Protein Science

Self-organizing tree-growing network for the classification of protein sequences

H. C. WANG, J. DOPAZO, L. G. DE-LA-FRAGA, Y. P. ZHU and J. M. CARAZO

Protein Sci. 1998 7: 2613-2622

Protein Science (1998), 7:2613–2622. Cambridge University Press. Printed in the USA.
Copyright © 1998 The Protein Society

Self-organizing tree-growing network for the classification of protein sequences

HUAI-CHUN WANG,[1,2] JOAQUIN DOPAZO,[3] LUIS GERARDO DE LA FRAGA,[1]
YUN-PING ZHU,[4] AND JOSE MARIA CARAZO[1]

[1]Centro Nacional de Biotecnologia-CSIC, Universidad Autonoma, 28049 Madrid, Spain
[2]Institute of Medical Information (AMMS), 27 Taiping Road, 100850 Beijing, China
[3]GlaxoWellcome S.A., C/ Severo Ochoa 2, 28760 Tres Cantos, Spain
[4]Institute of Radiation Medicine (AMMS), 27 Taiping Road, 100850 Beijing, China

(RECEIVED January 22, 1998; ACCEPTED August 21, 1998)

Abstract

The self-organizing tree algorithm (SOTA) was recently introduced to construct phylogenetic trees from biological sequences, based on the principles of Kohonen's self-organizing maps and on Fritzke's growing cell structures. SOTA is designed in such a way that the generation of new nodes can be stopped when the sequences assigned to a node are already above a certain similarity threshold. In this way a phylogenetic tree resolved at a high taxonomic level can be obtained. This capability is especially useful to classify sets of diversified sequences. SOTA was originally designed to analyze pre-aligned sequences. It is now adapted to be able to analyze patterns associated to the frequency of residues along a sequence, such as protein dipeptide composition and other n-gram compositions. In this work we show that the algorithm applied to these data is able to not only successfully construct phylogenetic trees of protein families, such as cytochrome *c*, triosephophate isomerase, and hemoglobin alpha chains, but also classify very diversified sequence data sets, such as a mixture of interleukins and their receptors.

Keywords: amino acid sequences; classification; neural network; phylogenetic reconstruction; self-organizing maps

Neural networks (NNs) have several unique features and advantages over conventional statistical methods: they incorporate both positive and negative information; they are able to detect second- and higher-order correlation in patterns and a preconceived model is not required. These features make them particularly suitable for molecular sequence analysis. Since NN methods were first introduced in the analysis of sequence data to distinguish ribosomal binding sites from nonbinding sites (Stormo et al., 1982), these techniques have found their applications in various fields of sequence analysis, including DNA intron/exon discrimination and gene identification, DNA and protein pattern analysis, protein secondary and tertiary structures prediction, protein family classification, and phylogenetic analysis (for a recent review, see Wu, 1997).

Neural networks may be classified as supervised or unsupervised according to their learning algorithms. A supervised network is trained by a data set of predefined organization scheme (e.g., a database organized according to family relationships), and used to classify new sequences into the data set. An unsupervised network, on the other hand, defines its own organization scheme according to the degree of sequence kinship (Wu et al., 1995). Unsupervised learning has the advantage that no previous knowledge about the system under study is required, and hence, it is appropriate for pattern analysis of diversified data sets. This approach has been applied to detect signal peptide coding region and potentially functional regions of nucleic acids (Arrigo et al., 1991; Giuliano et al., 1993), protein sequence classification (Ferran & Ferrara, 1991; Ferran et al., 1994; Andrade et al., 1997), and protein pattern recognition (Hanke et al., 1996; Hanke & Reich, 1996). In all the former applications, a special kind of unsupervised learning scheme, the Kohonen self-organizing map (SOM) algorithm (Kohonen, 1990, 1997), was implemented for network training and classification. This approach generates a mapping from a high-dimensional input signal space to, in general, a two-dimensional output space.

Recently, we proposed a new type of unsupervised growing self-organizing neural network that expands itself by following the taxonomic relationships that exist among the sequences being classified. This network, named self-organizing tree algorithm or SOTA (Dopazo & Carazo, 1997), combines Kohonen's SOM and Fritzke's unsupervised growing cell structures (GCS) (Fritzke, 1994) to classify sequences and construct phylogenetic trees. The network is capable of following a dynamic growth pattern, changing the number of nodes as dictated by the variability actually found in the specific data set under analysis. The algo-

Reprint requests to: Jose Maria Carazo, Centro Nacional de Biotecnologia-CSIC, Universidad Autonoma, 28049 Madrid, Spain; e-mail: carazo@cnb.uam.es.

rithm has been designed in such a way that the growth of the network can be stopped when the sequences assigned to a node have a similarity above a user-given threshold. In this way a phylogenetic tree resolved at a higher level can be obtained. This feature has proved very important when analyzing sets of diversified sequences.

The original SOTA was designed to analyze pre-aligned protein sequences. However, it is a fact that patterns associated to residue frequency along a sequence have been successfully used for sequence classification. Among these derived patterns we highlight protein dipeptide composition (van Heel, 1991; Ferran & Ferrara, 1991; Ferran et al., 1994) and the n-gram composition, that extracts and counts the occurrence of n consecutive residues (n-gram) from a sequence string in a sliding window fashion (Wu et al., 1992). Although some of the neighborhood information in a sequence may be lost in these codings, they allow for comparison of sequences of different lengths without having to align them. With these considerations in mind, we present in this work an application of SOTA that uses protein dipeptide composition and two best n-gram codings (AE12 and A2E4, Wu et al., 1992, 1996) as input data. The ability of SOTA to handle large sequence sets is demonstrated by constructing phylogenetic trees of three protein families—cytochrome *c*, triosephophate isomerase, and hemoglobin alpha chains, as well as classifying a mixture of sequences of interleukins/receptors.

Results

In this work we check the performance of SOTA with different types of input data. We use the name SOTA/SEQ for the SOTA algorithm using aligned sequences for its input, SOTA/DP for SOTA using a matrix of dipeptide composition for its input, SOTA/AE12 and SOTA/A2E4 when using the n-grams AE12 and A2E4 as inputs. PHYLIP/NJ (Felsenstein, 1993) is referred as the neighbor-joining method of PHYLIP to construct phylogenetic trees, which is used here for control tests. PHYLIP programs use the same aligned sequences as SOTA/SEQ.

Protein families: Cytochrome c, triosephosphate isomerase, and hemoglobin alpha chains

Cytochrome *c*, triosephosphate isomerase (TPI), and hemoglobin are often used as models for phylogenetic reconstruction due to their conservation in evolution and large numbers of sequences from various species are known. SOTA and PHYLIP programs were used to analyze sequences of the three families, respectively. The results are summarized in Tables 1–3.

