EJB Electronic Journal of Biotechnology ISSN: 0717-3458 Vol.1 No.3, Issue of December 15, 1998.

Received 26 August 1998 / Accepted 9 October 1998

RESEARCH ARTICLE

Using a neural network to backtranslate amino acid sequences

Gilbert White
Department of Biological Sciences
Clark Atlanta University
223 James Brawley Dr., S.W.
Atlanta, GA 30314 USA

William Seffens[1]
Department of Biological Sciences and Center for Theoretical Study of Physical Systems
Clark Atlanta University
223 James Brawley Dr., S.W.
Atlanta, GA 30314 USA
Tel: 404-880-6822 (USA) Fax: 404-880-6756 (USA)
E-mail: wseffens@cau.edu

http://www.cau.edu

A neural network (NN) was trained on amino and nucleic acid sequences to test the NN's ability to predict a nucleic acid sequence given only an amino acid sequence. A multi-layer backpropagation network of one hidden layer with 5 to 9 neurons was used. Different network configurations were used with varying numbers of input neurons to represent amino acids, while a constant representation was used for the output layer representing nucleic acids. In the best-trained network, 93% of the overall bases, 85% of the degenerate bases, and 100% of the fixed bases were correctly predicted from randomly selected test sequences. The training set was composed of 60 human sequences in a window of 10 to 25 codons at the coding sequence start site. Different NN configurations involving the encoding of amino acids under increasing window sizes were evaluated to predict the behavior of the NN with a significantly larger training set. This genetic data analysis effort will assist in understanding human gene structure. Benefits include computational tools that could predict more reliably the backtranslation of amino acid sequences useful for Degenerate PCR cloning, and may assist the identification of human gene coding sequences (CDS) from open reading frames in DNA databases.

Degenerate primers or probes, usually designed from partially sequenced peptides or conserved regions on the basis of comparison of several proteins, have been widely used in the polymerase chain reaction (PCR), DNA library screening, or Southern blot analysis. The degenerate nature of the genetic code prevents backtranslation of amino acids into codons with certainty. Numerous statistical studies have established that codon frequencies are not random (Karlin and Brendel, 1993). Many cDNA sequences have been mapped onto a "DNA-walk" and long-range power law correlations were found (Peng et.al., 1992). In consideration of the long-range correlations in DNA, a neural network approach may identify sequence patterns in coding regions that could be used to improve the accuracy of backtranslation.

Neural networks are able to form generalizations and can identify patterns with noisy data sets. To list just a few biological applications, neural networks have been used successfully to identify coding regions in genomic DNA (Snyder and Stormo, 1993), to detect mRNA splice sites (Ogura et. al., 1997), and to predict the secondary structure of proteins (Holley and Karplus, 1989, Chandonia and Karplus, 1996). Neural networks have also been used to study the structure of the genetic code. One such network was trained to classify the 61 nucleotide triplets of the genetic code into 20 amino acid categories (Tolstrup et.al., 1994). This network was able to correlate the structure of the genetic code to measures of amino acid hydrophobicity. Most neural network methods for identifying patterns in sequences can be classified as a search by signal or a search by content (Granjeon and Tarroux, 1995). Search by signal consists in identifying specific sites, such as splice sites. This method suffers from a lack of reliability when variable signals delimit the regions of interest. Search-by-content algorithms use local constraints, such as compositional bias, to characterize regions of DNA. The goal of the research reported here is to utilize the successful NN techniques to analyze and generalize codon usage in mRNA sequences beginning at the CDS start site. Local and global patterns of codon usage in genes may be identifiable by neural

[1] **Corresponding author**

networks of suitable architecture. This paper reports on some initial trials of altering the encoding of amino acids for the input neural layer. Future studies will address the architecture of the hidden layer to optimize for the NN ability to detect codon usage patterns in genes.

Materials and Methods

Training set. Human mRNA sequences were obtained from GenBank on the basis of several criteria. The coding sequences were relatively short in order to avoid splicing and other variants of the mRNA. The sequences were identified by keywords that would indicate a complete mRNA could be reconstructed. Such words would be complete coding sequence (CDS), 5' and 3' untranslated regions (UTR), and poly(A) site. Multiple members from gene families were excluded to prevent overtraining of those sequences. The sequences were downloaded from Entrez at the NIH web site (http://www.ncbi.nlm.nih.gov/Entrez/) and the coding sequence was saved from each into a file. Up to the first 75 nucleotides of the CDS were selected for this study in a window starting at the methionine ATG start site.

