Uses of Artificial Neural Networks in Macromolecular Structure and Function Predictions

An artificial neural network (NN) is a computer programmed model that attempts to mimic our understanding of the information processing and pattern matching that occurs in the brain. Our biological learning process centers around receiving certain input from our environment and shaping our response (the output) based upon positive and negative feedback supplied during our training. For example, we may be trained to answer the phone at certain times (such as weekends) when the person calling is more likely to be a family member or friend, or we may choose to let the answering machine pick up at another time (such as dinner time on a weeknight) when the caller is often a stranger who wants us to buy something. In such an example, our output (to either answer the phone or let our machine pick it up at some time of day) is trained based upon positive and negative feedback we might receive during our first few months living in New Haven. The same theory is applied to an artificial neural network computer program. Since our brains contain millions of neurons and connections, it is currently infeasible to achieve a similar NN architecture in the computer program. However, as has been discovered, the task of applying NNs to the problem of protein structure prediction requires many fewer artificial neurons and therefore far less total connections. The idea is to create a certain number of input "nodes" and connect each one to every node in a hidden layer. Each node in the hidden layer is then connected to every node in the final output layer. The connection strength between each and every pair of nodes is initially assigned a random value and is then modified by the program itself during the training process. Each node will "decide" to send a signal to the nodes it is connected to based on evaluating its transfer function after all of its inputs and connection weights have been summed (Figure 1). Training proceeds by holding particular data (say from an entry in the Protein Data Bank) constant onto both the input and output nodes and iterating the network in a process that modifies the connection weights until the changes made to them approach zero. When such convergence is reached, the network is ready to receive new experimental data. Now the connection weights are not changed and the values of the hidden and output nodes are calculated according to the functions in Figure 1 to achieve a prediction. Selection of unbiased and normalized training data, however, is probably just as important as the network architecture in the design of a successful NN.

A **B**

$$E_i = \sum_j w_{ij} s_j + b_i \qquad F(E_i) = \frac{1}{1 + e^{-E_i}}$$

Figure 1. Panel A: The state of each node (E_i) is calculated as the sum of the state (s) of every node serving input to E multiplied by the weight of the connection between E and s (w). A bias (b_i) for each node is also commonly added in the sum. Panel B: a typical sigmoidal transfer function for node E is then calculated after state of node E is determined as in Panel A.

NNs have been used to make a variety of predictions of macromolecular structure and function such as protein phosphorylation sites (1), intron splice boundaries (2), and the location of alpha-helices in transmembrane proteins (3, 4). Using NNs to predict protein tertiary structure has barely scratched the surface of this difficult problem, but a good start has included predicting residue contact rules in protein-protein interactions (5) and the prediction of solvent accessibility to protein surfaces (6).

The prediction of signal peptide cleavage sites in secreted/membrane proteins and protein secondary structure using NNs have clearly been the most successful pursuits to date. In fact, NNs had their debut in biology in the late 1980's to attempt protein secondary structure prediction. Qian et al. used X-ray crystal structures of globular proteins available at that time to train a NN to predict the secondary structure of non-homologous proteins (7). Since every residue in a PDB entry can be associated to one of three secondary structures (HELIX, SHEET or neither: COIL) the authors were able to design a NN that had 21 input nodes (one for each residue including a null residue) and three output nodes coding for each of the three possible secondary structure assignments (HELIX, SHEET and COIL). It was easiest to restrict the input and output nodes to binary values (1 or 0) when loading the data onto the network during training. This explains why three output nodes are needed: HELIX was coded as 0,0,1 on the three output nodes; SHEET is coded as 0,1,0 and COIL is coded as 1,0,0. A similar binary coding scheme was used for the 20 input nodes for the 20 amino acids. Since the authors wished to consider a moving window of seven residues at a time, their input layer actually contained 20 x 7 nodes plus one node at each position for a null residue. Over 100 protein structures were used to train this network. After training, when the NN was queried with new data, a prediction accuracy of 64% was obtained. The Matthews coefficients (Figure 2) have been a particularly useful means of calculating the prediction accuracy of a NN and are still used today. Perfect prediction accuracy should approach unity using this formula. These coefficients were found to be C_α = 0.41, C_β = 0.31, C_{coil} = 0.41, for alpha, beta and random coil, respectively, in these early prediction attempts of Qian et al.

$$C_\alpha = \frac{(p_\alpha * n_\alpha) - (u_\alpha * o_\alpha)}{\sqrt{(n_\alpha + u_\alpha)(n_\alpha + o_\alpha)(p_\alpha + u_\alpha)(p_\alpha + o_\alpha)}}$$

Figure 2. Matthews coefficient (in this case for alpha helix prediction) is a means to assess the performance of the prediction method. p_α is the number of positive cases that were correctly predicted by the NN, n_α is the number of negative cases that were correctly rejected, o_α is the number of false positives and u_α is the number of misses. The expression is similar for C_β and C_{coil}.

