Neural Network Method To Analyze Data Compression in DNA and RNA Sequences

T. Alvager,*,[†] G. Graham,[‡] D. Hutchison,[‡] and J. Westgard[†]

Departments of Physics and Mathematics and Computer Science, Indiana State University,
Terre Haute, Indiana 47809

Received July 23, 1996[⊗]

Neural network computations on RNA sequences are used to demonstrate that data compression is possible in these sequences. The result implies that a certain discrimination should be achievable between structured vs random regions. The technique is illustrated by computing the compressibility of short RNA sequences such as tRNA. The method should be valuable in measuring the information content of DNA, including noncoding DNA, which has been shown to display certain properties resembling natural language attributes.

INTRODUCTION

In the analysis of DNA and RNA there is often a need to determine the degree of randomness of the sequences. This is especially true for the noncoding regions of DNA. For instance, it is of interest to know how much of the segment between two consecutive coding sections of DNA is random in nature. On a broader scale it would be of importance to know how much of the total DNA is random.

In this work we discuss a method to analyze the nonrandom vs random regions of DNA and RNA using data compression techniques. A random sequence, for instance, would be incompressible, while structured sequences could show varying degrees of compressibility.[1] Randomness is therefore expressed in terms of compressibility.

Data compression is a well-developed topic with a large variety of applications in fields such as computer science and image analysis.[2] The methods used are often specialized to particular situations and not easily transferable to other applications. One technique, however, that is general in nature is the neural network method, which is applied in this work. A special recurrent network is applied in which the learning rule is to construct in the hidden layer an internal representation of the data presented at the input layer. Data compression is obtained if the number of units in the hidden layer is smaller than the number of units in the input layer.

THE NEURAL NETWORK METHOD

Neural networks in a variety of structures have been used for analysis in a large number of circumstances.[3−5] For study of language patterns a useful neural network architecture seems to be the recurrent network.[3] Such a system operates by letting each input pattern pass through the network more than once before it generates an output pattern.

In the present investigation a recurrent network has been applied and used on a Macintosh computer, Quadra 850 model. This network is commercially available in the NeuralWare software package (NeuralWorks Professional 11/Plus)[5] under the name recirculating network.[6] Figure 1 shows a simplified diagram of the general architecture of this network. It consists of three layers: input, hidden, and output, respectively. The network uses unsupervised learning. Thus no data are required to be presented at the output layer.

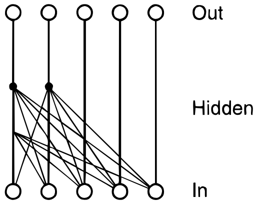

Figure 1. Simplified diagram of a recirculation network in the NeuralWare implementation.[5] Five input units and two units in the hidden layer are shown. Data for training of the system are presented at the input layer and filtered to the hidden layer. The processed data are then recirculated back and filtered to the input level. Finally the data are sent for a second time to the hidden layer through a third set of variable weight factors. Learning occurs after the second pass through the network in accordance with the result from the recirculation. The difference between the original inputs and the outputs from the units in the hidden layer after the second pass through the network is referred to as the reconstruction error (cf. Figure 2).

The units in the input and hidden layers are fully connected in both directions. When data for training of the system are presented at the input layer, they are first filtered to the hidden layer through the use of a set of constant weight factors. The processed data are then recirculated back and filtered to the input level through a second set of constant weight factors. Finally the data are sent for a second time to the hidden layer through a third set of (variable) weight factors. Learning occurs after the second pass through the network in accordance with the result from the recirculation. The constant weight factors applied were the values specified by the manufacturer of the software packages to obtain maximum stability of the calculations. The variable weights were also computed following the procedures recommended by the manufacturer and essentially obtained through methods in use for network computations. This complete process constitutes one cycling period of computations. In general a large number of cycling periods are needed for a satisfactory result as discussed in the next paragraph.

A measure of the result from the calculations is given by the difference between the original inputs and the outputs from the units in the hidden layer after the second pass through the network. This difference is referred to as the reconstruction error in the implementation used in this work.[5] Clearly the goal of the learning procedure is to reduce this

[†] Department of Physics.
[‡] Department of Mathematics and Computer Science.
[⊗] Abstract published in *Advance ACS Abstracts,* February 1, 1997.

