Biophysical Journal Volume 67 November 1994 2120–2122

Letter to the Editor

"Simulated Molecular Evolution" or Computer-Generated Artifacts?

Frank Darius* and Raúl Rojas‡

*Institute for Ecology, Technical University of Berlin, 12165 Berlin, and ‡Mathematics and Computer Science Department, Free University of Berlin, 14195 Berlin, Germany

INTRODUCTION

In a paper by Schneider and Wrede (1994) published recently in the *Biophysical Journal*, the authors introduce a method which they call "Simulated Molecular Evolution" and which should allow designing of "optimal" motif amino acid sequences using a computer. Leaving aside several technical details, the core of the problem with this article is that a function is fitted to so few experimental points that the reported results lack statistical significance. By calling the fitted function the "quality" of the amino acid strings, they give it a kind of biological interpretation which is not supported by a theoretical explanation. And by optimizing a fitted function which possesses a large error margin, they get some numerical results which are pure numerical artifacts. A comment by Ellis (1994) has already relativized the findings of the authors but we feel it does not go into the heart of the matter.

METHODS

The authors are interested in analyzing cleavage sites in some kinds of protein precursors. To do this, they fix their attention on 12 amino acid positions, 10 before the cleavage site and 2 thereafter. They take 24 known protein sequences from the SwissProt database, 17 for training and 7 for testing. By coding each amino acid in terms of four physical properties they get a 48-dimensional vector for each one of the sequences. They also generate 68 additional 48-dimensional vectors from sequences of length 12, where no cleavage site is present in the selected proteins. They let a neural network learn to classify both sets. The output of the network should be 1 for the 17 positive examples and 0 for the 68 negative ones. Since a feed-forward neural network is nothing but a fit with a continuous function, what they are effectively doing is fitting a function to their data (anywhere you read "neural network" translate it into "nonlinear fit"). It is not difficult to see that 17 positive and 68 negative examples in a 48-dimensional space allow many possible function fits. Compared with three dimensions, it is like having 1 point in three-dimensional space where the value of the function should be 1, and 4 points where it should be 0. The authors admit in training an error margin of about 0.5 in the fit. Many functions with wildly differing shapes can fit this data with such large margin of error.

In Fig. 1 we try to give a visualization of the main idea of the fit used by the authors. A few positive examples in 48-dimensional space are assigned the value 1 (cleavage site present) and a few negative examples the value 0 (no cleavage site present). A continuous function (a neural network) is used to approximate the function values.

LACK OF ERROR ANALYSIS

The authors do not include in the paper an error analysis of their fit. The neural networks are trained in the following manner: if the output of the network is greater or equal to 0.5 for a given 48-dimensional vector (a coded string of length 12), the string is declared a positive example, that is, one where a cleavage site is present. If the output is less than 0.5 it is considered to be a negative example (Schneider and Wrede, 1993). This means that after training, the fitted function is not producing the defined values 1 and 0 exactly (cleavage site or no cleavage site present) but with a tolerated error of ~0.5, and so the networks produce all kind of values between 0 and 1 when confronted with new data. By calling the output of the network the "quality" of the amino acid string, the authors provide these intermediate values with a kind of biological interpretation. They start looking for those strings where the fit achieves a maximum and interpret this as an idealized "optimal" amino acid motif. But by their own definition the fitted function should be 1 at the positive examples and 0 at the negative ones. The fact that there are all kinds of values is a deficiency of the fit with no a priori biological interpretation.

The authors multiply the output of the three best networks found trying to improve their results. However, by multiplying the output of the three networks that provided the best fits the error becomes larger. A positive example could have obtained the values 0.51, 0.51, and 0.5 by each of the three networks, and since all three quantities are greater or equal to 0.5, this would be interpreted as a good result, that is, each network by itself recognizes the positive example as a positive one. But the product of the three quantities is about 0.125, so that now positive examples which should be assigned the value 1 by the combined network can effectively

Received for publication 5 April 1994 and in final form 5 August 1994.

Address reprint requests to Dr. Raúl Rojas, Mathematics and Computer Science Dept., Free University of Berlin, Takustrasse 9, 14195 Berlin, Germany. Tel.: 49 30 83875141; Fax: 49 30 83875109; E-mail: rojas@inf.fu-berlin.de.