The results presented above show that the four types of SOTA using different types of input data (aligned sequences, dipeptide composition, AE12, and A2E4) as well as PHYLIP/NJ produce a clustering that is consistent with the taxonomy of the three protein families. For cytochrome *c* and hemoglobin alpha chain families, PHYLIP/NJ presented the best classification accuracy, followed by SOTA/SEQ. The three n-gram based methods (SOTA/DP, SOTA/AE12, and SOTA/A2E4) got similar classification accuracy. For the TPI family, while PHYLIP/NJ still produced the best result, SOTA/A2E4 performed much better than SOTA/SEQ, SOTA/DP, and SOTA/AE12, especially at the more stringent level (accuracy-1) that counts only the largest clusters of each taxa classified by the methods (Table 2).

The last columns of Tables 1–3 give the CPU time in minutes that was consumed on a SGI R10000 server by the four SOTA methods to classify the sequences. It is clear that SOTA/DP and SOTA/AE12 are much faster than SOTA/SEQ and SOTA/A2E4.

Table 1. *91 Cytochrome c sequences clustered by SOTA and PHYLIP/NJ*

Methods	Plant	Protozoa	Fungi	Insect	Ave	Reptilia	Pisces
Sota/seq	29+3[a];2+1[b]	3;1[c]	13+2;1	6;3	6	2;1	2;2
Sota/dp	27;2;2	3;1	14+2	9+1	6	2;1	2;1;1
Sota/ae12	27;2;2	3;1	14+2	9+1	6	2;1	2;1;1
Sota/a2e4	26;2;2;1	3;1	11+2;3+1	8+2;1	6	2;1	2;2+2
Phylip/nj	27;2;2	4+2	14+2	7;2	6	1;1;1	4
Taxa number	31	4	14	9	6	3	4

Methods	Amphibia	Mammalia	Others	Accuracy-1[d] (%)	Accuracy-2[e] (%)	CPU time (min)
Sota/seq	1	11;3	3;2	80.77	96.70	33.73
Sota/dp	1	11+1;3	3+1;2	82.97	93.96	22.03
Sota/ae12	1	11+1;3	3+1;2	82.97	93.96	30.35
Sota/a2e4	1	14	2;1;1;1	80.22	90.66	125.68
Phylip/nj	1	11;3	3;2	83.52	94.51	
Taxa number	1	14	5			

[a] "29+3" means that three additional species inserted in the cluster of this taxa (plant).

[b] "29+3;2+1" means the taxa is grouped into two clusters, one consisting of 29 species of the taxa and 3 other species, the other of 2 species of the taxa and 1 other species.

[c] "3;1" means the taxa is grouped into two clusters, one consisting of three species, the other of one species. Other similar expressions in the table and tables below have the same respective meaning.

[d] Accuracy-1 calculates the accuracy of the classification based on the biggest cluster of each taxa.

[e] Accuracy-2 calculates the accuracy of the classification based on the two biggest clusters of each taxa.

Table 2. *70 TPI sequences clustered by SOTA and PHYLIP/NJ*

Methods	Prokaryote	Archae[a]	Plant	Protozoa	Fungi	Insect	Vert[b]	Others
Sota/seq	20;5;1;1	5	11;1	3;2;1	4+1	6+2	7	2;1
Sota/dp	20+1;6+3;1	5	9;2;1	3;2;1	2;2+1	6+1	7	3
Sota/ae12	18+1; 9+3	5	9;3	3;2;1	4+1	6+2	7	2;1
Sota/a2e4	27+2	5	11;1	5;1	3+2;1	6+2	7	2;1
Phylip/nj	27	5	12	5;1	4	6	7	2;1
Taxa number	27	5	12	6	4	6	7	3

Methods	Accuracy-1[c] (%)	Accuracy-2[d] (%)	CPU time (min)
Sota/seq	80.71	92.86	27.55
Sota/dp	77.14	90.71	5.72
Sota/ae12	74.29	93.57	6.15
Sota/a2e4	90.00	95.71	56.72
Phylip/nj	97.14	100.00	

[a] Archae, archaebacteria.
[b] Vert, vertebrate.
[c] Accuracy-1 calculates the accuracy of the classification based on the biggest cluster of each taxa.
[d] Accuracy-2 calculates the accuracy of the classification based on the two biggest clusters of each taxa.

This is directly related with the number of input vector components of each sequence that they are handled: n being the number of the sequences, SOTA/DP has the least input vector components to handle, that is $n * 400$ for each sequence; SOTA/AE12 has $n * (20 + 6 + 400 + 36) = n * 462$ input vector components; SOTA/A2E4 has $n * (400 + 1{,}296) = n * 1{,}696$ input vector components; SOTA/SEQ handles $n * 21 * l$ (l is the aligned sequence length). In the case of hemoglobin alpha chain (134 ungapped aligned positions), the input components number for SOTA/SEQ is $n * 21 * 134 = n * 1{,}814$. So both SOTA/A2E4 and SOTA/SEQ usually takes much longer time than SOTA/DP and SOTA/AE12 to get a result. However, although the number of A2E4 input components is definite (1,696), the input components for SOTA/SEQ is determined by sequence length (multiplied by 21). Thus, for a long sequence alignment, SOTA/SEQ will take much more time than SOTA/A2E4 to classify the sequences.

Building a high-level tree

SOTA, but not PHYLIP/NJ or other often-used phylogenetic tree programs, presents a unique capability, that is to produce a non-fully branched tree, in which only the deeper branches of the phylogeny have been resolved. In this way, the classification can be stopped at different taxonomic levels. All SOTA programs (for

Table 3. *185 Hemoglobin alpha chain sequences clustered by SOTA and PHYLIP/NJ*

Methods	Ave	Reptilia	Pisces	Amphibia	Mammalia[a]	Primate
Sota/seq	40+1;1	4;2;2+1;1;1	11;4+2	2;1	89+3	26;1
Sota/dp	41+2	6+2;1;1;1;1	14+1;1	1;1;1	89+3	14;11;2
Sota/ae12	41+3	5+1;1;1;1;1	9;3+1;1;1;1	2;1	89+2	16;6+1;5
Sota/a2e4	40+1;1	6+2;3;1	8;3+1;2;2	2;1	86+8;2;1	20;6;1
Phylip/nj	40;1	7;2;1	15	3	89	27
Taxa number	41	10	15	3	89	27

Methods	Accuracy-1[b] (%)	Accuracy-2[c] (%)	CPU time (min)
Sota/seq	91.89	96.22	361.38
Sota/dp	87.03	94.59	68.27
Sota/ae12	85.95	91.35	82.52
Sota/a2e4	84.59	92.97	396.35
Phylip/nj	97.84	99.46	

[a] Mammalia, data excluding those of primates.
[b] Accuracy-1 calculates the accuracy of the classification based on the biggest cluster of each taxa.
[c] Accuracy-2 calculates the accuracy of the classification based on the two biggest clusters of each taxa.

