Biol. Cybern. 76, 441–450 (1997)

Biological Cybernetics

Classification of protein families and detection of the determinant residues with an improved self-organizing map

Miguel A. Andrade [1,*], **Georg Casari** [2], **Chris Sander** [2,*], **Alfonso Valencia** [1]

[1] Protein Design Group, Centro Nacional de Biotecnología-CSIC, Cantoblanco, E-28049 Madrid, Spain
[2] Protein Design Group, EMBL, Meyerhofstrasse 1, D-69120 Heidelberg, Germany

Received: 25 July 1996 / Accepted in revised form: 13 February 1997

Abstract. Using a SOM (self-organizing map) we can classify sequences within a protein family into subgroups that generally correspond to biological subcategories. These maps tend to show sequence similarity as proximity in the map. Combining maps generated at different levels of resolution, the structure of relations in protein families can be captured that could not otherwise be represented in a single map. The underlying representation of maps enables us to retrieve characteristic sequence patterns for individual subgroups of sequences. Such patterns tend to correspond to functionally important regions. We present a modified SOM algorithm that includes a convergence test that dynamically controls the learning parameters to adapt them to the learning set instead of being fixed and externally optimized by trial and error. Given the variability of protein family size and distribution, the addition of this feature is necessary. The method is successfully tested with a number of families. The *rab* family of small GTPases is used to illustrate the performance of the method.

1 Introduction

Protein families contain a wealth of information accumulated during the process of evolution (Zuckerkandl and Pauling 1965). This information is of obvious interest in many different fields of biology, from molecular evolution to protein engineering.

Protein structure prediction methods have clearly benefited from the use of family information (Rost and Sander 1994), including secondary structure prediction (Barton 1992; Benner 1992; Rost and Sander 1993), prediction of contacting residues from correlated mutations (Göbel et al. 1994), threading (Ouzounis et al. 1993) and sequence profiles (Blundell 1992). However, more information can be extracted from a distribution of proteins within a family, namely, the key residues responsible for the biological properties of the family and of the subfamily specifity.

All these methods incorporate sequence information in a very simplistic manner, regardless of the structure of the protein family. From biology we know that during evolution protein families have been structured into subfamilies and groups.

The internal organization of protein families is one of the most striking aspects of multiple sequence alignments. The relationship between proteins belonging to the same family can be described in terms of protein functions with sub-families and groups (paralogous sequences), e.g., the *ras*, *rab* and *rho* subfamilies of the *ras–p21* family, and in terms of the differences between the same protein in different species (orthologous sequences). This classification has been taken into account in the field of molecular evolution but it is not used in other fields and it is not incorporated in prediction methods based on family alignments.

New tools for the analysis of protein families could help in extracting more information from multiple sequence alignments. New representations of family relations will bring to our attention aspects of the problem overlooked before, such as the determination of key regions of the sequence for differentiation within a family and their relationship to functional specificity.

Casari et al. (1995) have used a rigorous vector analysis treatment to provide a visual representation that can be helpful in analyzing clear cases. A more simplistic approach was developed later by Lichtarge et al. (1996) with a similar aim. We propose to apply a different strategy using a clustering algorithm – a self-organizing map (Kohonen 1982) – for the coded sequences.

Correspondence to: M. A. Andrade; Tel: + 44 (0)1223-494450, Fax: + 44 (0)1223-494471, e-mail: andrade@ebi.ac.uk
**Present address:* European Bioinformatics Institute, Hinxton, Cambridge CBIO 1SD, UK

2 The algorithm

2.1 Sequence encoding

The starting point of the method is a multiple alignment of N protein sequences. As in any other analysis methods the quality of the results depends largely on the quality of the input data. In this case, wrong alignments will lead to wrong conclusions. In the usual process of aligning the sequences, gaps are introduced to align equivalent regions of the sequences from one protein to the other. These gaps are treated as positions in the alignment and subsequently all the sequences have the same length.

Each aligned sequence is binary coded as one sequence vector (denoted as F_k, where $k = 1, \ldots, N$): each position of the sequence is descdribed by 20 components corresponding to all possible amino acids (Casari et al. 1995). A 'one' is placed in the component corresponding to the amino acid type present at this position, and the rest of the components are set to zero (Fig. 1a). If any of the sequences contains a gap, the 20 components are set to zero. This is equivalent to excluding positions with gaps, a common practice in other methods for the analysis of multiple sequence alignments. The resulting sequence vector has a length of $20 \times L$ where L is the length of the sequence alignment. The N resulting sequence vectors are used to train the network.

2.2 The clustering algorithm

2.2.1 Architecture of the map. The self-organizing map (SOM) is a two-dimensional layer of $n \times m$ slots arranged in a square topology (in general n and m can take any integer value, but values ranging from 1 to 10 will be used). There is one slot vector for each of the slots. The slot vectors are denoted by $W_{ij}(t)$, where the pair of sub-indexes are the coordinates of the slot with $i = 1, \ldots, n$ and $j = 1, \ldots, m$ (Fig. 1b). These vectors have the same number of components as the sequence

