A Two-Step Neuro-Fuzzy Predictor for Genomic Time Series[1]

Horia-Nicolai Teodorescu [*,**], **Lucian-Iulian Fira** [**]

[*] Technical University of Iasi
[**] Institute for Theoretical Informatics of the Romanian Academy – Iasi Branch

Abstract: In previous papers, we used a two-component decomposition of the predicted signal consisting in genomic time series; namely low and fast varying components have been determined. In this paper, we improve the methodology of time series preprocessing for prediction. The third component consists in the part of the original series which is not well predicted using the previous decomposition. The prediction using this new method of decomposition show significantly improved results.

1. Introduction

Time series prediction is a topic of interest in many applications, ranging from economy to medical science. Due to the importance of the topic, a large number of tools have been applied to prediction, including linear models and nonlinear models as neural networks (see [1] for a specific neural network used in the Santa Fe contest for prediction), hidden Markov models [2], and fuzzy systems [3]. In this paper, we present a method for improving prediction and we apply it to a neuro-fuzzy predictor used in predicting genomic time series.

Genomic time series analysis is today a major topic in bioinformatics. This field can be defined as "the computational organisation and analysis of biological information" [4]. Bioinformatics and its tools are needed because of the huge amount of genomic data in the DNA sequences available today [5]. In the field of modern bioinformatics, the study of viruses has contributed to many of the methods, even though viruses are minute in genome size and complexity relative to their host genomes [5]. The need for methods to collate analyse genomic sequence data was imposed by complete viral genomes sequencing over the last 30 years [5].

For a better understanding of host-virus interactions in a biologic system, we need an integration of the knowledge dispersed at various levels: virus specific information in databases, the literature and the 'walking' expert systems [5].

In the post genomic era, the next stage is the functional genomics - the study of genes, their resulting proteins, and the role played by the proteins in the body's biochemical processes [6].

In the functional genomics, for a sequence with unknown role, the searching of the similar sequences from those with known functions is performed. As a tool for identifying the similar sequences, we selected a neuro-fuzzy predictor. Based on an idea of the first author, a predictor which learned a specific sequence may

[1] A version of part of this paper has been presented to ECIT2004 conference, 21-23 July, 2004, Iasi

recognize a similar sequence and reject a foreign sequence [7]. The decision of recognizing or rejecting could be made based on the small, respectively high prediction error values [8].

The time series prediction is applied in gene prediction. In [8-10], we tested a neuro-fuzzy system for time series predictions. The time series are obtained in [8-10] by calculating the distances between successive occurrences of the same basis (A, C, G or T) in a genomic sequence. This methodology for prediction was first introduced for application to the prediction on natural language texts, namely for the distance between words [11].

In this paper, we improve the time series preprocessing methodology for prediction and we discuss some consequences of the results obtained on the genome of a virus. While in the previous papers we have decomposed the time series only in two components, in this paper we add a third component. This component could have the significance of a new part of the series, which was not well separated in our previous approach.

The organization of the paper is as follow. The second section is devoted to the methodology. In the third section, results are presented. The last two sections contain a discussion and conclusions.

2. Methodology

The basic predictor structure used in this paper is depicted in Fig. 1. This is a one-step predictive system based on Sugeno fuzzy systems. For the slow and fast varying component, we used two such neuro-fuzzy predictors. The network architecture is a finite response predictor topology using Sugeno systems for the "weights" [12]. A number of twelve fuzzy systems act as "multipliers" of the delayed samples. The type-0 Sugeno fuzzy system with single input and single output was chosen for these cells. The input of fuzzy systems is characterized by seven Gauss type membership functions.