0006-3495/94/11/2120/03 $2.00

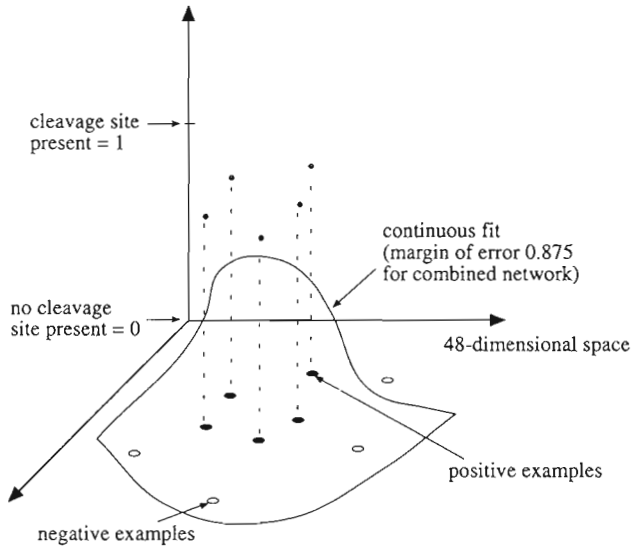

FIGURE 1 Fitting a continuous function to the data.

be assigned the value 0.125. The margin of error has thus grown from 0.5 during training to 0.875 in the actual application. That means that the output signal of the combined network is of about the same order of magnitude as the error. And yet the authors perform all kind of calculations up to the fourth and fifth decimal place and compose a list of strings (Table 3) where these are ordered according to the differences in the third or fourth decimal place. It is clear that a difference of 0.001 has no statistical significance in a case where the margin of error is 0.875.

A CONTINUOUS FIT?

But even if the authors had used more points in 48-dimensional space to make the fit by using some other protein sequences, why should nature care about it? Since they can find a fit to their data they then invert the whole process and assume that the fit found can be used to detect strings of 12 amino acids with even more "quality" than the initial data. Apart from the fact that there are many possible fitting functions, why should a continuous function fit this biological data? The coding of the protein sequences in 48-dimensional vectors is suppressing very important information, like the exact three-dimensional form of the protein. We are handling only partial information and, even worse, we are using only a few data points. It is unjustified to think that the fitting function is capable of detecting even better sequences than the natural ones. It all depends on the kind of fit you have.

By fixing the number of hidden units of the neural network the kind of functions that can be approximated is determined. The network with two hidden units used by the authors corresponds to two smooth step functions, which can be oriented any way you like. The other two networks with 4 and 11 hidden units are similarly simple and smooth. The fit is not exact (i.e., it does not produce the value 1 exactly for the positive examples) because the chosen functions have this simple shape. Since the number of experimental points is so reduced, increasing the accuracy of the fit does not help and makes it only more arbitrary. It is like fitting 12-dimensional polynomials to 10 data points.

OPEN CLUSTERS

The authors cite a well-known paper by Hornik to argue that neural networks are universal function approximators (i.e., they can provide a good fit to any continuous function you like). But this is only true if 1) you know which kind of function you are trying to approximate, and 2) you have enough variability, i.e., computing units, in the network. The authors used only networks with 2, 4, and 11 hidden units, which were combined by multiplying their outputs. If you are trying to isolate a cluster of points in 48-dimensional space you need in general 49 cutting hyperplanes to define a convex region of space. The 2, 4, and 11 hidden units (each one equivalent to a cutting hyperplane) of the authors are not enough to isolate a well-defined cluster in 48-dimensional space. The regions they analyze are open in many directions. And even if they had enough cutting hyperplanes the authors would still need to justify why the clustering occurs in a convex region of space. The result will be that many more points will be included in the cluster than the ones which represent real cleavage sites. This can be clearly seen in Fig. 6 of Schneider and Wrede (1994). The combined neural network fires at a multitude of sites in which there is no cleavage site. Although the authors write that "the majority" of the seven sequences tested yielded good results, Fig. 6 shows that in 4 of 7 cases the network could not distinguish the cleavage site from other false positions. In the other three the difference between the output of the network at the real cleavage site and false alarms is not very distinct. In the 7 tests there are many more sites in the sequences where the networks give a false result than real cleavage sites. Fig. 6 of the paper shows indeed that the function that was found cannot distinguish between cleavage sites and artifacts.

It is easy to see why the authors decided to multiply the output of their three neural networks: each network alone is too limited and identifies as optimal a whole lot of sequences which are not (since too few units have been used and the clustering is too coarse). By multiplying the three outputs, any time one or two of the networks produces a one for a sequence it can be killed if another network produces a 0. Contrary to custom the minority (one network) can overrule the majority (two networks). This helps to improve the clustering but can also kill the output of the network for the seven test points and makes the margin of error extremely large. This is what can be seen in Fig. 6.

