Topology preservation in nonvectorial SOM

Susana Vegas-Azcárate
Statistics and Decision Sciences Group
Rey Juan Carlos University
28933 Móstoles, Spain
susana.vegas@urjc.es

Abstract - *Although the SOM algorithm has been widely used with vectorial data, its principle is not restricted to metric vector spaces. Indeed, any set of items for which a similarity or pseudo-distance measure is available could be mapped onto the SOM grid in an ordered fashion. As Kohonen and Somervuo (2002) pointed out, the optimal speed of shrinking of the neighbourhood range function on nonvectorial SOM algorithm should be experimentally determined. This paper presents the use of the UDL monitoring algorithm for the nonvectorial approach to SOM learning rule.*

Key words - Bioinformatics, nonvectorial topographic maps, monitorization.

Acknowledgement - The author is grateful to Drs. Jorge Muruzábal, Marc M. Van Hulle and Temujin Gautama for their constructive comments on this paper.

1 Introduction

We are living in the new era of *silicon-based* biology, where investigations and comparative analysis of complete genomes are, for the first time, possible. Genome analysis is based on crucial concepts, concerning the processes of evolution, the mechanism of protein folding and the manifestation of protein functions. The use of computers to model such processes is restricted by the current limits of our understanding of these concepts. Indeed, no technique can be applied without a reference to the underlying biology, in other words, "no algorithm does biology".

The term Bioinformatics was coined in the mid-1980s to encompass computer applications in biological sciences. In its broad sense, the term can be considered to mean information technology applied to the analysis of biological data. In the context of genome analysis, the term was originally applied to the computational manipulation and analysis of biological sequence data, such as DNA and proteins.

2 Neural networks in Bioinformatics

This section first reviews the more relevant approaches to protein classification by means of supervised and unsupervised learning methods.

Supervised learning methods have been used to predict, for example, immunoglobulin domains [Bengio and Pouliot (1990)], surface exposure of amino acids [Holbrook et al. (1990)], disulfide-bounding states of cysteines [Muskal et al. (1990)], signal peptides [Ladunga et al. (1991)], ATP-binding motifs [Hirst and Sternberg (1991)], water-binding sites [Wade et al. (1992)], three dimensional structure of proteins [Brunak et al. (1990)] and recognizing distantly related protein sequences [Frishman and Argos (1992); Baldi (2001)].

The secondary structure of proteins has been widely studied with these supervised learning approaches [Bohr et al. (1988); Quian and Sejnowski (1988); Holley and Karplus (1989); McGregor et al. (1989); Andreassen et al. (1990); Kneller et al. (1990); Vieth and Kolinski (1991); Muskal and Kim (1992); Stolorz et al. (1992); Zhang et al. (1992); Rost and Sander (1993a,b); Baldi and Brunak (2001)].

Feed-forward artificial neural networks have also been applied to the analysis of biological sequences [Petersen et al. (1990); Von Heijne (1991); Hirst and Sternberg (1992); Baldi and Hatfield (2002)] by considering some representation of the sequences as vectorial inputs for the network.

Concerning nucleic acid sequences, this approach has been used to predict DNA-binding sites [Stormo et al. (1982); Lukashin et al. (1989); Demeler and Zhou (1991); O'Neill (1991, 1992); Horton and Kanehisa (1992)], mRNA splice sites [Brunak et al. (1990, 1991); Engelbrecht et al. (1992)], and coding regions in DNA [Lapedes et al. (1990); Uberbacher and Mural (1991); Farber et al. (1992); Snyder and Stormo (1993)]. Wu et al. (1992) have also proposed another supervised neural-network-based method to classify protein sequences into families. They have trained multilayered networks by using the backpropagation algorithm.

Since the number of entries in DNA and protein databases are enormously increasing due to Genome Projects [Watson (1990); Maddox (1992); Stolorz et al. (1992)], the application of other methods such as unsupervised learning methods will be appropriated. Moreover, computing time in standard supervised learning algorithms is usually proportional to the database size. Furthermore, in many non-hierarchical statistical approaches to cluster data,

the number of expected classes should be defined before the supervised analysis [Auray et al. (1990)]. On the other hand, unsupervised learning methods are suitable for clustering proteins without having previous knowledge of the number and composition of the final clusters.