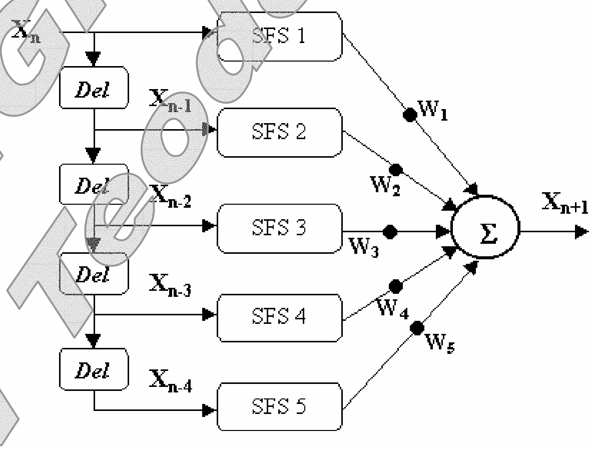

Fig. 1 – The topology of transversal type filter with Sugeno fuzzy system network [12]

We recall the input-output function of the neuro fuzzy predictor in Fig. 1, described by the equation (1), [10], [11].

$$Y=\sum_{k=0}^{M} w_k \cdot \frac{\sum_{l=1}^{N} \beta_{kl} \cdot e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}}{\sum_{l=1}^{N} e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}} \tag{1}$$

In (1), M denotes the number of Sugeno fuzzy systems, N the number of membership functions for each Sugeno fuzzy system, a_{kl} denotes the centers of the Gauss type membership functions, β_{kl} are the value of the singletons and w_k are the weights.

The first stage is preprocessing the time series A_n and consists in a low pass filtering for obtaining the slow varying (trend) component, A_n^s. The fast varying component, A_n^f, is determined by subtraction:

$$A_n^f = A_n - A_n^s \tag{2}$$

The low pass filtering consists in a moving average procedure.

The first prediction error ε_n is obtained from the original time series A_n using the predicted slow varying component $\tilde{A}_n^s$ and the predicted fast varying component $\tilde{A}_n^f$, namely ε_n is determined by subtraction:

$$\varepsilon_n = A_n - \tilde{A}_n^s - \tilde{A}_n^f \tag{3}$$

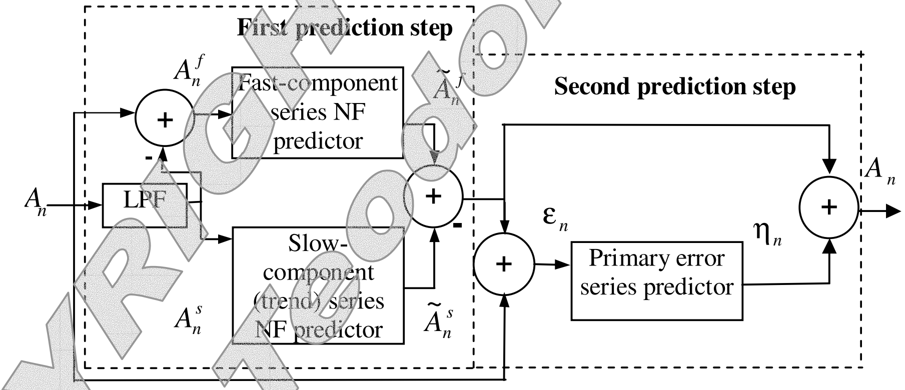

Fig. 2 - The general configuration of the predictor

Various predictor types can be chosen for the primary prediction error. In this study, we use a neuro-fuzzy predictor, similar with the predictor for the slow and fast varying component predictions.

The rational for using this type of predictor is based on the complexity of the input-output function of this predictor, on its large number of parameters, and on the possibility to use various algorithms to train it, as explained below.

The class of predictor is given by the input-output function of the predicting system. This function is a ratio with sums of exponentials at the nominator and the denominator. Therefore, the capabilities of this type of predictor are higher than for simple fuzzy logic systems with triangular or other piecewise input and output membership functions. Also, the characteristic function of this predictor is more intricate than the sum of sigmoidal functions, as in the case of a single layer perceptron. Compared to a MLP using sigmoidal neurons, which has the characteristic function essentially represented by composed sigmoidal functions, the characteristic function of this predictor is still more intricate, because it is a ratio of nonlinear functions. Compared to a MLP using Gaussian RBF neurons, which has the characteristic function represented essentially by composed Gaussian RBF functions, the characteristic function of this predictor is still more intricate, for similar reasons as above.

Regarding the number of parameters involved, showing the adaptation flexibility of the predictor, this predictor is similar to the ones discussed above. Moreover, by adding belief degrees to the rules of the TSK-fuzzy systems representing the neurons, the number of parameters is easily increased.