Binary representations. In order to train the neural network (NN) it is necessary to formulate a decoding scheme because the architecture of the NN is binary and does not allow a direct representation of nucleic or amino acid sequences. Therefore, a binary numeric representation was used to encode the amino acid data. Several Microsoft Word 97 macros were recorded to convert amino acids and nucleic acids into numerical values. The macros used the find and replace commands in Microsoft Word 97 for each of the twenty amino acids and for the four nucleotides. The individual numeric-encoded sequence files were then joined together into groups. For this study a total of sixty mRNAs were examined with different window sequence lengths which changed the total size of the training set (White, 1998). The nomenclature for each group identifies the number of sequences used and the number of codons taken from each sequence. For example, in Training Set 60S-10C there are sixty sequences with a window of ten codons taken from each sequence. Since ten codons were taken from each sequence, there are 600 codons in this set. A related study of predicting bases in tRNA sequences used a window size of 15 bases (Sun et.al., 1995), while this study used a window of 10 codons or 30 bases.

Neural network. All work with the NN was performed on a Sun SPARCstation™ 20 computer. The NN used was a utility of Partek 2.0b4, called a multi-layer perceptron (MLP). A MLP is a NN, which has at least three layers (the input, output and the hidden layer(s)). Each layer is attached to the next layer by connection weights that are changed during the training process to reduce the overall error. This allows the network to "learn" patterns in the mRNA sequences. Training was stopped when the change in the total output error became less than 0.1% from the previous iteration. This usually occurred after 500 - 1200 iterations using the backpropagation learning method. Test sets were assembled to assess the predictive accuracy of the trained NN. The test sets consisted of 3 randomly selected human gene sequences from the same group of sequences from which the training set was selected. The predicted output was measured in 3 categories: the overall percent correct, percent correct for degenerate bases, and percent correct for fixed bases. These measures allow the assessment of the various schemes used to encode the amino acids.

Results

Encoding the amino acids

Different amino acid decoding schemes were examined to determine how the input configuration would affect prediction accuracy of the networks in backtranslating amino acids into nucleic acids. The simplest and most direct scheme, called "Simple", is a 20-bit representation where each amino acid is represented by a one and nineteen zeros (Figure 1). Alanine would be 10000000000000000000 and the one would shift to the right alphabetically based on the one letter abbreviation of the amino acids. Another scheme called "Simple-Shuffle" is a rearrangement or shuffling of the amino acids in the previous scheme. This is to test if the order of amino acids in the input layer is important, since the composition can be quite different between abundant and rare amino acids. This scheme uses an alphabetical listing based on their codon representations using degeneracy codes (Table 1). Therefore, Lysine with AAR is first, and Leucine with YTX is last (00000000000000000001).

Adding degeneracy information

The "simple" representation ignores the nucleic acid bases already known from the genetic code. For example, all three bases are known for Methionine (ATG). IUPAC representations utilize degeneracy codes (Table 1) to denote which possible bases can be used for a particular amino acid at the first, second, or third position of a codon. An example of this would be GGX, the degeneracy code for Glycine, where four nucleotide endings are possible. Degeneracy codes can then be utilized for the input layer similar to the multiple sensor approach taken by Uberbacher and Mural (1991). Some input neurons could then convey processed information about limited codon choices. Thus by using these degeneracy codes we come closer to the actual nucleic acid sequence that encodes the amino acid. This results in a 33-bit unit in a scheme called "All-Degeneracy" (Figure 2). This scheme has a greater number of input neurons than the simple schemes, yet the fixed part of the genetic code is effectively preprocessed for the NN. As pointed out by Lapedes et.al. (1990), there is a trade-off between putting processing power into the network versus putting it into a pre-processing stage that changes the representation of the input data. In Figure 2 the

hidden layer is not shown between the input and output layers to highlight the representation of known or non-degenerate bases to the output layer.