2.1 Unsupervised learning methods

Ferrán and Ferrara (1991, 1992) have proposed the unsupervised Kohonen learning rule to cluster protein sequences into families according to their degree of sequence similarity. The final map they obtain transforms the degrees of similarity between the protein sequences of the learning set into a much simpler Euclidean distance relation in a 2D space. Furthermore, the SOM configuration results from an information compression that only retains the most relevant common features of the set of input sequences. This approach has also been applied to detect signal peptide coding regions [Arrigo et al. (1991)] and to cluster small organic molecules of analogue structure into families of similar activity [Rose et al. (1991)]. Ferrán and Ferrara (1991, 1992) studies show that the sequential SOM can be trained to obtain topological maps of protein sequences, where related proteins are finally associated to the same winner neuron, or to close neighbouring ones. The final map provides a two-dimensional geometrical representation of the relationships between the bipeptide compositions of the protein sequences. Hence, these trained maps can be applied to rapidly classify new sequences. Ferrán and Ferrara (1991) have also highlighted how this approach opens new possibilities to find efficient algorithms to organize and search for homologies in the whole protein database.

However, the predetermined structure and size of Kohonen's model may yield to limitations on the resulting mappings, especially when the data to be classified are biological sequences [Dopazo and Carazo (1997)]. A variety of models have been proposed concerning networks with variable topology or variable number of elements. Kangas et al. (1990) presented a minimum-spanning-tree network where the preservation of neighbourhood relations is done only to a small degree due to the sparse connectivity of the network. Blackmore and Miikkulainen (1992) introduced an approach with a network growing on a grid. The Neural Gas algorithm [Martinetz and Schulten (1991)] produces networks which preserve the neighbourhood relations extremely well [Fritzke (1994)]. However, this algorithm does not perform dimensionality reduction, so it is not indicated for the visualization of large biological data. Other models allow a variable number of elements, but have predefined structures such as rectangular arrays. Some examples are the interpolative algorithm [Rodrigues and Almeida (1990)] and the learning expectation method introduced by Xu (1990).

3 Vectorial representation of sequences

Since proteins may have different lengths, Ferrán and Ferrara (1991) have considered the input signals to be the 400 components of a 20×20 matrix obtained from the bipeptide composition of the protein to be learned. This way, each of the 400 components, say ς_{ij}, is the normalized frequency of the bipeptide ij in the sequence— i and j are integer numbers between 1 and 20, indicating one of the 20 possible different amino acids. These 20×20 matrices allow the algorithm to work with proteins of different lengths. A protein representation also based on the bipeptide composition was early used to classify proteins by applying statistical techniques [Nakayama et al. (1988); Van Heel (1991)]. The transformation of nucleic acid sequences having different lengths into a learning set of patterns with a constant number of

signals is also possible— by reducing the previous alphabet from 20 symbols (amino acids) to only 4 symbols (nucleic acids).

In this 20×20 matrix representation, each amino acid is taken as a different residue. In Ferrán et al. (1994) similar amino acids were grouped together before computing the 400-dimensional dipeptide histogram vectors. They consider three different representations. In the first, eleven groups of residues were considered, say, $\{V, L, I\}$, $\{T, S\}$, $\{N, Q\}$, $\{E, D\}$, $\{K, R, H\}$, $\{Y, F, W\}$, $\{M\}$, $\{P\}$, $\{C\}$, $\{A\}$ and $\{G\}$. A 11×11 matrix representation of the sequence was built by taking into account an alphabet of 11 symbols instead of 20, based on considering amino acids of similar physicochemical properties as a same kind of residue. In the second representation, six groups of residues were used to build the matrix, say, $\{V, L, I, M\}$ (hydrophobic), $\{Y, F, W\}$ (hydrophobic, aromatic), $\{P, A, G, S, T\}$ (weakly hydrophobic, neutral), $\{N, Q, E, D\}$ (hydrophilic, acid), $\{KRH\}$ (hydrophilic, basic) and $\{C\}$ (crosslink forming). Thus, a 6×6 matrix is obtained. This grouping is the one considered by the GCG software package [Devereux et al. (1984)] to determine the percentage of sequence similarity between 2 protein sequences according to the Needleman-Wunsh method. The third representation exposed by Ferrán et al. (1994) considers three groups of residues to build a 3×3 matrix, $\{V, L, I, W, A\}$ (hydrophobic), $\{Y, F, P, G, C, M\}$ and $\{N, Q, E, D, K, R, H, T, S\}$ (hydrophilic).