SEQ, DP, AE12, and A2E4) have this ability. The following is an illustration of the power of this capability by SOTA/SEQ as applied to the cytochrome *c* family.

We analyzed the set of 91 sequences of cytochrome *c* using different values of the SOTA parameter "resource," that is, the way to control within SOTA whether new nodes are going to be generated or not. We started with a resource value of 0.001, for which a complete tree was obtained (the result of this complete tree has been used to prepare Table 1). Then, we changed the resource value to 0.025, and the program stopped at the 86th cycle that assigned the input 91 sequences to 86 nodes, producing an "incomplete tree." Notably, the five pairs of sequences that are not separated are very similar: CYC_CANFA and CYC_MIRLE; CYC_HAEIR and CYC_LUCCU; CYC_EQUAS and CYC_HORSE; CYC_HUMAN and CYC_MACMU; CYC_DRONO and CYC_STRCA. As expected, the CPU time required (29.70 min) to generate an incomplete tree was shorter than the one to generate a fully branched tree of the 91 sequences (33.73 CPU min). As the resource parameter is set larger, more sequences will not be separated. For example, when the resource is set to 0.15, the program stops at the 25th cycle and all the sequences are assigned to 25 nodes. In this case it is important to stress again the general consistency of the result, remarking that node 21 contains all 28 vertebrates and only one fungus, node 29 contains six insects, node 41 contains five fungi, and node 17 contains 27 plants. These bigger nodes correspond to big branches of a complete tree, that is, to a high taxonomy of phylogeny (vertebrate, insect, fungi, plant, etc.). The classification at this stage just takes 0.48 CPU min, 70 times less than the time needed for the full-branched classification.

Interleukins/receptors

To examine SOTA's performance on diversified data sets, we choose to apply it to analyze 247 sequences of interleukins (IL) and IL receptors. Most of ILs belong to the hemopoietin family of cytokines and the IL receptors belong to the cytokine (hemopoietin) receptor superfamily. The hemopoietin family is characterized by a four-alpha-bundle helix structure, and the receptor superfamily characterized by four conserved cysteins and a WS x WS motif in the extracellular part (Bazan, 1990; Boulay & Paul, 1993; Cosman, 1993). Unlike cytochrome *c*, TPI and hemoglobin alpha chain, both the ligands and the receptors evolved very fast. Although their sequences are diversified even within a family, the conservation of key features in the structures of hemopoietins and receptors reflects a pattern of evolution from their respective common ancestor rather than convergence to advantageous structures (Shields et al., 1995, 1996).

SOTA/DP fully classified the 247 IL/receptor sequences, with almost all sequences successfully assigned to the correct subfamilies (Fig. 1). For example, all 14 IL-1 alpha chains are clustered together, so do all 14 IL-1 beta chains, all 17 IL-2, all 10 IL-3, all 16 IL-4, all 8 IL-1 receptors, all 8 IL-2 receptor alpha chains, etc. The branching orders within each subfamily are logical according to phylogeny. Further, the result shows that all interleukin receptors and only a few interleukins appear forming a distinct cluster that set them apart from the other interleukins. This is consistent with the classification that the interleukins and the receptors belong to different (super)families: the hemopoietin family and the receptor superfamily. Although IL-1 and IL-8 do not belong to the hemopoietin family and their receptors do not belong to the hemopoietin receptor superfamily, still, IL-1 and IL-8 were assigned to the interleukin cluster, while their receptors assigned to the receptor cluster. The 5 IL-1 receptor antagonists were assigned to the interleukin cluster. This is logical in the sense that, functionally, the receptor antagonist, like IL-1 alpha and IL-1 beta, combines IL-1 receptor. Also, sequences of IL-1 alpha, beta, and IL-1 receptor antagonists share about 25% amino acid sequence identity and the same beta-trefoil 12-stranded beta-barrel structure (Vigers et al., 1994). Although all receptors are assigned into the receptor cluster, three ILs (3 IL-11, 12 IL-12 beta chain, and human IL-14) are assigned to the "wrong" clusters, that is within the receptor cluster. However, it has been shown that the IL-12 beta chain is homologous to hemopoietin receptors (Shields et al., 1996).

SOTA/AE12 also completely classified the 247 IL/receptor sequences (figure not shown), and the classification is quite like that by SOTA/DP: almost all the interleukins and receptors are clustered into groups of same proteins; the receptors are clustered together distinctly from the interleukin clusters; IL-1 receptor antagonists are within the interleukin cluster; IL-11, IL-12 beta chains, and human IL-14 are within the receptor cluster. A close inspect of the clusters revealed that the IL-2/receptor and IL-4/receptor have a coupled relationship: IL-2 and IL-4 are clustered together and their receptors are together.

SOTA/A2E4 completely classified the IL/receptor sequences, with most of the sequences of the same proteins from different species clustered together and their branching orders are logical according to the taxonomic classification (figure not shown). However, it is not noticeable that all IL receptors are clustered into a cluster that is distinct from the interleukins, as the way observed in SOTA/DP and SOTA/AE12 results. SOTA/SEQ and PHYLIP/NJ, on the other hand, simply cannot classify the IL/receptors data set. This is due to the enormous sequence divergence within the data sets, which results in a great number of large and infinite distances among sequences of the different families and superfamily, and consequently, some of the finer branches of the tree could not be properly resolved.

Discussion

Comparison with SOM

We have used the self-organizing tree growing network (Dopazo & Carazo, 1997) to classify and reconstruct phylogenetic tree of protein families. The SOTA network is a special case of the unsupervised growing cell structure (GCS) (Fritzke, 1994), which in itself was derived from the Kohonen self-organizing map (Kohonen, 1990). The key points that SOTA differs from the GCS and SOM methods are that the network growth mimics a speciation event and the topology of the network is a binary tree, thus allow an appropriate description of the taxonomic relationships within sequences of same proteins from different species (as shown above on the analysis of the families of cytochrome *c*, TPI, and hemoglobin alpha chains). Classical SOM has been shown to map phylogenetically related sequences into same or neighboring neurons (Ferran & Ferrara, 1991). Ferran and Ferrara (1992) have observed that the Euclidean distances between the input vectors and the synaptic code vectors of a classical SOM can be sorted in a decreasing order to further classify the sequences and, thus, may construct a hierarchical trees of protein classification. But this approach is just a post-processing of a SOM result and is primarily applicable within the sequences assigned to the same winner neuron, and it is difficult to derive a complete hierarchy of all neurons