RANDOM SEARCH

The authors look for a maximum of the objective function (in this case the function fitted to the 17 data points) by starting at some point in 48-dimensional space, generating 500 other points a distance apart from the first point but in 500 different random directions. They then take the best of these 500 points and iterate until a maximum of the objective function

is found. This is classical random hill-climbing. It does not help calling it an "evolution strategy" and the whole search for a maximum of the fitting function simulated molecular evolution. This terminology only obscures the whole technique being used, which has nothing to do with actual evolution.

IRRELEVANT METRIC

The authors also compare the quality of different amino acid distance matrices. There are no significant differences between the matrix of Euclidean distances and other kinds of metric. In each case the optimization process converges to an optimal point. Although the authors write that the context matrix leads to the "best sequence," the difference to the best one found with the Grantham matrix does not appear until the fifth decimal place, hardly a statistical difference and another example of the fact that the authors are not aware of the large error margin of their fit. As a kind of check, they tested what happens when a random matrix of amino acid distances is used. It is clear that in this case the iteration process will not converge because a random matrix is not the same as a randomly chosen metric. If the matrix is not at least symmetric, then no reasonable metric is being used and the iteration process will just keep jumping from point to point in search space. The random matrix which is introduced to perform some kind of control function is not doing so at all and the coding used by the authors is not being justified. Note that the rescaled context matrix of Euclidean distances in Table 1 is not a symmetric matrix (since the authors divided each row of the matrix by a constant in order to get a maximum distance of 1 from one amino acid to each other). This is equivalent to using different step lengths in each direction of the search space and effectively destroys the Gaussian distribution of step lengths that the authors want to use. Any matrix which defines a sensible partial ordering in search space should therefore lead to equivalent results. Moreover, it is possible to optimize the fitted function directly, without using random iterations and any kind of metric. Since the neural network implements a continuous and differentiable function, it is possible to calculate the gradient at each point, follow the gradient direction and find some optimum point (Hertz et al., 1991). It suffices then to calculate a sequence of 12 amino acids sufficiently near to this synthetic optimum. It is clear that any of the four distance matrices should then lead to very similar results.

INSUFFICIENT VISUALIZATION

The authors check their local optima through a "new method for visualizing the search space." The "new method" consists of a cut in one direction of the search space to see if in this direction the function reached a maximum. Apart from the fact that this could hardly be called a "new method," many more cuts are needed to get a feeling for the shape of the function in 49-dimensional space. The authors only look in the forward direction and not backward, which gives no guarantee that the algorithm has stopped at a local maximum. But since the neural networks used are so simple, it is not difficult to see finding the optimal point should be straightforward. The one-dimensional cuts are really unnecessary.

SUMMARY

1. The authors define a function with value 1 for the positive examples and 0 for the negative ones. They fit a continuous function but do not deal at all with the error margin of the fit, which is almost as large as the function values they compute.
2. The term "quality" for the value of the fitted function gives the impression that some biological significance is associated with values of the fitted function strictly between 0 and 1, but there is no justification for this kind of interpretation and finding the point where the fit achieves its maximum does not make sense.
3. By neglecting the error margin the authors try to optimize the fitted function using differences in the second, third, fourth, and even fifth decimal place which have no statistical significance.
4. Even if such a fit could profit from more data points, the authors should first prove that the region of interest has some kind of smoothness, that is, that a continuous fit makes any sense at all.
5. "Simulated molecular evolution" is a misnomer. We are dealing here with random search. Since the margin of error is so large, the fitted function does not provide statistically significant information about the points in search space where strings with cleavage sites could be found. This implies that the method is a highly unreliable stochastic search in the space of strings, even if the neural network is capable of learning some simple correlations.
6. Classical statistical methods are for these kind of problems with so few data points clearly superior to the neural networks used as a "black box" by the authors, which in the way they are structured provide a model with an error margin as large as the numbers being computed.
7. And finally, even if someone would provide us with a function which separates strings with cleavage sites from strings without them perfectly, so-called simulated molecular evolution would not be better than random selection. Since a perfect fit would only produce exactly ones or zeros, starting a search in a region of space where all strings in the neighborhood get the value zero would not provide any kind of directional information for new iterations. We would just skip from one point to the other in a typical random walk manner.

REFERENCES

Ellis, L. B. M. 1994. The inverse protein folding question and simulated molecular evolution. *Biophys. J.* 66:275.

Hertz, J., A. Krogh, and R. Palmer. 1991. Introduction to the Theory of Neural Computation. Addison-Wesley, New York.

Schneider, G., and P. Wrede. 1993. Development of artificial neural filters for pattern recognition in protein sequences. *J. Mol. Evol.* 36:586–595.

Schneider, G., and P. Wrede. 1994. The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site. *Biophys. J.* 66:335–344.