In Hanke and Reich (1996) the sequences were aligned and then converted into vectors by fractal encoding. In Andrade et al. (1996), each position of the sequence was represented as a 20D vector— each vector component corresponded to one amino acid. The whole sequence is then converted into an L-by-20-dimensional matrix, where L is the length of the global alignment of all sequences.

However, the simplification in the protein representation implies a degradation in sensitivity. Next section deals with the organization and clustering of nonvectorial data items. Indeed, the final aim is to cope with the masses of biological nonvectorial data in an unsupervised way.

4 The nonvectorial SOM

Similarity and distance measures have been routinely used to compare two biological sequences, such as proteins or nucleic acids. The basis of such comparisons is the information from the biochemist as to the linear sequence of elements comprising such molecules [Smith and Waterman (1981)]. Similarity measures such as Smith-Waterman, BLAST or FASTA, are appropriate for clustering large protein sequence databases with topographic maps [Somervuo and Kohonen (2000)]. In nonvectorial topographic maps, unlike the previous vectorial ones, the data sequences are not converted into histogram vectors in order to perform the clustering.

Kohonen and Somervuo (2002) have shown how to implement the SOM algorithm principle to nonvectorial data in the case of fixed-size standard maps. Interestingly, they have illustrated their method by using protein sequences as basic items and FASTA scores [Pearson and Lipman (1988); Pearson (1999)] as similarity values. Specifically, if x and y are any entities, a sufficient condition for them to be mapped into a SOM diagram is that some kind of symmetric distance function, $d(x, y)$, is definable for all pairs (x, y).

Furthermore, Kohonen and Somervuo (2002) have shown how this extension of the SOM, called here SOM-nv (see Algorithm 1), can be used for the clustering, organization

and visualization of large databases of nonvectorial items such as protein sequences. The new method, originally suggested in Kohonen (1996), allows the construction of the SOM when only a similarity measure is defined for pairs of items. Hence, a vectorial representation is not really needed, avoiding the important drawbacks and limitations typically derived from the vectorial representation of biological data. To define an ordered projection, it will be sufficient to compare the pairwise distances or dissimilarities among items [Kohonen and Somervuo (2002)].

The nonvectorial SOM is based on the batch-learning version of the SOM, and it requires the computation of the *generalized median* of symbol strings [Kohonen (1985, 1995)]. Here, the way a winning neuron is selected is

$$i_n^* = \arg\min_i \{d[x_n, z_i]\} \tag{1}$$

where $d(\cdot,\cdot)$ is the underlying pseudo-distance measure. Notice that $\mathbf{v}_n$ and $\mathbf{w}_i$ were used to define input vectors and weight vectors, respectively, living in $\mathbb{R}^d$. Now, x_n and z_i term the items and the pointers, respectively, living in the symbolic (e.g. protein) space.

The generalized median is defined as follows [Kohonen (1985, 1995)]. Let $\Upsilon = \{x_n\}$ be a set of items, and let $d[x_n, x_{n'}]$ be some distance, pseudo-distance or dissimilarity measure between x_n and $x_{n'} \in \Upsilon$. The generalized median m over Υ is defined as the item that minimizes the sum of distances to all other items in Υ,

$$m = \arg\min_{x_n \in \Upsilon} \sum_{x_{n'} \in \Upsilon : n \neq n'} d[x_n, x_{n'}]. \tag{2}$$

This way, if the input samples had been real scalars and the distance measure were the absolute value of their difference, the generalized median would coincide with the usual arithmetic median.

The main features of SOM-nv are now highlighted. To initialize the algorithm, auxiliary vectorial pointers are introduced. Indeed, the convergence of this algorithm is significantly faster and safer if the initial pointers are already two-dimensionally ordered [Kohonen and Somervuo (2002)]. In the case of proteins, these vectorial pointers can be selected as the usual 400-dimensional dipeptide histogram vectors [Ferrán and Ferrara (1991)]. Thus, each map node is provided with a 400-dimensional vector, each component of which is initialized with a random value between zero and unity— the whole vector is finally normalized to unit length. The standard SOM-batch algorithm is then trained with the dipeptide vectors, and the final pointers obtained are recoded to get nonvectorial SOM initialized. Specifically, for each vectorial pointer the usual subset of input items (including all items having that pointer as winner in the vectorial sense) is associated to it, and the corresponding nonvectorial pointer is chosen as the generalized median of that subset. With this labelling, a 2D set of relatively ordered input sequences is achieved, so that the nonvectorial SOM can proceed. From this point on, all vectorial representations are dropped. This initialization method for SOM-nv is summarized in Algorithm 2.