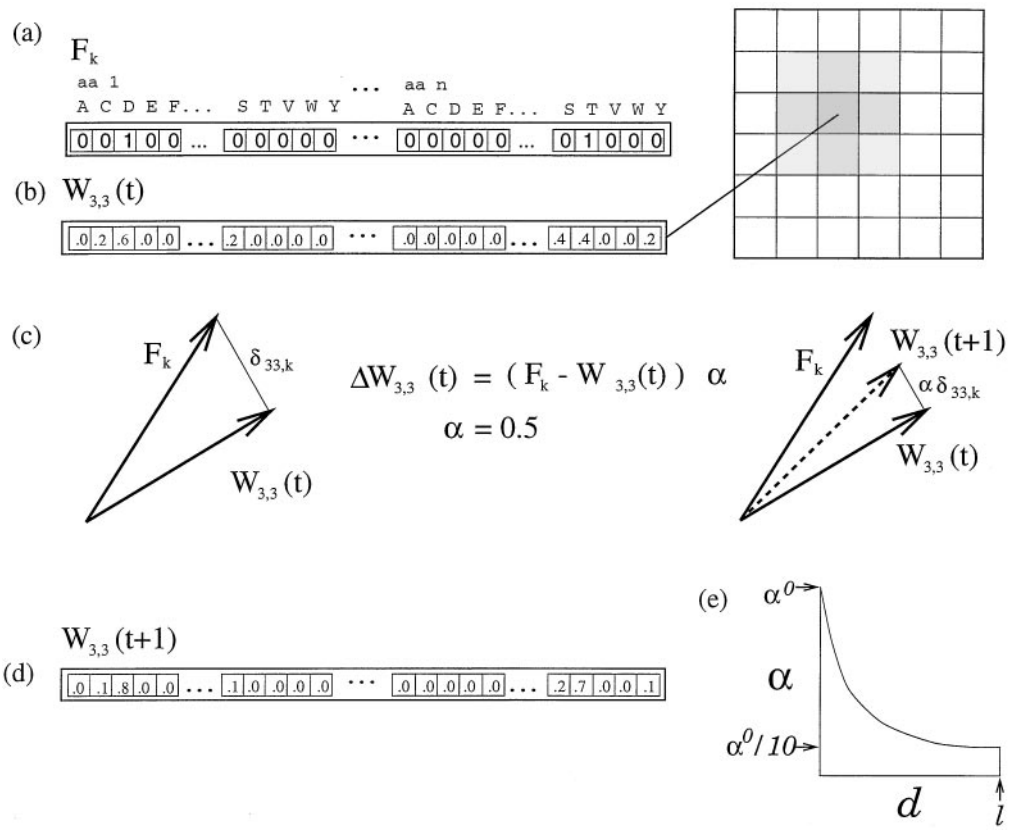

Fig. 1a–e. Training algorithm. **a.** Vector coding of sequences. Sequences are coded in the form of linear vectors of binary values (sequence vectors). The length of the vectors is equivalent to the 20 possible amino acids at each position times the length of the alignment. Here we show a sequence with D at position 1 and T at position n. **b** Self-organizing map. The map is represented as a square lattice. Each lattice position or slot holds a vector with the length of the sequence vectors. The components of these slot vectors are real values between 0 and 1. **c** Training. During the training of the map the sequence vectors are presented one after the other to each slot vector in a random order. A linear combination of the sequence vector and the closest slot vector is taken as the new slot vector. A weighting factor is used in the linear combination to control the speed of the training. **d** Resulting slot vector after a first iteration of the training process. In this example the closest slot vector to the sequence vector F_k is $W_{3,3}$. It contains real values, i.e., 0.6 in the third component. $W_{3,3}$ is updated according to the value of the sequence vector using an α-weighting factor of 0.5 and becomes closer to F_k. In the example, the third component of the slot vector changes from 0.6 to 0.8 (closer to 1) and the second component from 0.2 to 0.1 (closer to 0). **e** Information spread. To assure a soft transition between neighbor slots in the map a smoothing procedure is used. The updating procedure is applied not only to one single slot vector but also to its neighbor slots, represented in the lattice by the gray region. The neighbor region is defined by a decreasing function in such a way that a full α-weighting factor [as defined in (3)] is applied to the central slot and only partially to more distant slots

vectors. Contrary to the sequence vectors, the components of the slot vectors may take real values between zero and one.

2.2.2 Training of the network. The training of the SOM consists of updating the slot vectors according to the sequence vectors. The process of clustering implies that the slot vectors converge to values that are the consensus of a subset of the sequences. There are multiple solutions for partitioning a family in clusters. To find the optimal solution a training procedure with different stages is followed.

2.2.2.1 Initial conditions. At zero time, the slot vectors are arbitrarily set to values that are the mean of all sequence vectors.

2.2.2.2 Training action. A single training action is the update of the slot vectors according to one sequence vector.

Selection of the winning slot. The Euclidean distance ($\delta_{ij,k}$) from a slot vector W_{ij} to the presented sequence vector F_k is calculated in the following way:

$$\delta_{ij,k} = \sqrt{\sum_x | F_{k,x} - W_{ij,x}|^2} \tag{1}$$

where $x = 1, \ldots, 20 \times L$ is the index of the vector components. The most similar slot vector (winning slot vector) is identified by having the smallest distance to the sequence vector. This winning slot vector is updated with a linear combination of its previous value with the presented sequence vector in the following way (Fig. 1c):

$$W_{ij}(t+1) = (1 - \alpha^0) W_{ij}(t) + \alpha^0 F_k \tag{2}$$

where α^0 is a factor that sets the weight given to the example sequence in the updating step. The update makes the slot vector move closer to the example shown to the network (Fig. 1d).

In usual SOM implementations (e.g., Ferrán and Ferrara 1991; Andrade et al. 1993) a time-decreasing α^0 is used. In doing so, the rate of decrement must be finely tuned since when α^0 is zero the updating of the slot vectors stops [see (2)]. In such a case the convergence is imposed by α^0 and not by the data set. Instead, we have chosen a constant α^0 value ($0 \leqslant \alpha^0 \leqslant 1$).

The examples are presented to the system in random order, once for each training cycle. Then, the time devoted to one cycle is tN, where N is the number of examples. The random presentation of the examples adds noise to the dynamics of the slot vector evolution, which helps the system to avoid nonoptimal classifications.