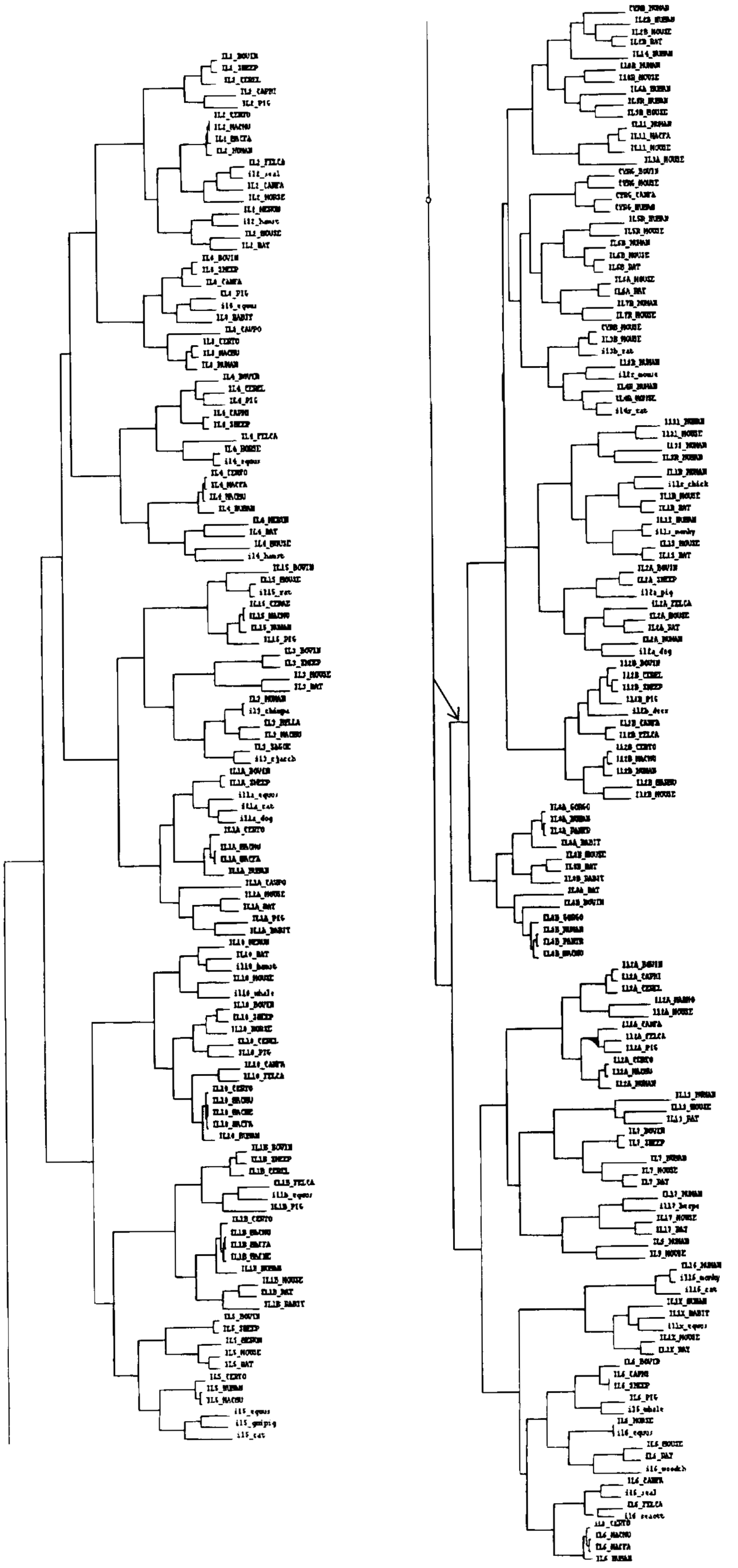

Fig. 1. Two hundred forty-seven interleukin/receptor sequences were classified by SOTA/DP using dipeptide composition as input. The receptors are grouped into a cluster distinct from the interleukins (indicated as an arrow on the graph). The same interleukins or receptors from different species are grouped together and branched consistent with the taxonomic classification. (Due to the limit of the page size, the tree was "cut" at the "root" node and printed in two columns. The top of the right side should be connected to the bottom of the left side.)

of a SOM map. In general, a single SOM training can obtain the clustering of a protein family at a definite resolution level. Andrade et al. (1997) used several SOMs with different sizes against the same protein subfamily and combine the classifications at several resolutions to get a tree classification of the protein subfamily. But this approach is not as efficient as SOTA, and it is hard to be applied to large data sets.

Classical SOM can classify different protein families into different clusters (Ferran & Ferrara, 1991, 1992; Ferran & Pflugfelder, 1993). However, the optimal number of cells is usually determined empirically (trial and error) or by statistical methods (Ferran & Pflugfelder, 1993). SOTA adopted from GCS the growing cell property: the number of cells and the connections among them are dynamically assigned during the network training. As such, SOTA can classify proteins of different families and superfamilies more efficiently. To confirm this, we have used the SOM to classify the same data set of 247 interleukin/receptors in the above section and compare with the classification by SOTA/DP. Figure 2 is a SOM topology of 16 * 16 neurons that was trained to classify the dipeptide compositions of 247 IL/receptors, the same input as used in Figure 1 by SOTA/DP. In the map, 56 neurons (22% of the total neurons) have assigned proteins to them. Similar to the result by SOTA/DP, most of the receptors are positioned to the low left corner of the map; the IL-12 beta chain, IL-14, and IL-13 are at the receptor corner. But IL-8 receptor are more distant from this corner than three interleukins (IL-13, IL-11, and IL-17) are. The interleukins are sparsely positioned over the rest of the map. Although most of the same proteins from different species are positioned into one or several neighboring or close neurons, seven neurons are shown to contain mixtures of different proteins within the same neuron. For example, cell [5, 12] contains two IL-1 receptors (IL1S_HUMAN and il1s_monky) and one IL3 (IL3_MOUSE); cell [5, 15] contains two IL-2 alpha chains (IL2A_BOVIN and IL2A_SHEEP), and one IL-3 receptor class II alpha chain