For each pointer z_i, two sets are then defined. First, one would recollect in Z_i the input items associated to it, i.e., the input items that have z_i as its best-matching unit. Winning neurons could be determined as usual according to the FASTA method, but note that an input item could then have exactly the same distance to two or more pointers. Therefore, in order to make the winner unique in this case, one would ask the winner to minimize the

Algorithm 1
The nonvectorial version of SOM algorithm (SOM-nv).

Initialize the map (see below).
repeat for each iteration, t,
  for each input sequence, x_n, **do**
    Find the best matching unit for x_n, see Equation 1.
  end for.
  Recollect in Z_i (see Equation 4) the input items associated to pointer z_i.
  Store in Ω_i (see Equation 5) the items associated to each pointer in its neighbourhood N_i.
  Update each z_i as the generalized median (see Equation 2) of Ω_i.
until $Z_i^t = Z_i^{t-1}$, $\forall i$.

Algorithm 2
Initialization method for SOM-nv in the case of proteins.

Convert input sequences into 400-dimensional dipeptide histogram vectors.
Provide each map neuron with a 400-dimensional vector.
repeat,
  Train a SOM-batch cycle,
until neurons are 2D ordered.
Label neurons by those proteins that represent the generalized medians of the sequences associated to them.

sum of distances from the input to all pointers in a small neighbourhood around the winner candidate i, say N_i. This neighbourhood includes all pointers within a certain radius from node i on the grid. Like in the traditional SOM, this radius can shrink monotonically with time. Mathematically, $x_n \in Z_i$ if and only if

$$z_i = \arg\min_l \sum_{k \in N_l} d[x_n, z_k]. \tag{4}$$

Recollect now in Ω_i the input items associated to each pointer in N_i in the previous sense, that is,

$$\Omega_i = \bigcup_{k \in N_i} Z_k, \tag{5}$$

and update each z_i as the generalized median of Ω_i. Thus, this is called the adaptation process. For each new pointer z_i, recollect in Z_i the new input items associated to it as before. If the old Z_i, say Z_i^{t-1}, coincide with the new Z_i^t for all i, then the process has converged. If not, continue with the adaptation process. When convergence is reached, pointers approximate the input items in an orderly fashion, since each pointer coincides with the generalized median of the input items mapped onto its neighbourhood.

Algorithm 3
UDL monitoring scheme for nonvectorial SOM. Initialization.

Convert input sequences into an auxiliar vectorial representation.
Provide each map neuron with the same vectorial representation.
repeat,
Train the map with the vectorial SOM-batch and a constant neighbourhood range, say,

$$\sigma_\Lambda(t) = \sigma_\Lambda(0) \tag{3}$$

until neurons are 2D ordered.
Store the obtained disentangled lattice, say Q^0.
Label neurons by those proteins that represent the generalized medians (see Equation 2) of the sequences associated to them.

In this context, El Golli et al. (2004a,b) have proposed an extension of the standard Kohonen learning rule that can also handle symbolic data. Specifically, they have presented an adaptation of the SOM-batch to dissimilarity data. As in Kohonen and Somervuo (2002) work, the main difference with traditional SOM is that El Golli et al. (2004a) are not working on $\mathbb{R}^d$ but on an arbitrary set on which a dissimilarity is defined. The experiments in El Golli et al. (2004a,b) show the usefulness of their method applied to symbolic data.

5 UDL monitoring scheme for nonvectorial SOM

This section shows how the novel UDL monitoring scheme can also be applied to nonvectorial algorithms, such as SOM-nv. The UDL-monitored algorithms presented in this section have been successfully tested on the data sets considered in Muruzábal and Vegas-Azcárate (2005).

Like in traditional self-organizing maps for vectorial data, the radius of the neighbourhood function at the beginning of the process may be selected as fairly large and put to shrink monotonically in further iterations. As Kohonen and Somervuo (2002) pointed out, the optimal speed of shrinking should be experimentally determined. UDL stopping policy estimates the neighbourhood range function during the training of SOM-nv automatically, see Algorithms 3, 4 and 5.

