USING ARTIFICIAL NEURAL NETWORK TOOLS TO ANALYZE MICROBIAL BIOMARKER DATA

C. C. Brandt[1], J. C. Schryver[1], J. S. Almeida[2], S. M. Pfiffner[3], and A. V. Palumbo[1]

[1]Oak Ridge National Laboratory, Oak Ridge, TN
[2]Medical University of South Carolina, Charleston, SC
[3]The University of Tennessee, Knoxville, TN

OBJECTIVES

- Develop new nonlinear data analysis tools for relating microbial biomolecular markers to geochemical conditions.
- Apply these nonlinear tools to field and laboratory studies relevant to the NABIR Program.
- Provide these tools and guidance in their use to other researchers.

OVERVIEW

A major challenge in the successful implementation of bioremediation is understanding the structure of the indigenous microbial community and how this structure is affected by environmental conditions. Culture-independent approaches that use biomolecular markers have become the key to comparative microbial community analysis. However, the analysis of biomarkers from environmental samples typically generates a large number of measurements. The large number and complex nonlinear relationships among these measurements makes conventional linear statistical analysis of the data difficult. New data analysis tools are needed to help understand these data.

We adapted artificial neural network (ANN) tools for relating changes in microbial biomarkers to geochemistry. ANNs are nonlinear pattern recognition methods that can learn from experience to improve their performance. We have successfully applied these techniques to the analysis of membrane lipids and nucleic acid biomarker data from both laboratory and field studies. Although ANNs typically outperform linear data analysis techniques, the user must be aware of several considerations and issues to ensure that analysis results are not misleading:

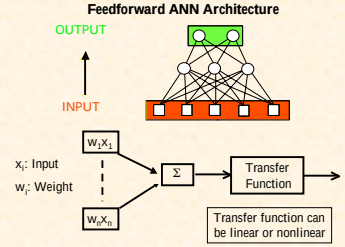
1) Overfitting, especially in small sample size data sets
2) Model selection
3) Interpretation of analysis results
4) Availability of tools (code)

This poster summarizes approaches for addressing each of these issues.

APPROACH

- ANNs are nonlinear and non-distributional. They make weaker assumptions than traditional statistical classifiers, are tolerant of missing or noisy data, and perform rapid analysis on new data.
- ANNs are networks of simple computational units interconnected by links or weights.
- **Supervised Learning**: Feedforward ANNs (FFANNs)
  - Used to predict a set of variables from a second set of variables.
  - Regression analysis; classification.
- **Unsupervised Learning**: Input Learning (modification of FFANNs)
  - Used to manage complexity in high-dimensional data sets in order to simplify analysis, visualize data, and increase understanding.
  - Dimension reduction; clustering; feature extraction.

Feedforward ANN Architecture



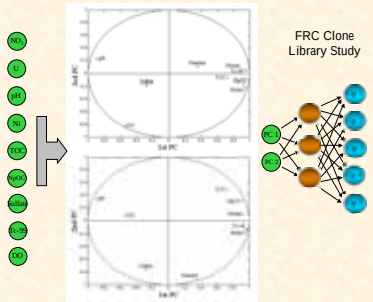
EXAMPLE DATA

- FRC Clone Library Study – relate geochemistry to *nirS*, *nirK*, and *dsrA* clones in groundwater samples taken from six wells (5 contaminated) at the FRC.
- Microcosm Study – relate addition and removal of metals to membrane lipids (PLFAs) in laboratory microcosms.
- Shiprock (UMTRA) Study – relate geochemistry to membrane lipids in groundwater samples collected at the Shiprock, NM UMTRA site.

(1) OVERFITTING

Input Reduction through Principal Components Analysis

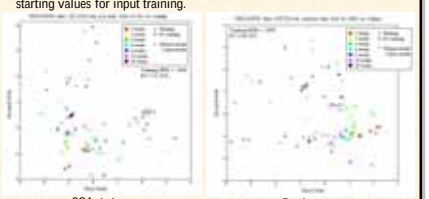
- A small sample size can increase the risk of overfitting in an ANN containing a large number of weights.
- If a small number of principal components explain a large proportion of the variance, then these components can be substituted for the original variables. Thus the model size is substantially reduced with little loss of information.