Updating of the neighborhood. In order to obtain a map, the slot vectors in the neighborhood of the winning slot are also updated. The updated slot defines the center of the updating region (the gray region in Fig. 1b). We have used a square-shaped region of variable size. The strength of updating in the neighboring slots decreases with the distance from the winning slot, in our case following an exponential dependence on the Euclidean distance (d) from the slot to be updated to the central slot:

$$\alpha(d) = \alpha^0 \exp(-d/f) \tag{3}$$

where α^0 (the updating strength on the winning slot) is a constant between 0 and 1, and

$$d = \sqrt{(i - i')^2 + (j - j')^2} \tag{4}$$

where (i, j) are the coordinates of the winning slot and (i', j') are the coordinates of the slot to be updated. The function is normalized using a factor (f) chosen such that $\alpha(l) = 0.1\alpha^\circ$, where $(2l + 1)^2$ is the area of the neighborhood (Fig. 1e). Then, the update of the neurons of the updating region takes place following:

$$W_{ij}(t+1) = [1 - \alpha(d)] W_{ij}(t) + \alpha(d) F_k \tag{5}$$

2.2.2.3 Training cycle. During the training procedure, the slot vectors must find a compromise between the many sequence vectors presented. At the beginning of the training an extended neighborhood is used, and the vectors acquire values closer to the mean of the set of training vectors. Later, the neighborhood is gradually decreased and regions of the network that only map part of the example set appear. At the end of the training the neighborhood is composed of only one slot and then each of the sequence vectors alters only one slot vector.

In previous work using SOMs, decreasing the neighborhood size has usually been done using a constant decreasing rate, from a neighborhood that covers all the network at the beginning of the training to a region of only one slot at the end. The rate decrease has to be fixed by trial and error and is usually kept constant (e.g., Andrade et al. 1993; Hanke et al. 1996). However, the optimal rate for map convergence is dependent on the set of examples. Given the heterogeneity of protein families it is desirable to find a more general procedure.

Rate of decrease of the updating area. We have implemented a new procedure that decreases the size of the neighborhood in steps along the training procedure. The rate of shrinkage is coupled to the degree of convergence of the slot vectors during the training. For a vector slot at position i, j (W_{ij}), the dispersion (s_{ij}) along a number of training steps (μ) is computed:

$$s_{ij} = \sqrt{\sum_{t=(k-\mu)N}^{kN} \frac{|\bar{W}_{ij} - W_{ij}(t)|^2}{\mu}} \tag{6}$$

where t runs over the μ previous training steps considered for the calculation, and $\bar{W}_{ij}$ is the mean of the W_{ij} values over the μ training steps.

In order to consider the behavior of the whole map, the mean over the dispersion values of all the slots

$$\bar{s} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{s_{ij}}{nm}} \tag{7}$$

is monitored. When the rate of change of $\bar{s}$ falls under a given threshold (t_s),

$$\frac{|\bar{s}(t) - \bar{s}(t-1)|}{\bar{s}(t)} < t_s \tag{8}$$

convergence is assumed and the updating area is decreased. Following this, a new training cycle with the smaller neighborhood begins. The last training cycle has

an updating neighborhood of only one slot. An example is shown in Fig. 2.

2.3 Quality of the final classification

After the training procedure, the slot vectors are an interpolation between the set of examples. Due to the neighbor relationships induced by the use of an updating region, the distribution of slot vectors may resemble a map of the family, i.e. adjoining slots tend to have similar sequences. However, the distribution of the proteins of a family in sequence space have, a in general, a dimensionality higher than two which can not be completely represented in a two-dimensional map. Therefore it is not possible to use the topology of the map for the protein classification, but instead the clustering properties can be used. Hence, the map is interpreted as a classification by assigning each example sequence to the slot that has the closest slot vector to it. One slot can hold many examples, one, or none of them. For each sequence, the distance between its sequence vector and its closest slot vector is an indication of how well an example is represented in the network.

As the system explores a very complex high-dimensional space, and since the training procedure is affected by the random presentation of the examples, different runs may give different final states, especially if the SOM slots do not adequately cover the family distribution.

For example, if a protein family is distributed in three clusters in sequence space but is classified using a two-slot SOM, some of the examples will necessarily be poorly classified. This can be seen by repeating the experiment with different initial conditions and finding that some of the examples go to different clusters in different experiments.

These ambiguities can be resolved by considering the distance of the sequences in each cluster from the mean of the cluster. For a subgroup Ω_{ij} of S sequences, the mean sequence vector ($\bar{F}_{ij}$) can be calculated. The the measure

$$D_{ij} = \sqrt{\sum_{k \in \Omega_{ij}} |\bar{F}_{ij} - F_k|} \tag{9}$$

can be used as an indication of the quality of a single cluster (by definition, D_{ij} for slots without any assigned example is zero), and

$$Q = \sum_{i,j} D_{ij} \tag{10}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, m$ will assess the quality of a whole cluster distribution. Thus, the clustering capacity of the map can be tested and different runs of the method can be compared.

2.4 Tree construction

By means of a single SOM training we can obtain the clustering of the family at a definite resolution level.