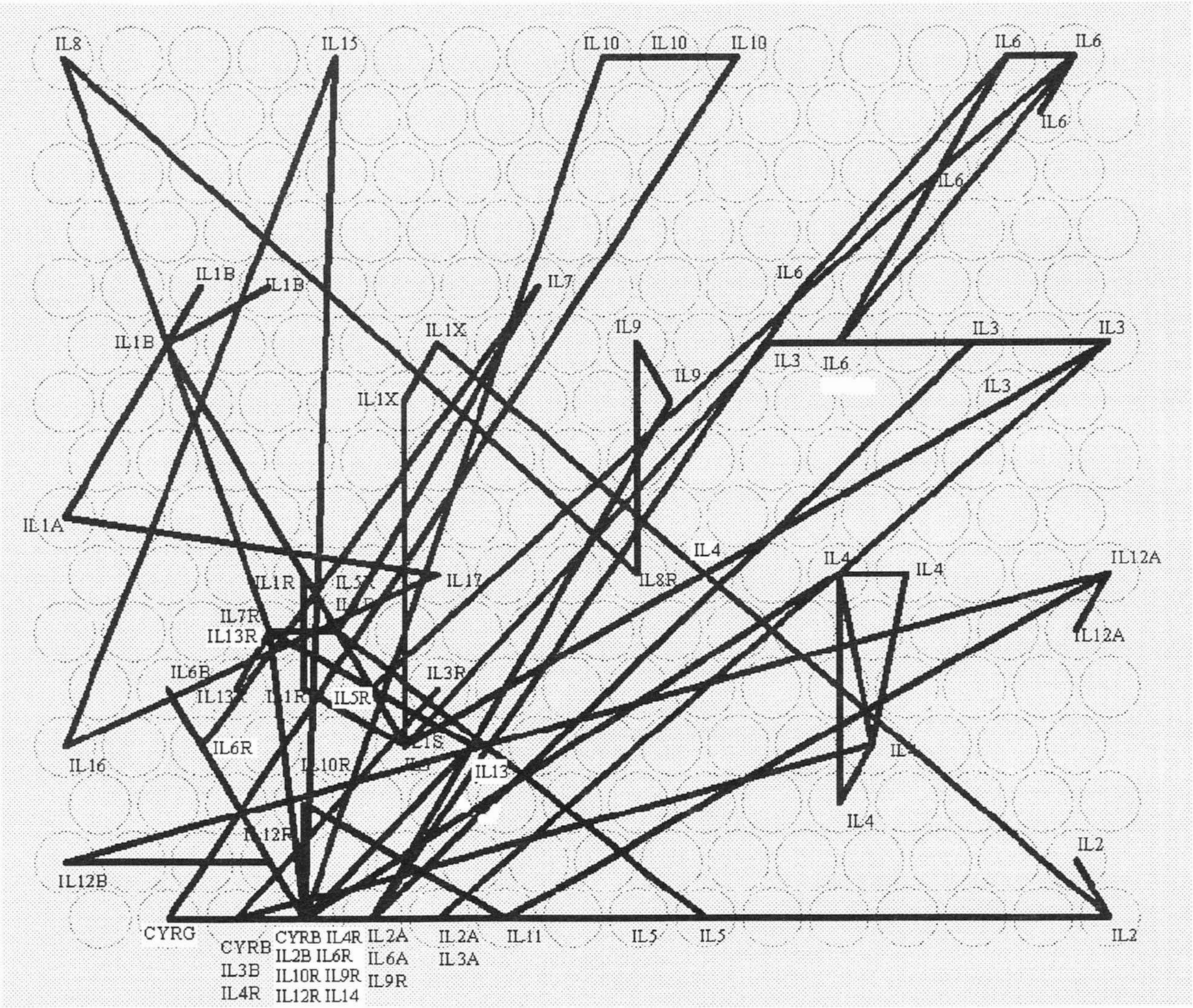

Fig. 2. Hexagonal topology of self-organizing map obtained with the dipeptide composition of the learning set of 247 interleukin/receptors. The trajectory was formed of the best-matching units (winners) and plotted by the program "planes" of the SOM package. Same proteins from different species are positioned to one neuron or several neighboring neurons. Most of the receptors are positioned at the low left corner of the map. The 16 * 16 network was trained during 1,000 and 5,000 epochs. The parameters of the learning process are given in Materials and methods (Self-organizing map section).

(IL3A_MOUSE); cell [3, 15] contains one cytokine receptor common beta chain (CYRB_HUMAN), one IL-10 receptor (I10R_HUMAN), one IL-12 receptor (i12r_mouse), one IL-14 (IL14_HUMAN), 3 IL-2 receptor beta chains (IL2B_HUMAN, IL2B_MOUSE, and IL2B_RAT), one IL-4 receptor (IL4R_HUMAN), two IL-6 receptor alpha chains (IL6A_MOUSE and IL6A_RAT), and one IL9 receptor (IL9R_MOUSE), etc. These mixtures of proteins do not occur in the result by SOTA/DP (Fig. 1). The above analyses prove that SOTA method outperforms SOM in this case.

Encoding of input sequence

For a defined neural network algorithm, such as SOTA or SOM, the coding of sequences for input vectors is key to its performance. In general, the sequences to be used as input to a neural net program such as SOTA can be encoded in two different ways: (1) direct sequence encoding, which uses an indicator vector of binary numbers (0 or 1) to represent the "identity" of each residue in the sequence string (Casari et al., 1995; Andrade et al., 1997; Dopazo & Carazo, 1997). For a protein sequence, an amino acid is represented as a vector of 21 input units (20 zeros and a single one), which includes an extra unit for the gaps. The sequences should be aligned before encoding. As a common practice, all positions that have at least one gap in a column of the sequence alignment are removed prior to be presented to SOTA/SEQ. (2) Residue frequency of a sequence, such as protein dipeptide composition (Ferran & Ferrara, 1991; van Heel, 1991). The main advantage of dipeptide composition is that the sequences do not need to be previously aligned. The main disadvantage is the loss of valuable information contained in contiguous amino acid ordering. This is the reason why SOTA/DP produces suitable sequence classifications in the data sets of cytochrome *c*, TPI and hemoglobin, but they are generally not as good as SOTA/SEQ in the three cases. However, the simplified encoding in SOTA/DP is especially useful for diversified data sets, in which a proper sequence alignment is not available. The case of interleukin/receptors clearly demonstrates this point: both SOTA/DP and SOM that uses dipeptide composition as inputs can easily handle this data set, and suitable classifications were made, while both SOTA/SEQ and PHYLIP/NJ, which need pre-aligning the sequences, cannot classify the data sets.

Despite the success of dipeptide composition used for sequence encoding in a neural net program, any improvement over this residue frequency-based method is expected to get a better result. The "n-grams" method (Wu et al., 1992) extends the dipeptide frequency coding. The results presented in the last section show that AE12, one of the best coding method in others application (Wu et al., 1992), seems to be at most marginally better than SOTA/DP. In fact, the classification of cytochrome *c* by SOTA/AE12 is almost the same as that by SOTA/DP (Table 1), and the classifications of interleukin/receptors by SOTA/AE12 and SOTA/DP are very similar. This is not surprising, because AE12 does not extract more information from the sequence ordering than the only A2 (dipeptide composition), and of the 462 vector values of AE12 that are derived from each sequence, 400 are the same vector values as A2. The result by SOTA/AE12 is much affected by this large component. SOTA/A2E4, using A2 and E4 as inputs, presented a similar classification accuracy as SOTA/DP and SOTA/AE12 did in the classification of the cytochrome *c* and hemoglobin alpha chain families. However, in the case of the TPI family, SOTA/A2E4 produced a much better result than SOTA/DP and SOTA/AE12 and the classification is even better than SOTA/SEQ (Table 2). This suggests that the information of sequence ordering kept in the E4 coding has enhanced the coding of A2. It is expected that using even larger n-grams, such as A3 and E5, may further increase the quality of the classification by SOTA. However, the input vector is too large and too sparse (too many zeros) in these codings, which makes SOTA to be too difficult to handle them directly. A proper data decompression, such as singular value decompression, has been recently introduced into a neural program for sequence classification (Wu et al., 1995). This suggests that the performance by SOTA may be further improved by these methods.