- pH, nitrate, Tc-99, nickel, total organic carbon (TOC) and nonpurgable organic (NpOC) carbon load heavily on PC1.
- Uranium and sulfate load most heavily on PC2.
- Dissolved oxygen (DO) loads most heavily on PC3.
- First three PCs accounted for 91% of variance in data set.

Dimension Reduction Through Input Training

- Reduce dimensionality of input data by finding optimum values for a small number of input nodes in an ANN. A FFANN predicts or reproduces the original (scaled) values in the output layer.
- Ordinary backpropagation algorithm was extended to the input layer where activation values are re-estimated during training along with weights to serve as a condensed representation of the original input.
- Nonlinear analogue of PCA (Tan and Mavrovouniotis 1996).
- Different reduced dimension representations can be generated by starting with random initial input values.
- A very efficient solution is obtained when the output of a PCA is used as the starting values for input training.



Cross-Validation Method (Leave-One-Out)

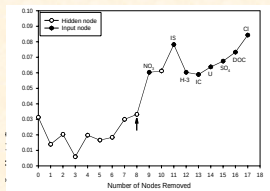
- For N sample data points, train the model on N-1 data points, and then test model generalization performance on data point left out of training.
- Repeat first step N times, leaving out a new data point on each iteration, until every data point has been tested on a different model.
- Estimate training error with a model training using all the data.
- Estimate the overall leave-one-out (LOO) error by combining the LOO estimates from the individual iterations.
- Data-efficient and very useful for small samples because part of the data does not have to be set aside for testing – all data is used for both training and testing.

Weight Decay

- A form of regularization that aids in minimizing overfitting.
- Improves generalization performance of ANNs (Mackay 1992).
- An additional penalty is added to the error term in computing the cost function during training. The extra term tends to reduce model complexity by imposing a penalty on large weights, where large weights are associated with more complex functions.

(2) MODEL SELECTION
Sensitivity-Based Pruning

- Many methods exist to define optimum ANN architectures. Two general categories are constructive and pruning methods. Pruning methods start with an architecture that is known to be too large and eliminate nodes/weights until a parsimonious model is found.
- Some pruning methods are: (1) optimal brain surgeon, (2) weight elimination and (3) sensitivity-based pruning (Moody 1992) which is exclusively used for node pruning. An advantage of (3) is that it is capable of simultaneously pruning nodes in both input and hidden layers.
- In sensitivity-based pruning a large initial network is trained. The training error is computed after deleting each individual node in the training and hidden layers. The node resulting in the smallest error increase is eliminated. The procedure is repeated until all possible deletions produce large errors.
- In this example, the optimum architecture is found after 8 deletions in the hidden layer, but none in the input layer.



Shiprock Groundwater Data

(3) INTERPRETATION OF ANALYSIS RESULTS

- ANNs are often treated as a black box which is unsatisfactory for understanding relationships between sets of variables. We often want to estimate the relative sensitivities or importances of a set of explanatory variables for a given set of predicted variables. However, sensitivity is often ill-defined outside a specific problem context.
- Many measures are only locally defined or are biased toward large or small values. Two popular definitions of local sensitivity (Saltelli et al. 2000):

$$(1)\, S_i = \frac{\partial Y}{\partial X_i} \quad (2)\, S_i = \frac{X_i^0}{Y^0}\frac{\partial Y}{\partial X_i}$$

These measures are converted to global values: mean(abs(Sj))

- We want to estimate the relative importances of a set of predictor variables to a set of predicted variables. We base the sensitivity (S_i) of input variable x_i on the increase in error resulting from the effective removal of x_i as a casual factor. This is done by substituting the mean value of x_i in each sample and then running the ANN using all the data. The sum of squared error (SSE_i) resulting from this method is:

$$SSE_i = \sum_j \left(y_j - \hat{y}_j \mid x_i \to x_i\right)$$

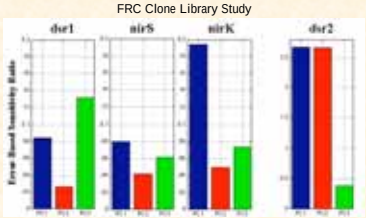
- The sensitivity is the proportional increase in model error after effective removal of the input variable:

$$S_i = \left(SSE_i - SSE_{orig}\right)/SSE_{orig}$$

where SSE_{orig} is the total error resulting from using the original values to train the ANN.