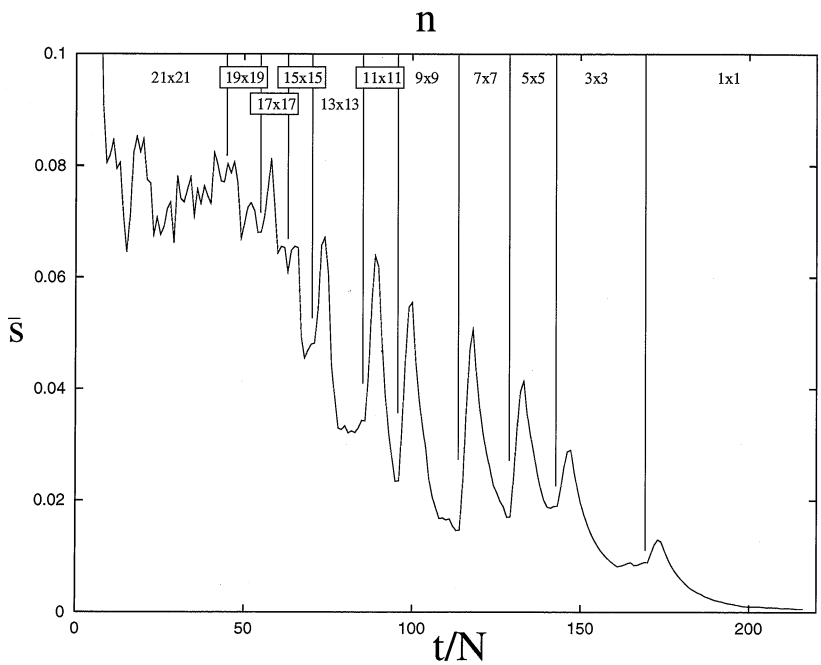

Fig. 2. Evolution of the weights. Evolution of the convergence parameter ($\bar{s}$) along the training procedure ($t/N = 250$ training cycles; where t is time and N is the number of proteins used for the training) for a 10×10 SOM. The numbers in the upper part of the figure indicate the size of the neighborhood ($n \times n$). Neighborhoods extending out of the limits of the network are truncated in the borders of the network. A small variation in $\bar{s}$ indicates that no further convergence can be obtained. When the rate of change of $\bar{s}$ from one training run to the next falls below a given threshold ($t_s = 0.005$) the neighborhood is shrunk. These points are indicated by *vertical lines* on the plot. A value of $\alpha^0 = 0.1$ was used for this and the following network trainings as it gives a smooth weight evolution. Larger values of α^0 give higher values of $\bar{s}$, and smaller values require longer training

However, several SOMs with several resolutions can show a sequence relationship not present in a single map, e.g., the hierarchy of sequences in the family. This can be extracted from a set of experiments with different map sizes that can be arranged in a tree-like fashion. The residues responsible for the family branching can be determined.

The tree is constructed by linking the clusters that contain the same sequence at successive levels (1, 2, . . . , i, . . . , n). With good classifications, we expect to see clusters splitting into subclusters as we increase the number of slots. However, the opposite situation occurs in a few cases, namely two sequences previously classified in different slots collapse into the same slot when a map with more slots is used.

In the case of such a collapse, sequence A is classified in a different slot from B at level i but it is classified in the same slot at level $i + 1$. In order to present a final tree we solve these inconsistencies and annotate the position in the tree at which they are found. The number and position of these collapses is used as a criterion regarding the degree of reliability of the classification. The conflict produced by a collapse is solved either by merging A and B at level i or by separating them at level $i + 1$.

To decide between these alternative clustering schemes we proceed in the following way: first, we find which of the clusters containing A or B at level i has the larger number of sequences; for example let us suppose that it is the one containing A. This cluster is then compared with the cluster that contains the A and B sequences at level $i + 1$, comparing the distances of the sequences of the cluster with its corresponding slot vector (smaller distances corresponding to a better cluster). If the cluster at level i containing A is better than the one containing A and B at level $i + 1$, then the first cluster is kept and the one at level $i + 1$ is split by taking apart B and putting it into a new cluster. Conversely, if the cluster at level $i + 1$ containing A + B is better, this one is kept and B is joined to the cluster containing A at level i.

In some cases a sequence is difficult to clasify for reasons intrinsic to the geometry of the family distribution. We propose here to detect this effect by counting the number of collapses occurring during the tree construction for each sequence.

3 Results

3.1 Branching scaling

Different criteria can be used in describing the clusters of a family. Here we have presented a proper clustering algorithm that uses plain sequence similarity to gather the sequences. However, other classification algorithms can be used for the same purpose and different results may be obtained.

The tree representation of the map information can be compared with phylogenetic trees that try to accommodate the evolutionary relationships of a group of sequences in a tree according to their sequence homology. A phylogenetic tree can be used to describe the clusters of a family by taking the groups of sequences under the branches maintained at a certain distance from the root. However, in using this method the cluster distribution is extremely dependent on where the root of the tree is positioned. In general, it is observed that, reflecting the evolutionary distances of the sequences, some subfamilies expand at shorter distances from the tree root than others. This different timing in branch expansion can be convenient for the study of the evolutionary properties of the family, but is not appropriate for the description of the sequence positions that characterize the subfamilies of a family.

Our method introduces a different branch scaling: it is based on sequence identity and depends on cluster population, in contrast to the phylogenetic tree branch scale that is based on sequence homology and which does not depend on cluster population. The effect is that the map may produce a less realistic (from the evolutionary point of view) but more homogeneous distribution of clusters. No matter what the evolutionary constraints are, the map tends to make clusters of similar size, grouping sequences that are distantly related and resolving very populated subfamilies into several clusters.

Figure 3 compares the distribution of clusters of the *ras–p21* family obtained with a 10×10 map with the equivalent distribution obtained with a standard neighbor-joining tree, CL [Clustalw] (Higgins and Sharp 1988). In the phylogenetic tree the distribution of clusters is more heterogeneous because some of the subfamilies expand at shorter distances from the root than others. The SOM picks most of the subfamilies at an intermediate level of expansion.

In Table 1 we show the comparison for five families (lactin dehydrogenases, a subset of the serine-threonine kinases, *ras*, serine proteases and ADP-ribosylation factors). The SOM trees display almost the same groups as

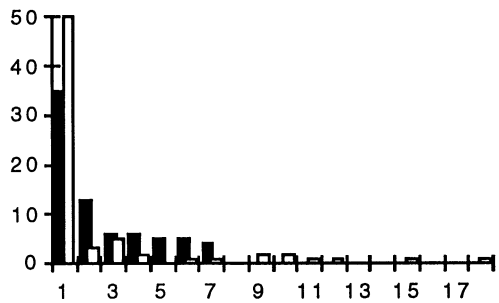

Fig. 3. Distribution of clusters depending on their size. In this histogram we compare the distribution of cluster size in our algorithm to that obtained using a neighbor-joining phylogenetic tree obtained from Clustalw (CL). The *horizontal axis* values are the number of sequences in a cluster; *vertical axis* values are the number of clusters with a particular number of sequences. *Empty bars* represent the distribution for the tree obtained from SOM classifications; *black bars* represent that generated by CL. The example is taken from the analysis of the *ras–p21* family. The clusters for the tree are taken from the 10×10 SOM. At this level there are 72 clusters. The clusters for the CL tree are obtained by cutting the tree at a given distance from the root so that the number of branches under this distance is 72. The SOM produces clusters of intermediate size with fewer very small or very big clusters