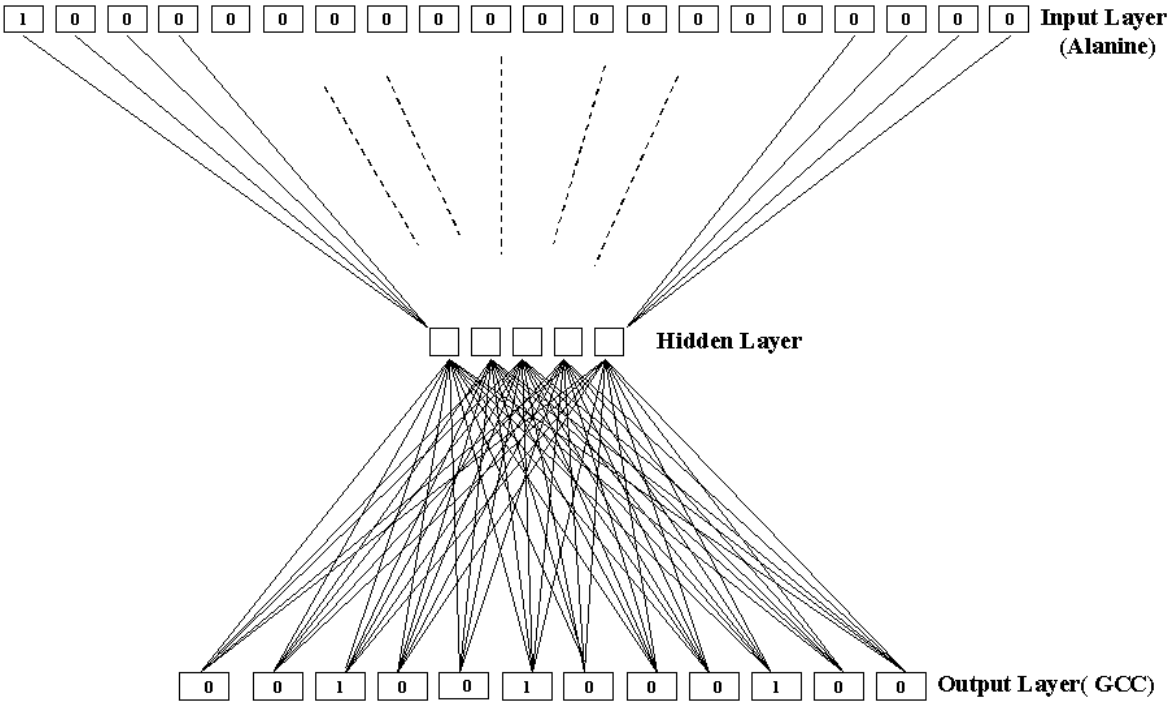

Figure 1. This decoding scheme showing one amino acid, Alanine, is the simplest representation. This scheme is called Simple. The dotted lines indicate that not all connections are drawn.

Table 1. Degeneracy codes for nucleic acids.

Code	A	C	G	T
M	*	*		
R	*		*	
S		*	*	
Y		*		*
W	*			*
H	*	*		*
X	*	*	*	*

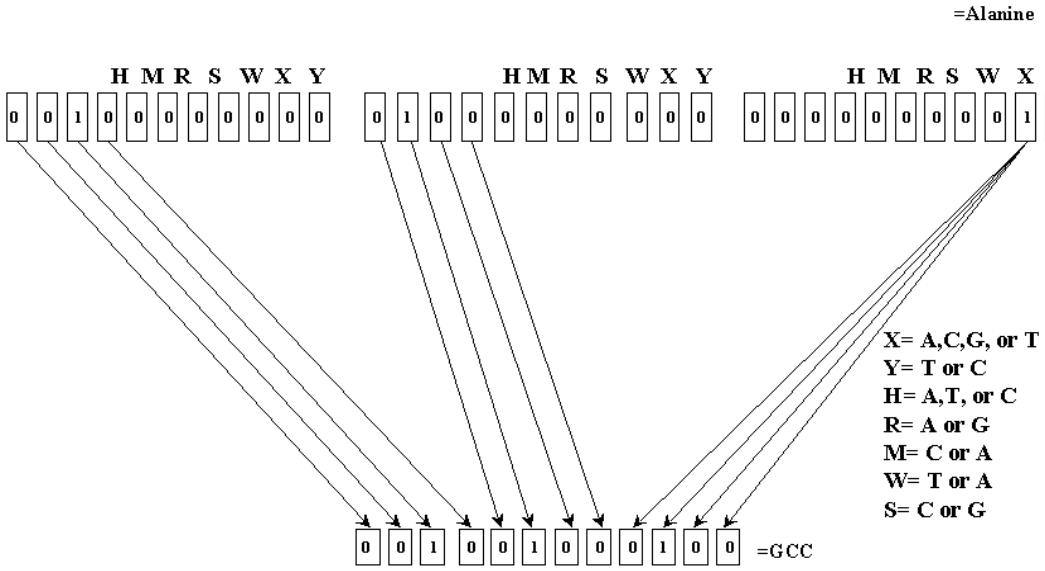

Figure 2. All-Degeneracy scheme. It uses all the degeneracy codes for the amino acids and thus comes closest to the actual nucleic acid sequence information.

