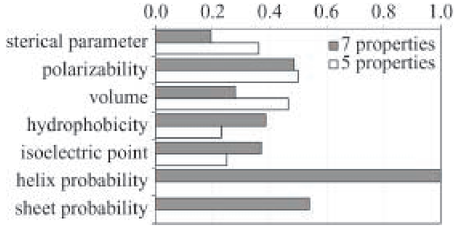

Fig. 6 Input sensitivity for the five or seven parameters out of the complete parameter set obtained from the neural network predicting secondary structure probabilities. The sensitivity is normalized to give 1 as maximum value

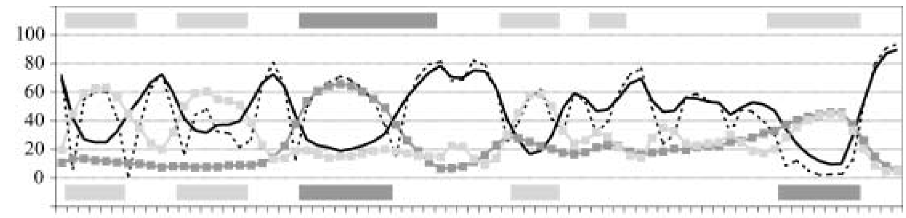

Fig. 7 Secondary structure of ubiquitin is given as obtained from the X-ray structure (*bars above* 100) and as seen from the neural network using the complete seven-parameter representation (*bars below* 0). *Light gray* stands for β-sheet and *dark gray* stands for α-helix. The individual probabilities normalized to give a sum of 100% are plotted. *Light gray squares* are again for β-sheet and *dark gray squares* for α-helix. The coil probability is given by a *black line*. The final network prediction is given by the highest out of these three values. The *dashed black line* is the one minus the fraction of the second highest and the highest probability in percent. This value provides a confidence value, monitoring how safe the judgment of the network is for each individual residue

As widely discussed, errors in secondary structure prediction often occur at the beginning and end of secondary structure elements, so that their length or their position becomes ambiguous. However, most of the secondary structure elements are found. The comparison of the three predicted probabilities for helix, sheet, and coil allows us to decide whether the network is "sure" about its judgment, or whether a second possibility or even all three possibilities are of similarly high probability.

Without going into too much detail, the available information is illustrated on one small protein, ubiquitin, in Fig. 7. Beside the probabilities for helix, sheet, and other (the sum is normalized to be one) the true secondary structure obtained from the X-ray structure as well as the predicted secondary structure are given. The overall prediction of the network is as good as 68% for this protein. The network misses the small β-sheet region 49–51 and converts the β-sheet 65–72 into an α-helix. The other secondary structure elements are found at their correct positions. The α-helix is one period too short and two of the β-sheets are one amino acid too short. However, the black dashed line shows 1 minus the second highest probability divided by the highest probability in %. This value would be 100%, if one of the three types had been predicted with 100% probability and the other with zero, and can reach 0% if two probabilities are the same and higher than the third. We therefore suggest using this value as a confidence measure for secondary structure predictions. As can be seen from Fig. 7, the value is high if changes in secondary structure occur and especially for the wrongly predicted α-helix at the end of the sequence, since the β-sheet probability is close to having the same magnitude. If only the predictions with a confidence value smaller than 50% are considered, 91% of the predicted secondary structure types are correct.

Conclusion

Symmetric neural networks have been implemented successfully to reduce the dimensionality of amino acid parameter sets. The relevant information of these parameter sets is projected into three numbers, which can be used for further data analysis. The use of reduced parameter representations has a general potential to increase the speed of data analysis. In this special case the number of necessary parameters is decreased by more than 50%, representing seven properties by three parameters without losing essential information.

The ability of the reduced parameter representations to provide the complete information is proven by predicting the secondary structure of proteins from their primary structure. The reduced parameter representations with three parameters give comparably good results to the complete parameter representations. The speed of computing the secondary structure is increased about linearly compared to the complete parameter representations by more than 100%. The time necessary for training the network is decreased by a factor of 6.

The approach can be adopted to predict reduced parameter representations for other amino acids and of course also for other structural building blocks, as for example nucleic acids. Moreover, additional or other parameter sets can be used to create reduced parameter representations for the solution of special problems.

Data processing inside a neural network is illustrated by analyzing input sensitivities of the networks. A confidence value for secondary structure prediction, which allows the critical analysis of the network suggestion of secondary structure and might be useful to detect false positives in the predicted secondary structure elements, is suggested.

Supplementary material. The derived reduced parameter representations for amino acids, the program for predicting secondary structure: “Secondary” as well as the program for training and analyzing artificial neural networks “Smart” are available for academic use as supplementary material or from http://www.jens-meiler.de.

Acknowledgements The authors thank B. Coligaev and W. Peti for downloading the subset of the PDB. The authors are especially thankful to Prof. C. Griesinger and Dr. R. Meusinger for useful discussion. J.M. is supported by a Kekulé stipend of the Fonds der Chemischen Industrie.

References

1. Zupan J, Gasteiger J (1993) Neural networks for chemists. VCH, Weinheim
2. Devillers J (1996) Neural networks in QSAR and drug design. Academic Press, London
3. Doucet JP, Panaye A, Feuilleaubois E, Ladd P (1993) J Chem Inf Comput Sci 33:320–324
4. Cherqaoi D, Villemin D (1994) J Chem Soc Faraday Trans 90: 97–102
5. Meusinger R, Moros R (1995) Application of genetic algorithms and neural networks in the analysis of multi-component mixtures using NMR spectroscopy in software. In: Gasteiger J (ed) Entwicklung in der Chemie. Gesellschaft Deutscher Chemiker, Frankfurt am Main, pp 209–216
6. Thomas S, Kleinpeter E (1995) J Prakt Chem/Chem-Z 337: 504–507
7. Svozil D, Kvasnicka V, Pospichal J (1997) Chemom Intell Lab Syst 39:43–62
8. Amendolia SR, Doppiu A, Ganadu ML, Lubinu G (1998) Anal Chem 70:1249–1254
9. Meiler J, Will M, Meusinger R (2000) J Chem Inf Comput Sci 40:1169–1176
10. Rost B, Sander C (1993) J Mol Biol 232:584–99
11. Rost B (1996) Methods Enzymol 266:525–539
12. Salamov AA, Solovyev VV (1997) J Mol Biol 268:31–36
13. Baldi P, Brunak S, Frasconi P, Soda G, Pollastri G (1999) Bioinformatics 15:937–946
14. Chandonia JM, Karplus M (1999) Proteins: Struct Funct Genet 35:293–306
15. Moult J (1999) Curr Opin Biotechnol 10:583–588
16. Petersen TN, Lundegaard C, Nielsen M, Bohr H, Bohr J, Brunak S, Gippert GP, Lund O (2000) Proteins: Struct Funct Genet 41:17–20
17. Livingstone DJ, Hesketh G, Clayworth D (1991) J Mol Graphics 9:115–18
18. Kocjancic R, Zupan J (1997) J Chem Inf Comput Sci 37:985–989
19. Livingstone DJ (1996) In: Devillers J (ed) Multivariate data display using neural networks in neural networks in QSAR and drug design. Academic Press, London
20. Fauchere JL, Charton M, Kier LB, Verloop A, Pliska V (1988) Int J Peptide Protein 32:269–278
21. Holm L, Sander C (1996) Science 273:595–602
22. Meiler J (2001) www.jens-meiler.de
23. Rost B, Sander C (1993) Proc Natl Acad Sci USA 90:7558–7562