S0095-2338(96)00119-9 CCC: $14.00

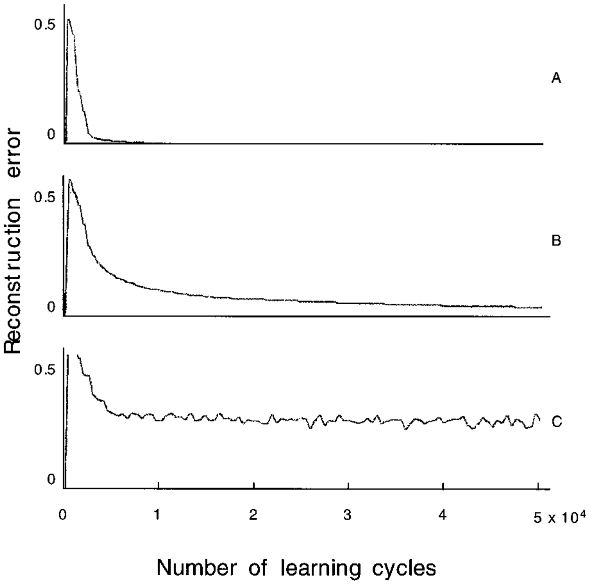

Figure 2. Calculated reconstruction error plotted as a function of the number of learning cycles for the three sequences listed in Table 1: (A) item 1, (B) item 2, and (C) item 3. There were nine input units and four units in the hidden layer.

Table 1. Three RNA Sequences Used in the Evaluation of the Compression Method Tests

item	description
1	AGAAAUAUUUCU plus repeats of this set, a total of 64 bases
2	the molecule tRNA$^{\mathrm{Trp}}$ (human),[8] 71 bases
3	sequence consisting of a listing of all the codons,[7] 64 × 3 bases

error. It can often be achieved by changing the values of the weight factors of the variable type in such a way that the error is reduced. In general a large number of passes of processed data is needed to reduce the reconstruction error to an acceptable level. Figure 2 shows graphs for some input data to be discussed in detail in the next section. The calculated reconstruction error is plotted as a function of the number of cycling periods.

In the learning process an internal representation of the data at the input layer is created in the hidden layer. If fewer elements are needed in this layer than at the input level to obtain a satisfactory result, a compression of the data has been achieved.

RESULTS

Three RNA sequences have been analyzed. They are given in Table 1. The standard nomenclature has been used to represent these sequences, i.e., U, C, A, and G stand for the bases uracil, cytosine, adenine, and guanine, respectively.[7] The combination listed as "item 1" is part of one of the fixed arms of the molecule tRNA. This sequence represents a highly ordered structure. Item 2 is the tRNA that transports the amino acid tryptophan in humans.[8] This tRNA is built up from 71 bases with part of it consisting of the combination listed under item 1. The tRNA molecule represents a combination of randomness and ordered structure. Item 3 is the list of the 64 codons such as UUU. These codons represent a random sequence.

Figure 2 shows graphs for the three sequences listed in Table 1. In all cases the number of hidden units was four and the number of input units was nine, corresponding to three bases with the values: U = 001, C = 010, A = 011, and G = 100. Thus, for instance, the combination UCA is 001010011.

Table 2. Reconstruction Error as a Function of the Number of Units in the Hidden Layer for the Three Items Defined in Table 1[a]

	no. of units in hidden layer						
item	7	6	5	4	3	2	1
1	0	0	0	0	0	0	0.21
2	0	0	0.02	0.04	0.18	0.5	>0.5
3	0	0.03	010	0.25	0.45	>0.5	>0.5

[a] The values are calculated numbers obtained after the curves have reached a constant level. There were nine input units. Compare the curves in Figure 2.

In Figure 2 the calculated reconstruction error is plotted as a function of the number of learning cycling periods. The protocol is as follows: The program selects in a random manner all the input values in the list and computes the reconstruction error. The procedure is then repeated a prescribed number of cycles, which in the graphed cases was set to be maximum at 50 000. The program is also capable of selecting the inputs in an orderly manner. No significant difference in the result was detected, however. As observed in Figure 2A the item 1 combination drops to zero very rapidly, while the reconstruction error for the list of all codons never reaches the zero level but stops at a value approximately equal to 0.25 (Figure 2C). The tRNA molecule is somewhere in between these two extreme cases as expected (Figure 2B).