Regarding the possibility to use various learning algorithms, including gradient-type algorithms, we notice that the gradient algorithms are easily adaptable to this predictor, in contrast to classic fuzzy systems with piecewise input and output membership functions, which do not accept classic gradient algorithms (because of the non-derivability of the function).

3. Simulation Results

The primary prediction error ε_n for a time series which represents distances between successive occurrence of the A basis in HIV-1 genome, ENV gene [13], is shown in Fig. 3.

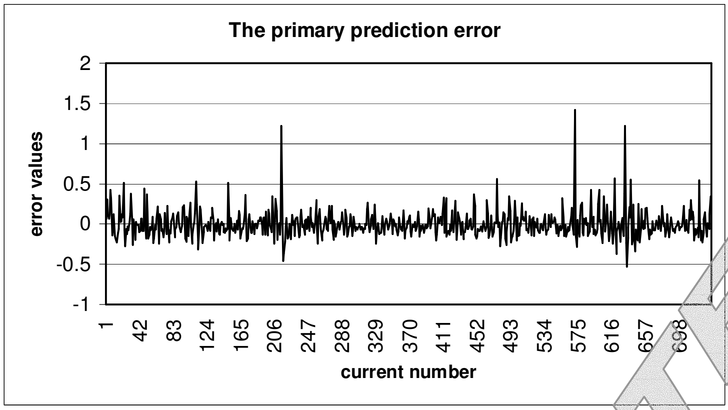

Fig. 3 - The primary prediction error ε_n

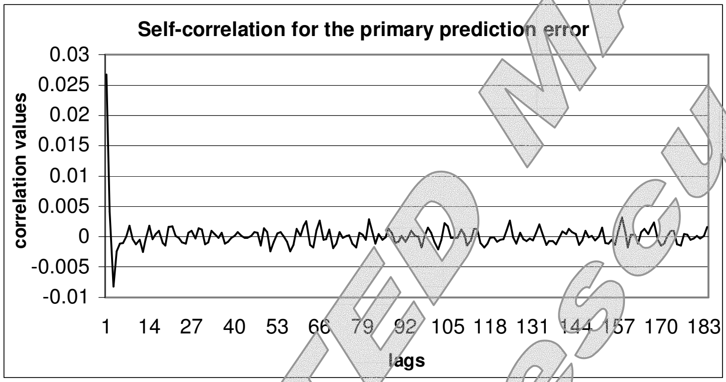

Fig. 4 - The self-correlation function of the primary prediction error ε_n

The self-correlation functions of the two errors (see Fig. 4 and Fig. 6) show very little correlation (< 0.05) meaning that the errors are mainly white noise – a fact confirmed by the histogram of the errors.

The secondary prediction error is shown in Fig. 5. Trying to train a predictor for the primary error signal, we obtained three influence zones of the predictor outcome, marked with the arrows in Fig. 5. These influence zones correspond with the peaks on the primary prediction error, ε_n. A way to reduce the influence of peaks is to reduce the amplitude of "aberrant values" to the value of the standard deviation.

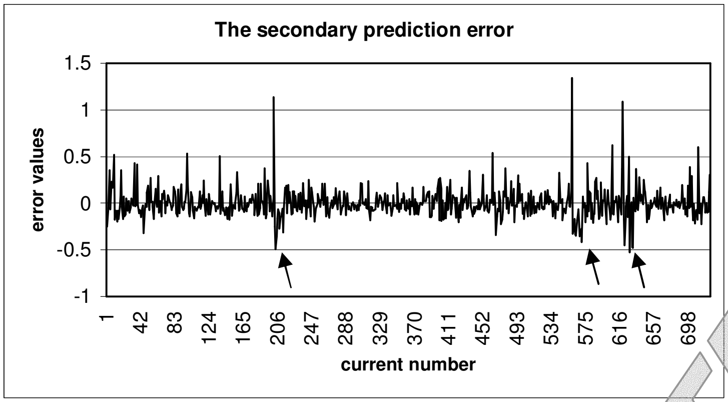

Fig. 5 - The secondary prediction error η_n . Arrows mark the influence the peaks have on the predictor outcome