Table 1. Comparison between CL (Higgins and Sharp 1988) and self-organizing Map (SOM) trees

Family	Sequences	Length	SOM map size	Groups	CL groups	CL sequences	Collapse index	Branches < 90
8ldh	40	329	6×6	21	100%	100%	0.025	37%
2cpk	79	336	10×10	41	100%	100%	0.025	50%
5p21	184	166	10×10	72	92%	91%	0.053	82%
4ptp	200	223	10×10	75	92%	93%	0.106	84%
arf	18	183	4×4	10	100%	100%	0.125	56%

Five examples are given: 8ldh, lactic dehydrogenases; 2cpk, a subset of the serine-threonine kinases; 5p21, *ras–p21* family; 4ptp, serine proteases; arf, ADP ribosylation factors. Family, the hssp identifier of the family (Schneider and Sander 1991); Sequences, the number of aligned sequences; Length, the length of the alignment; SOM map size, the size of the neural network used for the classification of the sequences; Groups, the number of groups obtained with the SOM; CL groups, percentage of these groups that can be found in the tree obtained with CL at least with a 50% of coincidence of sequences; CL sequences, percentage of sequences classified in those classes by the SOM; Collapse index, number of collapses divided by number of sequences and by the number of levels obtained in making a tree using SOMs with size $2 \times 1, 3 \times 1, 2 \times 2, \ldots, n \times n$ (this number indicates how difficult is to fit the family to the tree); Branches < 90, branches of the CL tree under the selected groups that had a bootstrapping value (Brown 1994) smaller than 90% indicating the unstable branches (this quantity is equivalent to that reflected by the previous column for the SOM)

those obtained by using CL to classify the same family (although at different distances from the root due to the difference in branch scaling).

3.2. *Stability and consistency of clusters*

Trees are not always a good representation of family relationships. There are uncertainties that are classically measured in phylogenetic trees by the bootstrap values (Brown 1994). In the trees obtained from the SOMs the same problem exists. The position and number of the collapses is an indication of the reliability of the tree.

Table 1 shows that there is a good agreement in the reliability of the classification as measured by bootstrap experiments in the phylogenetic tree obtained from CL and by the number of collapses in the tree constructed from SOMs.

3.3 *Values of the slot vectors*

The slot vector components are real values corresponding to a linear interpolation of the corresponding components of the closer sequences. These values are set along the training procedure in a way that better represents the diversity and specificity of sequences of the family. They can be seen as a consensus of the different sequences mapped on the SOM.

The comparison of the values of the different slot vectors of the same level of the SOM shows how some positions have more information about the classification than others. The components of the slot vectors indicate the level of conservation of one amino acid at a particular position of the alignment in a group of sequences. The differences in the values of the other slot vectors are responsible for a given classification.

In Fig. 4 some representative values for two of these slot vectors are displayed. Completely conserved positions can be clearly distinguished (position 36 in the figure). Position 34 is also interesting as the position is almost completely conserved in both slots but the conserved residue is different, i.e., *T* vs *V*. This position will have a strong influence in the classification of the different groups of the family.

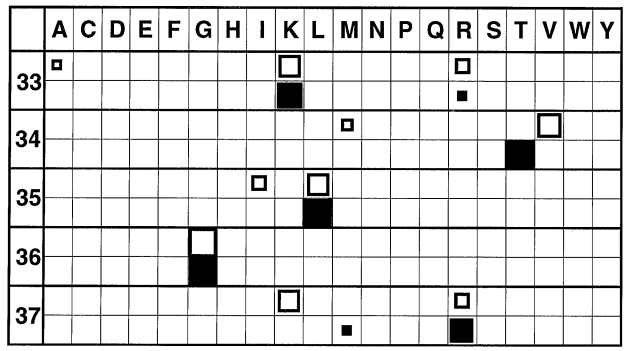

Fig. 4. View of the slot vectors. The comparison of the vector components of different slot vectors allows precise description of the degree of sequence conservation in protein families. An example to illustrate this point is shown by considering the alignment positions 33–37 for slots $\{1, 1\}$ and $\{2, 1\}$ of a 2×2 SOM trained with a subset of sequences from the serine-threonine protein kinase family. $\{1, 1\}$ clusters the cAMP-dependent kinases and $\{2, 1\}$ the Ca^{2+}-dependent kinases. *Horizontal axes* indicate the type of amino acid and *vertical axes* the position in the alignment. The size of the *squares* represents the value of the amino acid at the alignment position (ranging from 0, amino acid not present at the position, to 1, amino acid present in all sequences at that position). The *white squares* represent the values of the slot vector $\{1, 1\}$ and the *black squares* those of $\{2, 1\}$. Four different situations are described in this example: **a** The trivial case occurs at position 36 where G is conserved in the whole serine/threonine kinase family. Accordingly, this slot vector entry for both neurons is one. **b** At position 33 none of the clusters has a conserved amino acid. However, K is the most likely amino acid at this position as indicated by the proximity to one entry in both vectors. Note, however, that other amino acids are also possible (R, A). **c** At position 35 both clusters present the same conserved residue (L) and thus the entry is one or close to one. But this position is not conserved in the whole family, thus constituting a specific characteristic of these two clusters ($\{1, 1\}$ and $\{2, 1\}$). **d** At positions 34 and 37 $\{1, 1\}$ has a conserved amino acid different from $\{2, 1\}$. Thus, positions 34 and 37 are key in the separation of the Ca^{2+}-dependent kinases from the cAMP-dependent kinases

3.4 *An example*

As a full example of the method's performance we present the analysis of the *rab* family (42 proteins). The evolution of the SOM is monitored following the region of variation for each of the slots. In an ideal case, the variation

of the dispersion values ($\bar{s}$) should exponentially approach a final value that depends on the map topology, the training procedure and the set of examples. The behavior of the SOM approximates to this ideal case when the neighborhood relationships are of short range (Fig. 2).