Binary encoding

Another way of encoding amino acids is to form groups that are based on some ordering and to identify the amino acids within the groups. The scheme called "Binary-5-bit", is based on all the possible ways that ones and zeros can be combined in a five-bit group (Figure 3). There are 32 possible ways these numbers can be arranged. When the representations with no or all ones, and those with 1 or 4 ones are removed, there are exactly twenty representations left. This leaves just enough representations to code for the 20 amino acids. Other similar ways of grouping the amino acids were tried with results typical of the Binary-5-bit scheme (data not shown).

Comparing the schemes

These four NN schemes were used to predict the correct codons given an amino acid sequence. The percent correct in predicting degenerate bases was used to test the network's ability to backtranslate from amino acid sequences to nucleic acid sequences. The networks were trained and test sets were used to assess the accuracy for each scheme. The change in predictive accuracy of the schemes was analyzed as the window size was increased to determine which scheme or schemes would be most efficient with larger training sets. The best scheme in predicting correctly the degenerate bases was Simple, which predicted 85% of the degenerate bases in Training Set 60S-10C (Table 2). All of the schemes were predicting 100% of the fixed bases for all window sizes. The largest scheme, which has 33 input neurons per amino acid, shows a consistently better performance compared to the smallest scheme with 5 input neurons per amino acid (Table 2). There is little difference between Simple and Simple-Shuffle, so that the order of amino acids in the input layer is not important. The binary scheme for the smaller window size does not perform as well as the unitary schemes, a result also found by other researchers (O'Neill, 1991, Demeler and Zhou, 1991). However, with the largest window there is very little difference between the schemes. This may be due to more amino acids being present in the training set, allowing for a more complete representation of the genetic code. For the smaller windows not all the codons are represented in the training sets and may explain why Simple's accuracy did not exceed 85%. A codon usage table calculated from training Set 60S-10C found two codons for tyrosine and histidine missing, and one other codon was represented only once. All other codons had multiple occurrences in the 60S-10C training set. Therefore the genetic code was incompletely represented in the smaller training sets. The accuracy decreased as the window size increased for Simple, possibly due to the increased complexity or size of the input layer of the NN and the minimal increase of the hidden layer. The size of the hidden layer did not increase as fast as the input layer

for increased window sizes due to the default settings of the NN. Overall the four schemes are capable of backtranslating with high accuracy for the degenerate bases from a relatively small training set.

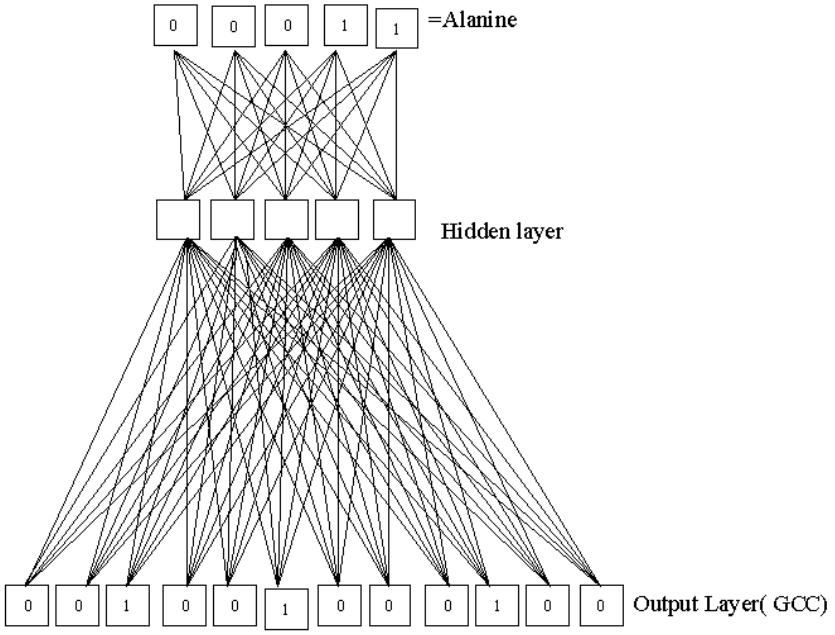

Figure 3. Binary-5-Bit scheme. It is based on all the possible ways that ones and zeros can be combined in a five bit unit. There are 32 possible ways to represent 20 amino acids. When the representations with no or all ones and those with 1 or 4 ones are removed there are exactly twenty representations.

Table 2. Accuracy of NN encoding schemes.

	Simple	Simple-Shuffle	All-Degeneracy	Binary 5-bit
Bits/amino acid	**20**	**20**	**33**	**5**
Training set 60S-10C	**85%**	**72%**	**80%**	**74%**
Training set 60S-15C	**72%**	**77%**	**77%**	**74%**
Training set 60S-20C	**80%**	**80%**	**80%**	**69%**
Training set 60S-25C	**74%**	**72%**	**74%**	**77%**

Shown are the percent of correctly predicted degenerate bases in a test set composed of three sequences selected randomly from the same group of sequences from which the training set was assembled.