Conclusions

SOTA has been developed to classify protein/DNA sequences and construct a phylogenetic tree by a self-organizing tree-growing approach, a special type of Kohonen neural mapping (Dopazo & Carazo, 1997). In this work SOTA has been further developed in such a way that it can use directly aligned sequence as input (SOTA/SEQ) as well as a matrix of dipeptide composition (SOTA/DP) and composition of other n-grams, such as AE12 and A2E4 (SOTA/n-gram). For well-conserved sequences, the usage of SOTA/SEQ is recommended, and a better classification should be expected. For diversified data sets of sequences on which a good alignment cannot be achieved, the SOTA/n-gram is the method of choice, because it is fast and does not require the sequences to be aligned.

Common to the two types of SOTAs is their ability to stop the classification at high taxonomy levels, a feature that may open new venues in the field of classifying very large and diversified data sets, such as, for instance, all human sequences.

Materials and methods

Self-organizing tree algorithm

A detailed description of the learning algorithm of SOTA can be found in Dopazo and Carazo (1997). The following is a brief summary of the procedures (Fig. 3): (1) encode input sequences: convert the sequences into input vectors; (2) initialize system: initialize the code vectors associated with each output node to random values; (3) run a cycle; and (4) if the end of the network growth is not reached, attach two new neurons to the neuron having the larger resource value and go to 3.

A cycle consists of as many epochs as necessary to get convergence in the network at a given taxonomic level. Convergence is achieved when the network error between two epochs is below a given threshold. An epoch consists of the presentation of all the input data (aligned sequences or n-gram composition) in the following steps:

Step 1: Compute distances to all external neurons (tree leaves)

Distance between input vector *j* and the neuron *i* is computed as the Euclidean distance if the input vectors are n-gram composition matrices, or as follows if the input vectors are sequences:

$$d_{S_j C_i} = \frac{\sum_{l=1}^{L}\left(1 - \sum_{r=1}^{A} S_j[r,l]\cdot C_i[r,l]\right)}{L} \qquad (1)$$

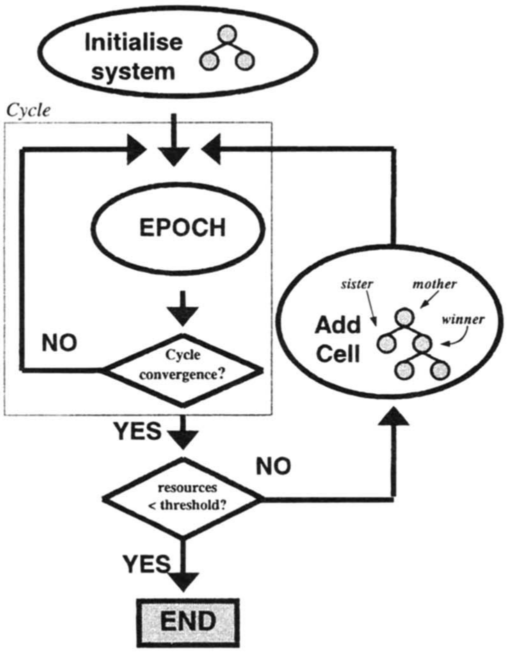

Fig. 3. A schematic diagram of the dynamics of the SOTA algorithm.

where $S_j[r,l]$ is the value for the residue r of the input sequence node j at the presentation t and $C_i[r,l]$ is the residue r of the neuron i at the presentation t, l accounts for the summation over all the L sites of the sequence and r accounts for the summation over all the A entries corresponding to all the possible residues of the alphabet (20 + 1 for protein sequence).

Step 2. Select output neuron i^* with minimum distance d_{ij}

Step 3. Update neuron i^* and neighbors

Neurons are updated as:

$$C_i(\tau+1) = C_i(\tau) + \eta_{t,i}\cdot(S_j - C_i(\tau)) \qquad (2)$$

where η is a gain term that decreases in time and depends on which neuron (winner, mother, or sister neuron) is updated:

$$\eta_{t,i} = \alpha_i\cdot\left(1 - \frac{t}{M_t}\right) \qquad (3)$$

where α_i is the constant parameter for the updated neuron (winner, mother, or sister neuron), t is the total number of presentations, M_t is the maximum number of presentations allowed and is obtained as $\mu \times A \times L$, where μ is user defined initial iteration times, A is the number of residues in the alphabet, L is sequence length. For the input being matrix of dipeptide composition, both A and L are constant values, i.e., 20.

At the end of each cycle, the network grows by attaching two new neurons to the neuron having higher resources (larger than the user defined resource threshold). The resource for a neuron is defined as the mean value of the distances among a neuron and the input sequences associated to it:

$$R_i = \frac{\sum_{k=1}^{K} d_{S_k C_i}}{K} \qquad (4)$$

where the summation is done over the K sequences assigned to the neuron j. The user-defined resource threshold controls the stopping of the network growth. If it is set to zero or a small value, the network will grow until every input sequence is associated to a unique neuron, producing a complete tree. A larger value of the resource threshold will cause the network to stop at higher taxonomic levels, clustering in a single neuron set of sequences whose heterogeneity has associated a value for the neuron resources that falls below the threshold, and thus, building a high-level tree.

SOTA parameters

The following main default parameters were used in the application of SOTA: initial iteration time, 100; threshold, 0.001; final resource, 0.001; updating parameters for winner and its direct neighbors (mother and sister cells), 0.1, 0.05, and 0.01. A bigger iteration time is needed sometimes to achieve neural net convergence. The updating parameters may need adjusted to "fine tune" the branch orders and produce a better classification. In all our tests using different combinations of the three parameters defined above, the relationship that, updating parameter for winner cell > updating parameter for mother cell ≥ updating parameter for sister cell, had to be maintained in order for SOTA to reach convergence. On the other hand, increasing the value of the resource threshold results in convergence without fully branching the sequences and, thus, produces a phylogenetic tree resolved at higher levels.