After the training procedure, each sample protein is assigned to the slot of the map with the closest slot vector (Fig. 5 shows an example for a 5×5 map). A map with areas of different density is obtained through the assignments of the proteins (Fig. 5a). The position of the different sequences resembles the phylogenetic relationships in

(a)

	1	2	3	4	5
1	RYH1_SCHPO RAB6_HUMAN RAB6_CAEEL		RAB5_TOBAC RHN1_NICPL RHA1_ARATH	YPT5_SCHPO	YP51_YEAST YP52_YEAST YP53_YEAST RABB_DICDI
2	ARA4_ARATH ARA1_ARATH		RB5A_HUMAN RB5A_CANFA RB5B_HUMAN		RB17_MOUSE
3	YPT3_SCHPO RAO3_DISOM RB11_HUMAN ARA2_ARATH RB11_DICDI	RPG1_ORYSA			RB18_LYMST YPT3_VOLCA
4					YPT5_VOLCA
5	RAB2_DICDI YPT4_VOLCA RAB2_LYMST RAB2_HUMAN RAB2_RABIT RAB2_RAT	RB14_RAT RAB4_DICDI	RAB4_RAT RB4B_CANFA RAB4_HUMAN		RAB7_PEA RAB7_DICDI RAB7_RAT RAB7_CANFA

(b)

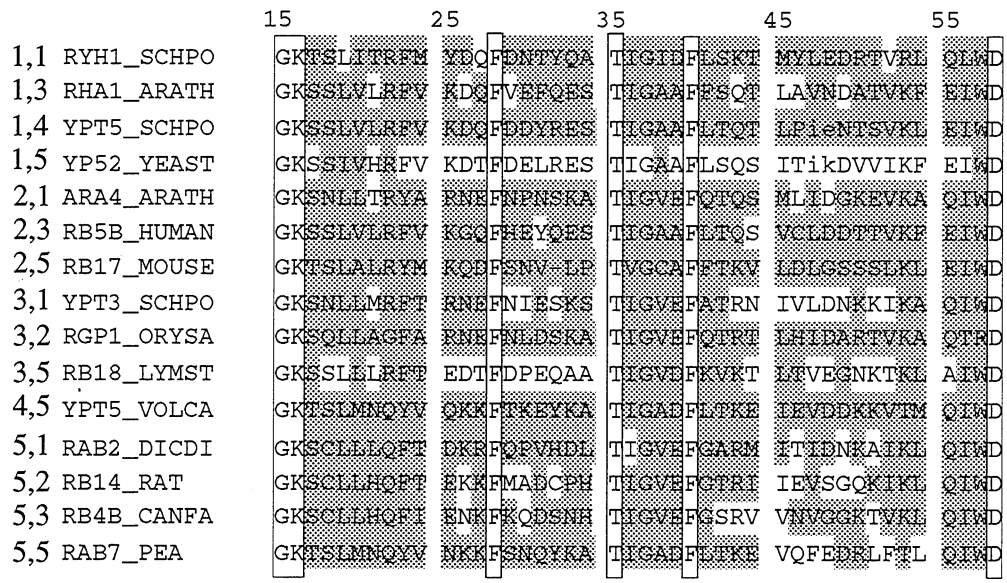

Fig. 5a, b. Example of 5×5 map classification and some representative sequences. **a** Forty-two sequences of the *rab* subfamily of small GTPases (Valencia et al. 1991) are classified in a 5×5 SOM. The sequence closest to the corresponding slot vector is shown in each slot (using its SwissProt identifier code). The classification obtained is in good agreement with that obtained by other methods (CL, and results not shown). **b** A fragment of the alignment (positions 15–58) of a set of representative sequences, one per slot is shown. *Empty boxes* indicate those amino acids conserved in the whole 42 sequences. *Gray background* indicates those amino acids conserved in the sequences classified in the slot but not in the whole *rab* family. The mapping of the sequences in the network corresponds to the differences found between the sequences. Closer sequences tend to be mapped either in the same slot or in neighboring ones. For example, the sequences classified in the neighboring slots $\{5, 1\}$, $\{5, 2\}$ and $\{5, 3\}$ have a C at position 18 in the alignment whereas $\{2, 1\}$ and $\{3, 1\}$ have an N, the sequence classified at $\{3, 2\}$ has a Q and the rest has an S (except TPT3_VOLCA at $\{3, 5\}$)

the family. Groups of similar sequences tend to stay in either the same or adjacent slots (Fig. 5b).

Though the family distribution in a highly multidimensional sequence space cannot be completely maintained in the two-dimensional map, the results of a whole set of experiments represented in the tree of the *rab* family (Fig. 6) can hold this information. The tree is a combination of independent experiments with SOMs of different size. The clusters obtained in the different experiments are linked as they share the same sequences, i.e., slot 14 in