Discussion

One of the possible uses of this research is to improve the design of oligonucleotide probes (Eberhardt, 1992). One primer-design study found an overall homology greater than 82% between the predicted probe and the target sequence when codon utilization and dinucleotide frequencies were taken into account (Lathe, 1985). When sequence stretches lacking Serine, Arginine, and Leucine are selected the overall homology became 85.7% in Lathe's study. Our best network predicted 85% of the degenerate bases, and 93% of the overall bases. The data set used Lathe's study contained 13,000 nucleotides and our largest training set had 4500 nucleotides. Therefore, an increase in our network or training set size could lead to even greater accuracy by detecting patterns of codon choice within the mRNA sequences. The architecture of the amino acid encoding method apparently does not have a large impact on predictive accuracy as found in this study. Therefore other factors, such as computational time or memory size may be a criteria used to select an encoding scheme for a larger training set. It is also interesting to note that the network that predicted the highest percentage of correct overall bases did so on a test set that had eight Leucines, one Arginine, and two Serines. These amino acids present difficulties for algorithms based on codon lookup tables, such as Lathe's work or common primer selection programs (such as Nash, 1993). The work reported here demonstrates that a NN approach may yield improvements in predictive accuracy for PCR primer selection.

Financial Support

This work was supported (or partially supported) by NIH grant GM08247, Research Centers in Minority Institutions award G12RR03062 from the Division of Research Resources, National Institutes of Health and NSF CREST Center for Theoretical Studies of Physical Systems (CTSPS) Cooperative Agreement #HRD-9632844.

References

Chandonia, J. and Karplus, M. (1996) The importance of larger data sets for protein secondary structure prediction with neural networks. Protein Science 5:768-774.

Demeler, B. and Zhou, G. (1991). Neural Network Optimization for E. Coli Promoter Prediction. Nucleic Acid Research 19:1593-1599.

Eberhardt, N. (1992) A shell program for the design of PCR primers using Genetics Computer Group software. BioTechniques 13:914-916.

Granjeon, E. and Tarroux, P. (1995) detection of compositional constraints in nucleic acids using neural networks. CABIOS 11:29-37.

Holley, L. and Karplus, M. (1989). Protein Secondary Structure Prediction with a Neural Network. Proceedings of the National Academy of Sciences USA 86:152-156.

Karlin, S. and Brendel, V. (1993) Patchiness and correlations in DNA sequences. Science 259:677-680.

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. In G. Bells and T. Marr (eds.), *Computers and DNA: SFI Studies in the Sciences of Complexity*. Addison-Wesley, Reading, MA. Vol 7, pp157-182.

Lathe, R. (1985) Synthetic Oligonucleotide Probes Deduced from Amino Acid Sequence Data.. Journal of Molecular Biology 183:1-12.

Nash, J. (1993) A computer program to calculate and design oligonucleotide primers from amino acid sequences. CABIOS 9:469-471.

Ogura, H. Agata, H., Xie, M., Odaka, T., and Furutani, H. (1997). A Study of Learning Splice Sites of DNA Sequence by Neural Networks. Comput. Biol. Med. 27:67-75.

O'Neill, M. (1991). Training Back-Propagation Neural Networks to Define and Detect DNA binding Sites. Nucleic Acid Research 19:313-318.

Peng, C., Buldyrev, S., Goldberger, A., Havlin, S., Sciortino, F., Simons, M., and Stanley, H. (1992) Long-range correlations in nucleotide sequences. Nature 356:168-170.

Snyder, E. and Stormo, G. (1993). Identification of coding Regions in Genomic DNA Sequences: an Application of Dynamic Programming and Neural Networks. Nucleic Acid Research 21:607-613.

Sun, J., Song, W.-Y., Zhu, L.-H., and Chen, R.-S. (1995) Analysis of tRNA gene sequences by neural network . J. Comp. Biol. 2:409-416.

Tolstrup, N., Toftgard, J., Englebrecht, J., and Brunak, S. (1994) Neural Network Model of the Genetic Code is Strongly Correlated to the GES Scale of Amino Acid Transfer Free Energies. Journal of Molecular Biology 243:816-820.

Uberbacher, E. and Mural, R. (1991) Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. Proceedings of the National Academy of Sciences USA 88:11261-11265.

White, G. (1998) Detection of Codon Usage Patterns for Backtranslation Using a Neural Network, Masters Thesis, Biology Department, Clark Atlanta University, Atlanta, GA.