6 Discussion

The visualization of large protein and DNA databases in a compact way may give insights into the data, leading to the development of new ideas and theories. Since the number of known DNA and proteins sequences is growing exponentially as a result of Genome projects, the management of the resulting databases is of central interest in modern Bioinformatics analysis. Many powerful algorithms for comparing two [Needleman and Wunsch (1970); Smith and Waterman (1981)] or more proteins [Waterman (1984); Corpet (1988); Higgins (1994);

Algorithm 4
UDL monitoring scheme for nonvectorial SOM. First run.

Select Q^0 as the initial neurons configuration.
repeat for each iteration, t,
  for each input sequence, x_n, **do**
    Find the best matching unit for x_n,
  end for.
  Select the neighbourhood range function to be

$$\sigma_\Lambda(t) = \sigma_\Lambda(0) \cdot \exp\left(-2 \cdot \sigma_\Lambda(0) \cdot \frac{t}{T^1} \cdot \gamma^1\right), \tag{6}$$

  where $\gamma^1 = 1$ and $T^1 = \#neurons$.
  Recollect in Z_i^1 (see Equation 4) the input items associated to pointer z_i.
  Store in Ω_i^1 (see Equation 5) the items associated to each pointer in neighbourhood N_i^1.
  Update each z_i as the generalized median of Ω_i^1.
  Obtain the dataloads and store their standard deviations in $SD^1(t)$.
until $t = T^1$.
Determine the number of epochs, t_{udl}^1, and the corresponding range, $\sigma_{\Lambda,udl}^1$, for which the speed of decrease of SD^1 function is nearly zero.

Mahabhashyam et al. (2005)] have been developed. Although these methods are sensible, they are extremely time consuming. Faster but less precise algorithms for searching homologies have been proposed [Wilbur and Lipman (1983); Lipman and Pearson (1985); Altschul and Lipman (1990); Altschul et al. (1990)]. In this way, a variety of neural networks have been used to organize protein sequences into clusters or families according to their sequence homologies. However, since the number and composition of the families are not known, the use of unsupervised learning algorithms, such as the SOM type algorithms, seems indeed very appropriate. The corresponding topological maps so obtained should be very useful in organizing large protein or DNA databases and for rapidly classifying new sequences.

In contrast to earlier works, the extension of the SOM batch allows for the use of any similarity measure in sequences. The combination of the nonvectorial topographic maps with the previously presented UDL monitoring ideas is expected to be a helpful tool to deal with biological sequences.

Acknowledgements

The author is grateful to Drs. Jorge Muruzábal, Marc M. Van Hulle and Temujin Gautama for their constructive comments on this paper.

Algorithm 5
UDL monitoring scheme for nonvectorial SOM. Monitoring runs.

repeat for each monitoring run, $j > 1$,
  Select Q^0 as the initial neurons configuration.
  repeat for each iteration, t,
    for each input sequence, x_n, **do**
      Find the best matching unit for x_n,
    end for
    Select the neighbourhood range function to be

$$\sigma_\Lambda(t) = \sigma_\Lambda(0) \cdot \exp\left(-2 \cdot \sigma_\Lambda(0) \cdot \frac{t}{T^j} \cdot \gamma^j\right), \tag{7}$$

    where $T^j = 2 \cdot t_{udl}^{j-1}$ and

$$\gamma^j = -\frac{\ln \frac{0.9 \cdot \sigma_{\Lambda,udl}^{j-1}}{\sigma_{\Lambda(0)}}}{2 \cdot \sigma_{\Lambda(0)}}. \tag{8}$$

    Recollect in Z_i^j the input items associated to pointer z_i.
    Store in Ω_i^j the items associated to each pointer in N_i^j.
    Update each z_i as the generalized median of Ω_i^j.
    Obtain the dataloads and store its standard deviation in $SD^j(t)$.
  until $t = T^j$.
  Determine the epochs, t_{udl}^j, and the range, $\sigma_{\Lambda,udl}^j$, for which the speed of decrease of SD^j function is nearly zero.
until $\sigma_{\Lambda,udl}^j \simeq \sigma_{\Lambda,udl}^{j-1}$.

Références

Altschul, S., Gish, W., Miller, W., Myers, E. and Lipman, D. (1990). Basic local alignment search tool, *J. Mol. Biol.*, **215** : 403–410.

Altschul, S. and Lipman, D. (1990). Protein database searches for multiple alignments, *Proc. Natl. Acad. Sci. USA*, **87** : 5509–5513.