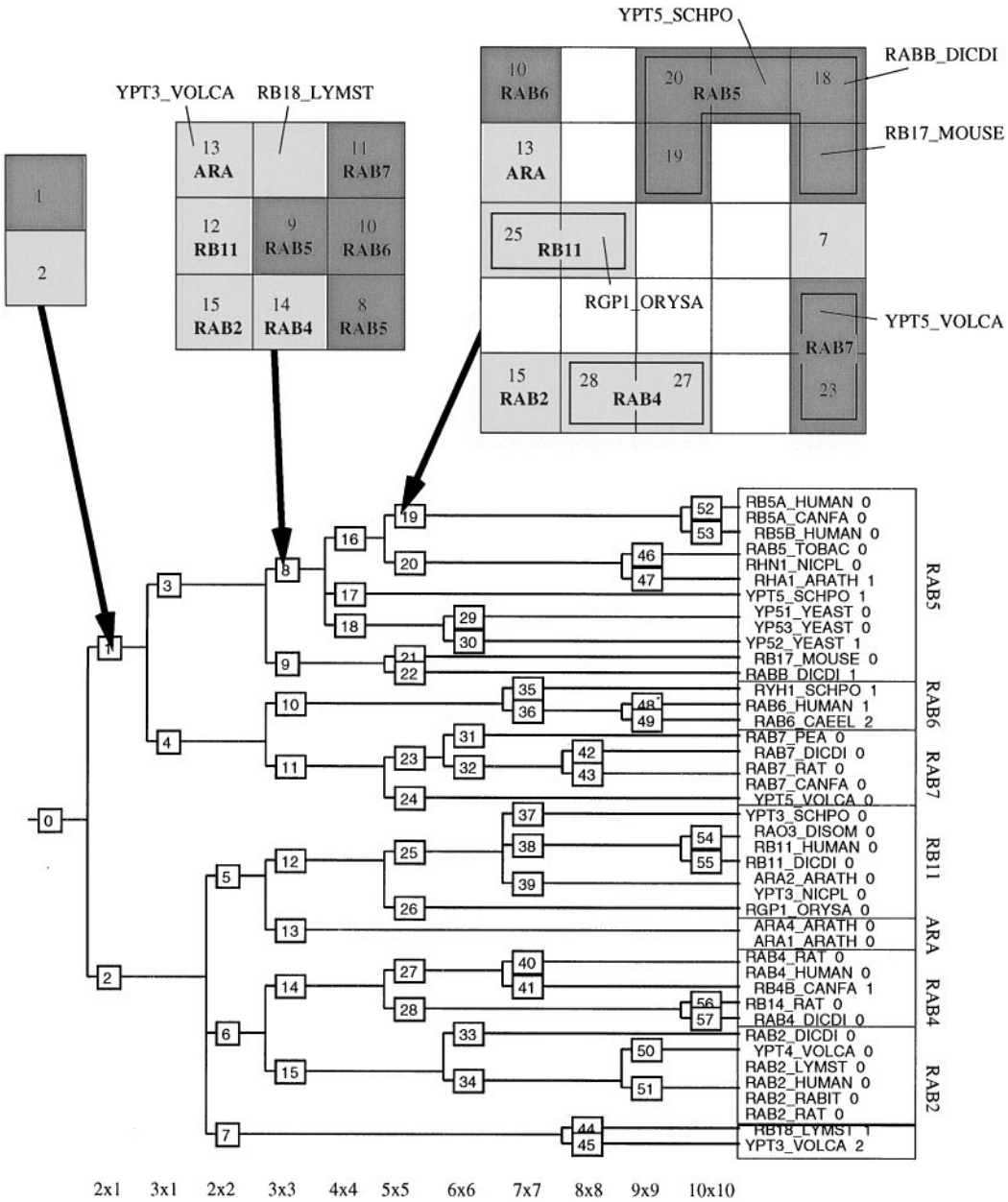

Fig. 6. Clustering of sequences of the *rab* subfamily. The *upper part* of the figure represents the final classification in the 1×2, 3×3 and 5×5 neural networks of the 42 sequences of the *rab* subfamily used in the example shown in Fig. 5. The position in the map of some characteristic sequences is indicated as well as some biologically meaningful groups of sequences. *Light* and *dark grey* indicate the sequences that were classified together in the 1×2 map. The *bold lines* extending over several slots in the 5×5 map indicate which sequences were classified together in the 3×3 map. In the *lower part* of the figure, the results of individual experiments with each of the different neural networks are connected in a tree form. The connections are established joining each cluster to its subclusters at the next-highest level. Possible conflicts between independent clustering experiments done at different resolution levels are solved as described in Sect. 2. The nodes are numbered from left to right. The same numbers are used to label the slot in the neural networks shown in the upper part. *Bottom line* indicates the size of the SOM used for each resolution level. The number after each sequence name indicates the number of incoherences found in the clustering of the sequence through the tree construction. These numbers are inversely proportional to the reliability of the classification. Most sequences are correctly classified without problems and only ten sequences present some conflict because they were ambiguously classified in some step of the procedure

the 3×3 map contains the same sequence as slots 27 and 28 in the 5×5 map.

It is interesting to follow some selected examples to understand how the SOM works. For example, YPT5_VOLCA and the *rab7* subfamily are classified together at low levels (e.g., in the 2×1 map or in the 3×3 map) and split later (5×5 map; Fig. 6), though they remain close in the map (Fig. 5a). If we look at the sequence information (or at the corresponding slot vectors) we can find that YPT5_VOLCA and the *rab7* proteins share a common motif, which is not present in the rest of the family, from position 19 to position 27 (Fig. 5b).

Low-resolution classifications could force spurious clusters that are not supported by sequence similarities. The collapses help to detect this problem. For example, the *rab6* and *rab7* subfamilies are classified in separate clusters in the 3×1 map and in the same cluster in the 2×2 map. During the procedure of tree construction, such a collapse is solved by separating *rab6* and *rab7* subfamilies at the 3×1 map. The *rab7* subfamily is the most distant group of the *rab* family and it has sequence characteristics that make its similarity with other groups unreliable in statistical terms (Valencia et al. 1991). Therefore, it is not surprising that the SOMs find problems in its classification.

Collapses can also be found at high resolution levels. For example, two members of the *rab6* subfamily, RAB6_CAEEL and RAB6_HUMAN, are clustered together in the 7×7 and 8×8 maps, separated in the 9×9 map and, finally, clustered together in the 10×10 map. The collapse is solved by separating them at the last level.

Another example shows the problems that may arise with groups that are not clearly defined. YPT3_VOLCA and RAB6_CAEEL share the same effector domain, a typical sequence element of the *rab* family (TIGVDFK at positions 41–47). Accordingly, they are classified as a group in medium/high-resolution maps. However, at low resolution they go to different clusters: YPT3_VOLCA with the *ara* subfamily (2×2 and 3×3 maps) and RB18_LYMST with the *rab5* subfamily (2×2 maps). In the construction of the tree, the assignment of these sequences to either *rab5* or *ara* subfamilies scores lower than the formation of a separate subfamily with both, and thus this latter option is automatically chosen by the clearing algorithm for the 2×2 and 3×3 levels of the tree (Fig. 6).