Since then, a host of labs have made several changes to the original NN design and training process, and in some cases, this resulted in an increase in protein secondary structure prediction. Holley et al. (8) designed a NN architecture having a larger sliding window size of 17 residues, less nodes in the hidden layer and fewer protein structures (n = 48) in their training set. This

group reported a secondary structure prediction accuracy that was quite similar to Qian et al. Stolorz et al. have achieved slightly better prediction accuracy by improving the training function (9). Rost et al. took advantage of the fact that a multiple sequence alignment contains more information about a protein than the primary sequence alone (10). Instead of using a single sequence as input into the network, they used a sequence profile that resulted from the multiple alignment. This resulted in an significant improvement in prediction accuracy to 71.4%. Recently, more radical changes to the design of NNs including bi-directional training and the use of the entire protein sequence as simultaneous input instead of a shifting window of fixed length has led to prediction accuracy above 75% (C_α = 0.72, C_β = 0.59, C_{coil} = 0.56) (11). Petersen et al. have reported the highest accuracy to date at 80% by combining the predictions of 800 separately trained NNs, as well as using a second NN to filter the output of the first and by increasing the number of output nodes so that three residues are predicted simultaneously (12).

The progess of secondary structure prediction using NNs has been excellent and should prove to be very useful input into future developed tertiary prediction algorithms. Another very successful use of NNs lies in the more simplistic prediction of the cleavage site in signal peptides of proteins in the secretion pathway. The process "SignalP" developed by Nielsen et al. (13, 14) uses the outputs of two NNs to predict the likely existence of a signal peptide near the beginning of a query sequence, as well as the likely site of peptide cleavage. Both NNs are trained on three separate groups of data derived from eukaryots, Gram-positive and Gram-negative eubacteria. The first network is trained to classify each residue as either belonging to a signal peptide or not (S-value) and the second network assigns a score to each residue indicating it's probability of being the cleavage site (C-value). A mathematical smoothing function then takes the S-values and C-values produced from the NNs to generate the most likely location of signal peptide cleavage. If no value is found to exceed a certain cutoff score, then the protein is not considered to contain a signal peptide. The prediction of signal peptide cleavage site, in combination with other prediction methods already developed such as intron-exon splice predictions and transmembrane regions will probably be combined in the near future and used in consort for genome wide analysis. Prediction methods for finding the AUG start codon and resulting ORFs in mRNA has been the long standing achievement of Marilyn Kozak (15). However, to my knowledge, it still remains to use her results to train a NN to automate this task, thereby adding to the tools that may someday be used to drive automated proteome projects.

Many of the NN prediction schemes described here can be found at the Center for Biological Sequence analysis http://www.cbs.dtu.dk/services.

References:

1. Blom, N., Gammeltoft, S. & Brunak, S. (1999) *Journal of Molecular Biology* **294,** 1351-1362.
2. Hebsgaard, S. M., Korning, P. G., Tolstrup, N., Engelbrecht, J., Rouze, P. & Brunak, S. (1996) *Nucleic Acids Res* **24,** 3439-52.
3. Rost, B., Casadio, R. & Fariselli, P. (1996) *Ismb* **4,** 192-200.
4. Casadio, R., Fariselli, P., Taroni, C. & Compiani, M. (1996) *Eur Biophys J* **24,** 165-78.
5. Fariselli, P. & Casadio, R. (1999) *Protein Eng* **12,** 15-21.
6. Rost, B. & Sander, C. (1994) *Proteins* **20,** 216-26.
7. Qian, N. & Sejnowski, T. J. (1988) *J Mol Biol* **202,** 865-84.
8. Holley, L. H. & Karplus, M. (1989) *Proc Natl Acad Sci U S A* **86,** 152-6.
9. Stolorz, P., Lapedes, A. & Xia, Y. (1992) *J Mol Biol* **225,** 363-77.
10. Rost, B. & Sander, C. (1993) *Proc Natl Acad Sci U S A* **90,** 7558-62.
11. Baldi, P., Brunak, S., Frasconi, P., Soda, G. & Pollastri, G. (1999) *Bioinformatics* **15,** 937-946.
12. Petersen, T. N., Lundegaard, C., Nielsen, M., Bohr, H., Bohr, J., Brunak, S., Gippert, G. P. & Lund, O. (2000) *Proteins: Structure, Function, & Genetics* **41,** 17-20.
13. Nielsen, H., Engelbrecht, J., Brunak, S. & von Heijne, G. (1997) *Protein Engineering* **10,** 1-6.
14. Nielsen, H., Engelbrecht, J., Brunak, S. & von Heijne, G. (1997) *International Journal of Neural Systems* **8,** 581-99.
15. Kozak, M. (1984) *Nucleic Acids Research* **12,** 857-72.