Andrade, M. A., Casari, G., Sander, C. and Valencia, A. (1996). Classification of protein families and detection of the determinant residues with a self-organizing neural network.

Andreassen, H., Bohr, H., Bohr, J., Brunak, S., Bugge, T., Cotterill, R., Jacobsen, C., Kusk, P., Lautrup, B., Petersen, S., Sæ rmark, T. and Ulrich, K. (1990). Analysis of the secondary structure of the human immunodeficiency virus (HIV) proteins p17, gp120 and gp41 by computer modeling based on neural network methods, *J. Acquired Immun. Defic. Syndrome*, **3** : 615–622.

Arrigo, P., Giuliano, F., Scalia, F., Rapallo, A. and Damiani, G. (1991). Identification of a

new motif on nucleic acid sequence data using Kohonen's self-organizing map, *Comput. Appl. Biosci.*, **7** : 353–357.

Auray, J., Duru, G. and Zighed, A. (1990). *Analyse des données multidimensionnelles : les méthodes de structuration*, Lyon : Alexandre Lacassagne.

Baldi, P. (2001). *The Shattered Self : The End of Natural Evolution*, Publisher : MIT Press.

Baldi, P. and Brunak, S. (2001). *Bioinformatics : the Machine Learning Approach, 2nd edition*, MIT Press.

Baldi, P. and Hatfield, G. (2002). DNA *Microarrays and Gene Regulation*, Cambridge University Press.

Bengio, Y. and Pouliot, Y. (1990). Efficient recognition of immunoglobulin domains from amino acid sequence using a neural network, *Comput. Appl. Biosci.*, **6** : 319–324.

Blackmore, J. and Miikkulainen, R. (1992). Incremental grid growing : encoding high-dimensional structure into a two-dimensional feature map, *University of Texas, Austin, TX*, **AI** : 92–192.

Bohr, H., Bohr, J., Brunak, S., Cotterill, R., Lautrup, B., Norskov, L., Olsen, O. and Petersen, S. (1988). Protein secondary structure and homology by neural networks. the α-helices in rhodopsin, **241** : 223–228.

Brunak, S., Engelbrecht, J. and Knudsen, S. (1990). Neural network detects errors in the assignment of mRNA splice site, *Nucleic Acids Res.*, **18** : 4797–4801.

Brunak, S., Engelbrecht, J. and Knudsen, S. (1991). Prediction of human mRNA donor and acceptor site from the DNA sequence, *J. Mol. Biol.*, **220** : 49–65.

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering, *Nucleic Acids Res*, **16** : 10881–10890.

Demeler, B. and Zhou, G. (1991). Neural network optimization for E. Coli promoter prediction, *Nucleic Acids Res*, **19** : 1593–1599.

Devereux, J., Haeberli, P. and Smithies, O. (1984). A comprehensive set of sequence-analysis programs for VAX, *Nucleic Acids Research*, **12(1)** : 387–395.

Dopazo, J. and Carazo, J. (1997). Phylogenetic reconstruction using an unsupervised growing neural netrwork that adopts the topology of a phylogenetic tree, *Journal of Molecular Evloution*, **44** : 226–233.

El Golli, A., Conan-Guez, B. and Rossi, F. (2004a). Self organizing map and symbolic data, *Journal of Symbolic Data Analysis*, **2**.

El Golli, A., Conan-Guez, B. and Rossi, F. (2004b). A self organizing map for dissimilarity data, *Classification, Clustering, and Data Mining Applications (Proceedings of IFCS 2004)*, pp. 61–68.

Engelbrecht, J., Knudsen, S. and Brunak, S. (1992). GC-rich tract in 5' end of human introns, *J. Mol. Biol.*, **227** : 108–113.

Farber, R., Lapedes, A. and Sirotkin, K. (1992). Determination of eukaryotic protein coding regions using neural networks and information theory, *J. Mol. Biol.*, **226** : 471–479.

Ferrán, E. A. and Ferrara, P. (1991). Topological maps of protein sequences, *Biological Cybernetics*, **65** : 451–458.

Ferrán, E. A. and Ferrara, P. (1992). Clustering proteins into families using artificial neural networks, *Cambios*, **8(1)** : 39–44.

Ferrán, E. A., Pflugfelder, B. and Ferrara, P. (1994). Self-organized neural maps of human protein sequences, *Protein Science*, **3** : 507–521.