Sequence encoding: n-gram method

The n-gram method, as proposed by C. Wu for sequence encoding (Wu et al., 1992, 1995), extracts and counts the occurrences of n-gram patterns (i.e., *n* consecutive residues) from a sequence string in a sliding window fashion. In the encoding, the standard 20 amino acids are defined as set A; then A1 is the count of each of the 20 amino acids, i.e., amino acid composition of a protein; A2 is the count of every two consecutive residues, or dipeptide composition, etc. Amino acids can be grouped according to their physicochemical, structural, and evolutionary features. For example, one commonly used grouping is {MILV}, {FYW}, {STPAG}, {DENQ}, {HRK}, {C}. The 20 amino acids can be changed to a six-letter expression to reflect the exchange relationship of the residues (defined as set E), and a protein sequence can be changed to a sequence of these six letters, then there are counts (or frequencies) of E1, E2, E3, etc. The first character of the different types of n-gram patterns is a letter designating the alphabet set (A or E); the second character is a digit representing the size (length) of the n-gram. Different types of n-grams can be concatenated for input vectors, such as A2E4 merging patterns of A2 and E4, and AE12 merging A1, E1, A2, and E2. The counts of the n-gram patterns are scaled to values between 0 and 1; different types of n-grams (such as A1, E1, A2, and E2) are scaled separately, so that their values will not be skewed.

Evaluation mechanism for classification accuracy

The classification accuracy is based on both the total number of correct phylogenetic assignments (true positive) and the total number of incorrect assignments (false positives). A sequence is considered to be accurately classified if it is assigned to the correct phylogenetic taxa. For example, if human hemoglobin alpha chain is assigned to the "primate" cluster as well as the "mammalia" cluster, it is believed to be correctly classified; if chicken hemoglobin alpha chain is not assigned to the "ave" cluster but any other cluster, then it is wrongly classified. For every sequence that is correctly assigned a score of +1 is given, and for an incorrectly assigned sequence a score of −0.5 was given for penalty. The classification accuracy is calculated as the sum of the scores for each taxa divided by the total number of sequences of a query protein family. It is measured at two stringencies: accuracy-1 calculates the accuracy based on the biggest cluster of each taxa, and accuracy-2 calculates the accuracy based on the two biggest clusters of each taxa.

Self-organizing map

The SOM program package (version 3.1) was downloaded from the Web (http://www.cis.hut.fi/nnrc/nnrc-programs.html). The learning process was accomplished by running the following batch file:

```
randinit -din il.dat -cout il.cod -xdim 16 -ydim 16 -topol hexa
-neigh gaussian

vsom -din il.dat -cin il.cod -cout il.cod -rlen 1000 -alpha 0.05
-radius 5 -rand

qerror -din il.dat -cin il.cod

vsom -din il.dat -cin il.cod -cout il.cod -rlen 5000 -alpha 0.03
-radius 3 -rand

qerror -din il.dat -cin il.cod

vcal -din il.dat -cin il.cod -cout il.cod

visual -din il.dat -cin il.cod -dout il.vs
```

The program "visual" generates a list of coordinates (the final topology) corresponding to the best-matching unit in the map for each data sample in the data file.

Data sets

The protein family members were retrieved from SWISS-PROT and other sequence databases based on sequence annotation. For the cytochrome *c* family, sequences of cytochrome *c*2, cytochrome *c*550, and cytochrome *c*553 were removed from the data set. Redundant sequences were removed. Sequences having less than half of the average full length of the query family members were also discarded. The following is a simplified list of the ID codes of all sequences used in this study. The real SWISS-PROT IDs are in capital letters. Sequences taken from other databases were renamed in the style of ID codes of SWISS-PROT, but in small letters, followed by their accession number or locus name of the respective database [e.g., tpis_pwoes (PIR S66212)].

(1) **Cytochrome *c*.** 89 CYC_*; CYC1_DROME, CYC1_YEAST, CYC2_ASCSU. Total 91 cytochrome *c* sequences were collected, each representing a different specie, length ranging from 93 to 113 amino acids. (2) **Triosephosphate isomerase** (TPI). 62 TPIS_*; TPIC_SECCE, TPIC_SPIOL, TPI1_GIALA, TPI2_GIALA, tpis_phori (DDBJ accession AB009528), tpis_pwoes (PIR locus S66212), tpis_mther (GenBank accession AE000876), tpis_afulg (GenBank AE001014). A total of 70 TPI sequences were compiled, the length ranging from 150 to 322 amino acids. (3) **Hemoglobin alpha chains**. 178 HBA_*; hba_pigeon (PIR A37011), hba_flamin (PIR HAGDA), hba_turdov (PIR S55247), hba_goose (PRF locus 754932A), hba_caiman (PRF 0901255A), hba_ltard (PRF 0409309A), hba_tsp (PRF 765952A). (For PRF database, see http://prfsun2.prf.or.jp/index.html.) A total of 185 sequences were compiled, the length ranging from 141–143 amino acids. (4) **Interleukin/receptors**. 247 interleukins (IL), IL receptors, and IL-related proteins were compiled from the databases, the length ranging from 60 to 918 amino acids, including the following sequences: **IL-1 alpha**, 14: 11 IL1A_*, il1a_equus (DDBJ D42146), il1a_cat (GenBank AF047012) and il1a_dog (GenBank AF047011); **IL-1 beta**, 13: 12 IL1B_*, il1b_equus (DDBJ D42147); **IL-1 receptor antagonist**, 5: 4 IL1X_*, il1x_equus (DDBJ D83714); **IL-2**, 17: 15 IL2_*, il2_seal (GenBank U79187), il2_hamst (GenBank AF046212); **IL-3**, 10: 8 IL3_*, il3_cjacch (PIR S42721), il3_chimpa (PIR S42720); **IL-4**, 16: 14 IL4_*, il4_equus (GenBank AF035404), il4_hamst (GenBank U50415); **IL-5**, 11: 8 IL5_*, il5_equus (GenBank U91947), il5_cat (GenBank AF051372), il5_guipig (GenBank U34588); **IL-6**, 18: 13 IL6_*, il6_whale (GenBank L46803), il6_equus (GenBank U64794), il6_seaott (GenBank L46804), il6_seal (GenBank L46802), il6_woodch (EMBL accession Y14139); **IL-7**, 5: 5 IL7_*; **IL-8**, 10: 9 IL8_*, il8_equus (GenBank AF062377); **IL-9**, 2: 2 IL9_*; **IL-10**, 17: 15 IL10_*, il10_whale (GenBank U93260), il10_hamst (GenBank AF046210); **IL-11**, 3: 3 IL11_*; **IL-12 alpha**, 11: 11 I12A_*; **IL-12 beta**, 12: 11 I12B_*, il12b_deer (GenBank U10160); **IL-13**, 3: 3 IL13_*; **IL-14**, 1: IL14_HUMAN; **IL-15**, 7: 6 IL15_*, il15_rat (GenBank U69272); **IL-16**, 3: IL16_HUMAN, il16_monky (GenBank S80645), il16_cat (GenBank AF003701); **IL-17**, 4: 3 IL-17_*, il17_herpe (EMBL Y13183). **IL-1 receptor type I**, 4: 3 IL1R_*, il1r_chick (PIR JQ1526); **IL-1 receptor type II**, 4: 3 IL1S_*, il1s_monky (GenBank U64092); **IL-2 receptor alpha**, 8: 6 IL2A_*, il2a_dog (GenBank AF056491), il2a_pig (GenBank U78317); **IL-2 receptor beta**, 3: 3 IL2B_*; **IL-2 receptor gamma (cytokine receptor common gamma chain)**, 4: 4 CYRG_*; **cytokine receptor common beta chain**, 2: 2 CYRB_*; **IL-3 receptor alpha**, 2: IL3A_MOUSE, IL3R_HUMAN; **IL-3 receptor beta**, 2: IL3B_MOUSE, il3b_rat (PIR I56563); **IL-4 receptor alpha**, 3: 2 IL4R_*, il4r_rat (PIR S31575); **IL-5 receptor alpha**, 2: 2 IL5R_*; **IL-6 receptor alpha**, 3: 3 IL6A_*; **IL-6 receptor beta**, 3: 3 IL6B_*; **IL-7 receptor**, 2: 2 IL7R_*; **IL-8 receptor A chain**, 5: 5 IL8A_*; **IL-8 receptor B chain**, 8: 8 IL8B_*; **IL-9 receptor**, 2: 2 IL9_*; **IL-10 receptor**, 2: 2 I10R_*; **IL-12 receptor**, 2: 2 I12R_*; **IL-13 receptor**, 3: 2 I131_*, I132_HUMAN.