This example shows the power of the SOM to obtain a reliable classification that agrees with the classifications obtained by phylogenetic trees. A problem for all the methods based on tree representation is that in certain cases the non-continuity of the evolutionary process makes it difficult to fit the protein family topology into a tree topology. However, by using the slot vector values it is always possible to deduce the discriminant residues for the subfamily segregation. They define the protein regions that are usually responsible for the discrimination of the protein function and thus of biological interest.

4 Discussion

In this work we have used a SOM to classify different protein families. To make the training process more independent of the training set we have introduced a new feature in the SOM: an adaptive rate of neighbor decreasing size dependent on map convergence. This makes it possible to set a fixed learning rate (α^0). The result is a considerable saving of time for the user since no optimization process of the SOM learning parameters is needed.

SOMs have previously been used in other fields of sequence analysis. In particular they have been shown to be useful for identification of uncommon DNA regions from the coding region of the human insulin receptor gene (Arrigo et al. 1991), for classification of proteins both from circular dichroism spectra (Andrade et al. 1993) or from dipeptide composition (Ferrán and Ferrara 1991; Ferrán et al. 1994) and for motif recognition (Hanke et al. 1996). SOMs have shown their power of discrimination in these cases even when little information is used as input. The goal, scope and method used here are different from these other approaches: (a) we deal with sequences that belong to a given protein family and that can be further classified into subfamilies, (b) the coding of proteins is not done by their composition but by their complete sequence, and therefore the full information contained in the multiple sequence alignment is used, (c) the aim is not to classify new sequences in their corresponding families but to analyze protein families (multiple sequence alignments), joining several maps with a tree-like representation and, more importantly, to pinpoint the key residues for this classification (tree determinant residues).

The clustering abilities of the improved SOM were successfully tested, comparing the groups obtained with those from phylogenetic trees (Table 1). SOMs add a number of interesting qualities to a phylogenetic classification:

1. The reliability of the classification can be assessed by the number of ambiguities found during the classification, here reflected in collapse of sequences between different levels of the SOM. This criterion is completely different from the simulations normally used to evaluate the quality of classifications in phylogenetic trees (bootstrapping experiments; Brown 1994). These values can be used to assess the quality of the final classification.
2. The classification depends more on the quality of the examples and on their distribution than on their quantitative differences, i.e., a subfamily with many sequences tends to expand more than a subfamily with a few sequences. Subfamilies with more examples are more finely resolved.
3. The slot vectors represent the sequence profile of the closer sequences. Here we present a way of using them to detect the positions in a multiple sequence alignment having more information for the classification of the sequences (tree determinants). This information can easily be obtained from the direct comparison of the slot vector values and in biological terms can be used for the localization of active sites or binding regions (Casari et al. 1995).

All these features make classification of protein sequences with a SOM a method for studying protein families from a new perspective that can bring light to the field of protein functional analysis and molecular evolution. Future work includes the analysis of interesting protein families where the visualization of the different levels of conservation of residues may help an understanding of functional specificity within a family.

References

Andrade MA, Chacón P, Merelo JJ, Morán F (1993) Evaluation of secondary structure of proteins from UV circular dichroism using an unsupervised learning neural network. Protein Eng 6:383–390

Arrigo P, Giuliano F, Scalia F, Rapallo A, Damiani G (1991) Identification of a new motif on nucleic acid sequence data using Kohonen's self-organizing map. CABIOS 7:353–357

Barton GJ (1992) Multiple sequence alignment and flexible pattern matching. In: Taylor WR (ed) Patterns in protein sequence and structure. Springer, Berlin Heidelberg New York, pp 29–52

Benner SA (1992) Predicting de novo the folded structure of proteins. Curr Opin Struct Biol 2:121–181

Blundell TL (1992) Patterns of sequence and 3-D structure variation in families of homologous proteins: lessons for tertiary templates and comparative modelling. In: Taylor WR (ed) Patterns in protein sequence and structure. Springer, Berlin Heidelberg New York, pp 189–204

Brown JK (1994) Bootstrap hypothesis tests for evolutionary trees and other dendograms. Proc Natl Acad Sci USA 91:12293–12297

Casari G, Sander C, Valencia A (1995) A method to predict functional residues in proteins. Nature Structural Biol 2:171–178

Ferrán EA, Ferrara P (1991) Topological maps of protein sequences. Biol Cybern 65:451–458

Ferrán EA, Pflugfelder B, Ferrara P (1994) Self-organized neural maps of human protein sequences. Protein Sci 3:507–521

Göbel U, Sander C, Schneider R, Valencia A (1994) Correlated mutations and residue contacts in proteins. Proteins 18: 309–317

Hanke J, Beckmann G, Bork P, Reich JG (1996) Self-organizing hierarchic networks for pattern recognition in protein sequence. Protein Sci 5:72–82

Higgins DG, Sharp PM (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene 73:237–244

Lichtarge O, Bourne HR, Cohen FE (1996) An evolutionary trace method defines binding surfaces common to protein families. J Mol Biol 257:342–358

Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Ouzounis C, Sander C, Scharf M, Schneider R (1993) Prediction of protein structure by evolution of sequence-sequence fitness aligning sequences to contact profiles derived from three-dimensional structures. J Mol Biol 232:805–825

Rost B, Sander C (1993) Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 232:584–599

Rost B, Sander C (1994) Combining evolutionary information and neural networks to predict protein secondary structure. Proteins 19:55–77

Schneider R, Sander C (1991) Database of homology-derived structures and the structural meaning of sequence alignment. Proteins 9:56–68

Valencia A, Chardin P, Wittinghofer A, Sander C (1991) The *ras* protein family: evolutionary tree and role of conserved amino acids. Biochemistry 30:4637–4648

Zuckerkandl E, Pauling L (1965) Evolutionary divergence and convergence in proteins. In: Bryson V, Vogel HJ (eds) Evolving genes and proteins. Academic Press, New York, pp. 97–166