Frishman, D. and Argos, P. (1992). Recognition of distantly related protein sequences using conserved motifs and neural networks., *J Mol Biol*, **228** : 951962.

Fritzke, B. (1994). Growing cell structures : a self organizing network for unsupervised and supervised learning, *Neural Networks*, **7** : 1141–1160.

Hanke, J. and Reich, J. (1996). Kohonen map as a visualization tool for the analysis of protein sequences : multiple alignments, domains and segments of secondary structures, *Comput. Appl. Biosci.*, **12(6)** : 447–454.

Higgins, D. (1994). CLUSTAL V : multiple alignment of DNA and protein sequences., *Methods Mol Biol.*, **24** : 307–318.

Hirst, J. and Sternberg (1992). Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural network, *Biochemistry*, **31** : 7211–7218.

Hirst, J. and Sternberg, M. (1991). Prediction of atp-binding motifs : A comparison of a perceptron-type neural networks and a consensus sequence method, *Protein Eng.*, **4** : 615–623.

Holbrook, S., Muskal, S. and Kim, S. (1990). Predicting surface exposure of amino acids from protein sequence, *Protein Eng*, **3** : 659–665.

Holley, L. and Karplus, M. (1989). Protein secondary structure prediction with a neural network, *Proc. Natl. Acad. Sci. USA*, **86** : 152–156.

Horton, P. and Kanehisa, M. (1992). An assessment of neural network and statistical approaches for prediction of E. Coli promoter sites, *Nucleic Acids Res.*, **20** : 4331–4338.

Kangas, J. A., Kohonen, T. K. and Laaksonen, J. T. (1990). Variants of self-organizing maps, *IEEE Trans. Neural Networks*, **1** : 93–99.

Kneller, D., Cohen, F. and Langridge, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network, *J. Mol. Biol.*, **214** : 171–182.

Kohonen, T. (1985). Median strings, *Pattern Recognition Letters*, **3** : 309–313.

Kohonen, T. (1995). *Self-Organizing Maps*, Berlin : Springer-Verlag.

Kohonen, T. (1996). Self-organizing maps of symbol strings, in *Technical Report A42*, Laboratory of Computer and Information Science, Helsinki University of Technology, Finland.

Kohonen, T. and Somervuo, P. (2002). How to make large self-organizing maps for nonvectorial data, *Neural Networks*, **15** : 945–952.

Ladunga, I., Czakó, F., Csabai, I. and Geszti, T. (1991). Improving signal peptide prediction accuracy by simulated neural networks, *Comput. Appl. Biosci.*, **7** : 485–487.

Lapedes, A., Barnes, C., Burks, C., Farber, R. and Sirotkin, K. (1990). Application of neural networks and other machine learning algorithms to DNA sequence analysis, *Computers and DNA. SFI studies in the science of complexity*, **VII** : 157–182.

Lipman, D. and Pearson, W. (1985). Rapid and sensitive protein similarity searches, *Science*, **227** : 1435–1441.

Lukashin, A., Anshelevich, V., Amirikyan, B., A.I., G. and Frank-Kamenetskii, M. (1989). Neural network models for promoter recognition, *J. Biomol Struct. Dyn.*, **6** : 1123–1133.

Maddox, J. (1992). Ever-longer sequences in prospect, *Nature*, **13** : 357–370.

Mahabhashyam, C., Brudno, M. and Batzoglou, S. (2005). PROBCONS : Probabilistic consistency-based multiple sequence alignment., *Genome Research*, **15** : 330–340.

Martinetz, T. and Schulten, K. (1991). A Neural-Gas network learns topologies, in *In Artificial Neural Networks*, edited by T. Kohonen, K. Mäkisara, O. S. and (Eds.), J. K., pp. 397–402, Amsterdam : North-Holland.

McGregor, M., Flores, T. and Sternberg, M. (1989). Prediction of b-turns in proteins using neural networks, *Protein Engineering*, **2** : 521–526.

Muruzábal, J. and Vegas-Azcárate, S. (2005). On equiprobabilistic maps and plausible density estimation, in *5th Workshop On Self-Organizing Maps, Paris*.

Muskal, S., Holbrook, S. and Kim, S. (1990). Predicting of the disulfidebonding state of cystein in proteins, *Protein Eng.*, **3** : 667–672.

Muskal, S. and Kim, S. (1992). Predicting protein secondary structure content. a tandem neural network approach, *J. Mol. Biol.*, **225** : 713–727.