- This unbiased local model-error-based importance index is based on amount of degradation in the accuracy of ANN predictions when a particular input is effectively removed from the model.
- Inputs that are "important" predictors should severely degrade ANN performance when their effects are removed from the model.

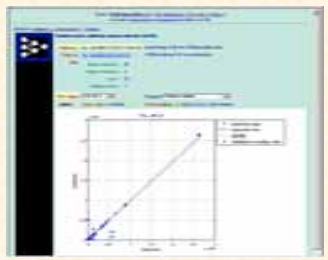
FRC Clone Library Study



- *nirK* and *nirS* are most sensitive to the 1st principal component (nitrate, pH, carbon, Tc-99, and nickel).
- *dsr*2 is most sensitive to the 1st and 2nd principal components while *dsr*1 is most sensitive to the 3rd principal component (DO).

(4) AVAILABILITY OF ANN DATA ANALYSIS TOOL KITS

- **Web-Based Toolkit** - http://www.bioinformatics.edu/webann
  - User-friendly interface that allows users to easily submit data
  - Includes modules for: (1) Data summarization; (2) Architecture selection; (3) Prediction, classification and dimension reduction
  - Uses readily available software components.
  - Allows new components to be easily added.



- **Matlab Toolkit**
  - Code (m-files) is available to perform FFANN training, input training, error-based pruning, k-fold cross-validation, simulated annealing, importance analysis, smooth predictions using kernel regression, and input reduction with PCA.
  - M-files run with freely available Netlab toolkit (Nabney 2000).
  - Contact authors for distribution

SUMMARY

- ANNs are robust and powerful data analysis tools for uncovering nonlinear relationships in complex microbial data sets.
- ANNs typically outperform linear models in predicting microbial community structure from geochemistry. An important exception is when the underlying relationship is simple or approximately linear. In this case, it is probably better to use an accurate parametric model.
- Small number of samples combined with a large number of measurements increases the danger of overfitting with ANNs with many inputs and outputs. Techniques that help mitigate the possibility of overfitting are:
  - Dimension reduction of inputs or outputs with PCA
  - Error-based pruning of inputs and hidden nodes
  - Weight decay
- Nonlinear PCA can be accomplished with a simple modification of an FFANN and is guaranteed to be at least as accurate as PCA.
- The behavior of ANNs can be examined through importance/sensitivity analysis and the visualization of smooth ANN predictions with the aid of kernel regression. The proper use of different importance/sensitivity metrics is an open area of future research, because the choice of metric can have large effects on the results.

ACKNOWLEDGEMENTS

- This research was funded by the Natural and Accelerated Bioremediation Research (NABIR) program, Biological and Environmental Research (BER), U.S. Department of Energy (ERKP280).
- Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. Department of Energy under contract DE-AC05-00OR22725.

REFERENCES

Mackay, D. J. C. 1992. Bayesian Interpolation Neural Computation. 4:415-447.

Moody, J. 1994. Prediction risk and architecture selection for neural networks. *In* J. H. F. V. Cherkassky, and H. Wechsler (ed.), From Statistics to Neural Networks: Theory and Pattern Recognition Applications. Springer-Verlag.

Nabney, I. T. 2001. NETLAB: Algorithms for pattern recognition. Springer, London, UK.

Saltelli, A., S. Tarantola, and F. Campolongo. 2000. Sensitivity analysis as an ingredient of modeling. Statistical Science 15:377-395.

Tan, S. and Mavrovouniotis, M.L. 1996. Reducing data dimensionality through optimizing neural network inputs. AIChE Journal 41:1471-1480.