As indicated above the objective of the calculations is to measure the compressibility of a sequence by determining, for a fixed number of input units, the number of hidden units needed to reconstruct the original input within a certain error. Thus a series of computations with varying number of hidden units was performed. Table 2 gives the result. The highly structured sequence (item 1) is clearly compressible almost by a factor of 5, since the number of inputs is nine and only two hidden units seem sufficient to reproduce the input. The least compressible sequence is clearly the list of all codons.

DISCUSSION AND CONCLUSION

It is clear from the results presented above that there is a difference between sequences of different origin. Not only is the minimum number of units in the hidden layer an index for compressibility but also the shape of the reconstruction curves can serve as an indicator of what kind of sequence is at hand.

The sequences used in this work were short. More interesting situations may be obtained in applying the method to extended DNA sequences. Such a case is the recent discovery by Mantegna et al.[9] From statistical analysis it was found that noncoding DNA seems to have nonrandomness. It was found that the rank of arbitrary, contiguous sequences in noncoding DNA and their frequencies were approximately inversely proportional. Such a relation is known to be approximately true for natural languages, a property discovered by Zipf.[10] The discovery by Mantegna et al. has renewed interest in Zipf's law and its possible significance in relation to the DNA molecule. Previous investigations seemed to indicate that the law was not satisfied for coding regions in DNA.[11] These properties of

DNA may be evaluated in an independent way using the method discussed in this paper.

The feasibility of investigating long sequences of DNA using the method discussed here depends, of course, on the calculating power available. The Macintosh computer applied in this work is only suitable for short sequences. The time involved in the computations presented above is of the order of minutes. Sequences 10^6 longer than those introduced here clearly are out of the question for a standard desk computer. However, with the computer system chip Ni1000 developed by Nestor, Inc.[12] for desk computers, the computing speed can be improved by a factor of 10^4-10^5. With such a system even the total length of human DNA (approximately 10^9 bases long) may be an attractive target for the neural network method. It should be stressed, however, that for the present method to be of importance in DNA analysis shorter sequences may be analyzed separately. Such sequences may be as those considered in this paper or somewhat larger corresponding to protein coding. The need for a computing power suitable for very long sequences is therefore not a severe restriction on the method.

REFERENCES AND NOTES

(1) Gell-Mann, M. *The Quark and the Jaguar*; W. H. Freeman: New York, 1994; Chapter 3.

(2) Rabbani, M.; Jones, P. *Digital Image Compression Techniques*; SPIE Optical Engineering Press: Bellingham, WA, 1991.

(3) Haykin, S. *Neural Network*; Macmillan College Publishing Co.: New York, 1994.

(4) Alvager, T.; Smith, T.; Vijai, F. The Use of Artificial Neural Networks in Biomedical Technologies. *Biomed. Instrum. Technol.* **1994**, *28*, 315−322.

(5) *NeuralWare, Inc.*: Technical Publications Group: Penn Center West, Pittsburgh, PA, 1993.

(6) Hinton, G.; McClelland, J. Learning Representations by Recirculation. In *Neural Information Proc. Syst.*; Anderson, D., Ed.; American Institute of Physics: New York, 1988; pp 358−366.

(7) Watson, J. D.; Hopkins, N. H.; Roberts, J. W.; Steitz, J. A.; Weiner, A. M. *Molecular Biology of the Gene*, 4th ed.; The Benjamin/Cummings: menlo Park, NJ, 1987; Vol. 1, Chapter 15, p 431.

(8) Anderson, S.; Bankier, A. T.; Barrell, B. G.; de Bruijn, M.; Coulson, A.; Drouin, J. Sequence and Organization for the Human Mitochondrial Genome. *Nature* **1981**, *290*, 457−65.

(9) Mantegna, R. N.; Buldyrev, S. V.; Goldberger, A. L.; Havlin, S.; Peng, C. K.; Simons, M.; Stanley, H. E. Linguistic Feature of Noncoding DNA Sequences. *Phys. Rev. Lett.* **1994**, *73*, 3169−3171.

(10) Zipf, G. K. *Human Behavior and the Principle of Least Effort*; Addison-Wesley Press: Cambridge, MA, 1949.

(11) Borodovsky, M.; Gusein-Zade, S. A General Rule for Ranged Series of Codon Frequencies in Different Genomes. *J. Biomol. Struct. Dyn.* **1989**, *6*, 1001−1012.

(12) Ni1000 Development System, 1996. Nestor, Inc., One Richmond Square, Providence, RI.

CI960119K