Nakayama, S., Shigezumi, S. and Yoshida, M. (1988). Method for clustering proteins by use of all possible pairs of amino acids as structural descriptors, *J. Chem. Inf. Comput. Sci.*, **28** : 72–78.

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

O'Neill, M. (1991). Training back-propagation neural networks to define and detect DNA-binding sites, *Nucleic Acids Res.*, **19** : 313–318.

O'Neill, M. (1992). Escherichia Coli promoters : Neural networks developed distinct descriptions in learning to search for promoters of different spacing classes, *Nucleic Acids Res.*, **20** : 3471–3477.

Pearson, W. (1999). The FASTA program package, `ftp ://ftp.virginia.edu/pub/fasta`.

Pearson, W. and Lipman, D. (1988). Improved tools for biological sequence comparison, in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 85, pp. 2444–2448.

Petersen, S., Bohr, H., Bohr, J., Brunak, S., Cotterill, R., Fredholm, H. and Lautrup, B. (1990). Training neural networks to analyse biological sequences, *Trends. Biotechnol.*, **8** : 304–308.

Quian, N. and Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models, *J. Mol. Biol.*, **202** : 865–884.

Rodrigues, J. and Almeida, L. (1990). Improving the learning speed in topological maps of patterns, in *Proc. of INNC, Paris*, pp. 813–816.

Rose, V., Croall, I. and MacFie, H. (1991). An application of unsupervised neural network methodology (Kohonen topology-preserving mapping) to QSAR analysis, *Quant. Struct. Act. Relat.*, **10** : 6–15.

Rost, B. and Sander, C. (1993a). Improved prediction of protein secondary structure by use of sequence profiles and neural networks, *Proc. Natl. Acad. Sci. USA*, **90** : 7558–7562.

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

Smith, T. and Waterman, M. (1981). Comparison of biosequences, *Advances in Applied Mathematics*, **2** : 482–489.

Snyder, E. and Stormo, G. (1993). Identification of coding regions in genomic DNA sequences : An application of dynamic programming and neural networks, *Nucleic Acids Res.*, **21** : 607–613.

Somervuo, P. and Kohonen, T. (2000). Clustering and visualization of large sequence databases by mean of an extension of the self-organizing map, in *Proceedings of the Discovery Science*, edited by Arikawa, S. and Morishita, S., pp. 76–85, Berlin : Springer.

Stolorz, P., Lapedes, A. and Xia, Y. (1992). Predicting protein secondary structure using neural net and statistical methods, *J. Mol. Biol.*, **225** : 363–377.

Stormo, G., Schneider, T., Gold, L. and Ehrenfeucht, A. (1982). Use of the perceptron algorithm to distinguish translational initiation sites in E. Coli, *Nucleic Acids Res.*, **10** : 2997–3011.

Uberbacher, E. and Mural, R. (1991). Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach, *Proc. Natl. Acad. Sci. USA*, **88** : 11261–11265.

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

Vieth, M. and Kolinski, A. (1991). Prediction of protein secondary structure by an enhanced neural network, *Acta Biochim. Pol.*, **38** : 335–351.

Von Heijne, G. (1991). Computer analysis of DNA and protein sequences, *Eur. J. Biochem.*, **199** : 253–256.

Wade, R., Bohr, H. and Wolyness, P. (1992). Prediction of water binding sites on proteins by neural networks, *J. Am. Chem. Soc.*, **114** : 8284–8285.

Waterman, M. S. (1984). General methods of sequence comparison, *Bull. Math. Biol.*, **46** : 473–500.

Watson, J. (1990). The human genome project : Past, present and future, *Science*, **248(4951)** : 44–49.

Wilbur, W. and Lipman, D. (1983). Rapid similarity searches of nucleic acid and protein data banks, *Proc. Natl. Acad. Sci. U.S.A.*, **80** : 726–730.

Wu, C., Whitson, G., McLarty, J., Ermongkonchai, A. and Chang, T. (1992). Protein classification artificial neural system, *Protein Science*, **1** : 667–677.

Xu, L. (1990). Adding learning expectation into the learning procedure of self-organizing maps, *Int. Journal of Neural Systems*, **1** : 269–283.

Zhang, X., Mesirov, J. and Waltz, D. (1992). Hybrid system for protein secondary structure prediction, *J. Mol. Biol.*, **225** : 1049–1063.