Programs and availability

The SOTA is a set of programs and subroutines written in ANSI-C. They have been implemented and tested on SGI R10000 server (IRIX6.2, 196 MHz, 768 MB memory). The SOTA result is a plain text file. DRAWER, a Microsoft Windows based program in Visual C/C++, has been written to transfer the plain SOTA result to a tree graph as shown in Figure 1. These programs as well as some utility programs are available on the Internet by anonymous FTP (ftp://cnb.uam.es/pub/cnb/sota) or via World Wide Web (http://www.cnb.uam.es/~bioinfo/Software/sota/sotadocument.html). SOTA source codes, Makefile, README, sample input/output files, and data files used in this study can be found in the compressed "tar" file (sotasrc.tar.gz).

The PHYLIP package has been used to construct phylogenetic trees that were then used to evaluate trees built by SOTA. A consensus PHYLIP tree is constructed following a series of executions of PHYLIP programs: First bootstrap the data set to 100 duplicate sets using SEQBOOT, then calculate distance matrices by PRODIST using Dayhoff's method, cluster the sequences by the neighbor-joining method, and finally get a consensus tree by CONSENSE.

Acknowledgments

We thank Cathy Wu and Oswaldo Trelles for discussion of sequence encoding in a neural network program and the two anonymous referees for critical reviews of the manuscript. Carlos Sanchez helped preparing Figure 1. H.-C. Wang is supported by a CSIC (Spanish National Research Council) fellowship for scholar exchange between Spain and China. Financial help from the Spanish CICYT through project number BIO95-0768 and BIO98-0761 is greatly appreciated.

References

Andrade MA, Casari G, Sander C, Valencia A. 1997. Classification of protein families and detection of the determinant residues with an improved self-organizing map. *Biol Cybern 76*:441–450.

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. *Comp Appl Biosci 7*:353–357.

Bazan JF. 1990. Structural design and molecular evolution of a cytokine receptor superfamily. *Proc Natl Acad Sci USA 87*:6934–6938.

Boulay J-L, Paul WE. 1993. Hematopoietin sub-family classification based on size, gene organization and sequence homology. *Curr Biol 3*:573–581.

Casari G, Sander C, Valencia A. 1995. Functional residues predicted in protein sequence space. *Nat Struct Biol 2*:171–178.

Cosman D. 1993. The hematopoietin receptor superfamily. *Cytokine 5*:95–106.

Dopazo J, Carazo JM. 1997. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *J Mol Evol 44*:226–233.

Felsenstein J. 1993. *PHYLIP (phylogeny inference package)*, version 3.5. Seattle, Washington: Department of Genetics, University of Washington.

Ferran EA, Ferrara P. 1991. Topological maps of protein sequences. *Biol Cybern 65*:451–458.

Ferran EA, Ferrara P. 1992. Clustering proteins into families using artificial neural networks. *Comp Appl Biosci 8*:39–44.

Ferran EA, Pflugfelder B. 1993. A hybrid method to cluster protein sequences based on statistics and artificial neural networks. *Comp Appl Biosci 9*:671–680.

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

Fritzke B. 1994. Growing cell structures—A self-organizing network for unsupervised and supervised learning. *Neural Network 7*:1141–1160.

Giuliano F, Arrigo P, Scalia F, Cardo PP, Damiani G. 1993. Potentially functional regions of nucleic acids recognized by a Kohonen's self-organizing map. *Comp Appl Biosci 9*:687–693.

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

Hanke J, Reich JG. 1996. Kohonen map as a visualization tool for the analysis of protein sequences: Multiple alignments, domains and segments of secondary structures. *Comp Appl Biosci 12*:447–454.

Kohonen T. 1990. The self-organizing map. *Proc IEEE 78*:1464–1480.

Kohonen T. 1997. *Self-organizing maps*, 2nd ed. New York: Springer.

Shields DC, Harmon DL, Nunez F, Whitehead AS. 1995. The evolution of haematopoietic cytokine/receptor complexes. *Cytokine 7*:679–688.

Shields DC, Harmon DL, Whitehead AS. 1996. Evolution of hemopoietic ligands and their receptors. *J Immunol 156*:1062–1070.

Stormo GD, Schneider TD, Gold L. 1982. Use of the perceptron algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Res 10*:2997–3011.

van Heel M. 1991. A new family of powerful multivariate statistical sequence analysis techniques. *J Mol Biol 220*:877–887.

Vigers GPA, Caffes P, Evans RJ, Thompson RC, Eisenberg SP, Brandhuber B. 1994. X-ray structure of interleukin-1 receptor antagonist at 2.0-Å resolution. *J Biol Chem 269*:12874–12879.

Wu CH. 1997. Artificial neural networks for molecular sequence analysis. *Comput Chem 21*:237–256.

Wu CH, Berry M, Shivakumar S, McLarty J. 1995. Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition. *Machine Learn 21*:177–193.

Wu CH, Whitson G, McLarty J, Ermongkonchai A, Chung T. 1992. Protein classification artificial neural system. *Protein Sci 1*:667–677.

Wu CH, Zhao S, Chen H-L, Lo C-J, McLarty J. 1996. Motif identification neural design for rapid and sensitive protein family search. *Comp Appl Biosci 12*:109–118.