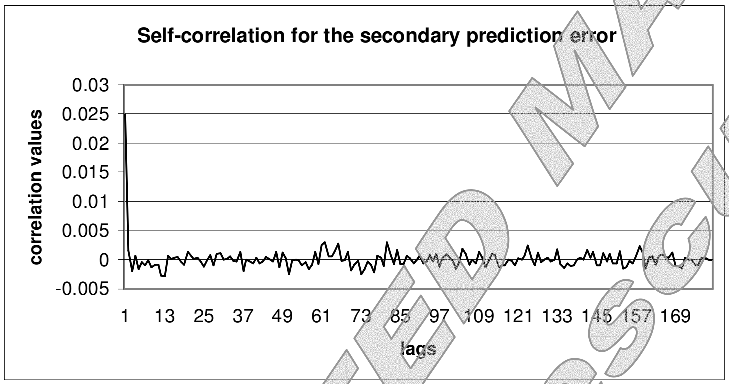

Fig. 6 - The self-correlation function of the secondary prediction error η_n

The residual error η_n shows regions where the predictor has been significantly fooled by peaks ("aberrant values") in the time series; large errors with both positive and negative values occur in the prediction just after the peaks.

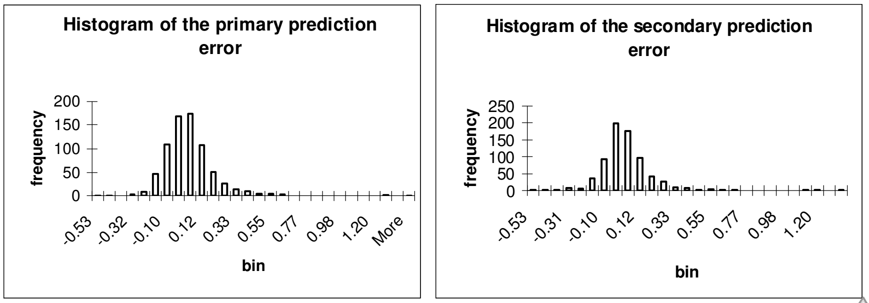

Fig. 7 - The histograms of the errors: a) initial error. b) histogram for the secondary prediction error η_n. Notice that there is no reduction in error range from -0.1 to 0.33.

The prediction does not change the general appearance of the error time series, as the error ε_n (solid thin line) and the error η_n (dotted line) look quite similar, as shown in Fig. 8. With solid line, was represented the difference between ε_n and η_n. For a better visibility, a zoom for the last 100 samples is illustrated in Fig. 9.

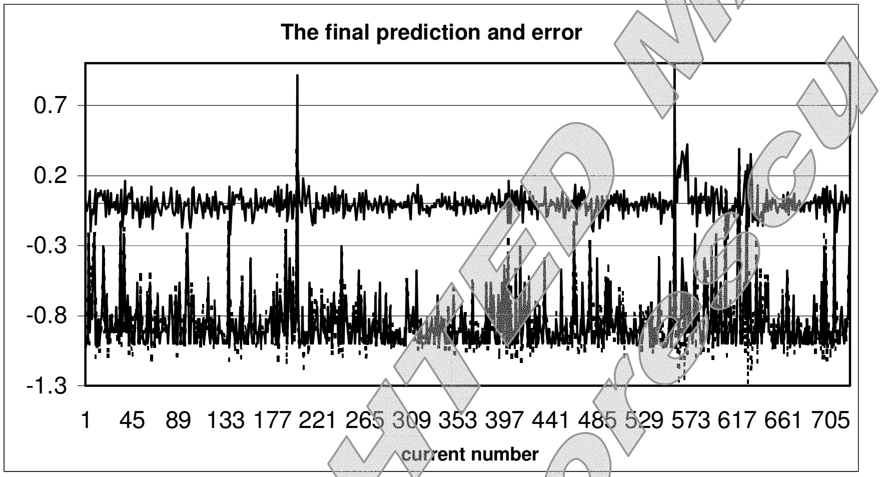

Fig. 8 - The true and the predicted A time series, as trained with the system in Fig. 2 on the HIV1 genomic segment of the ENV gene.

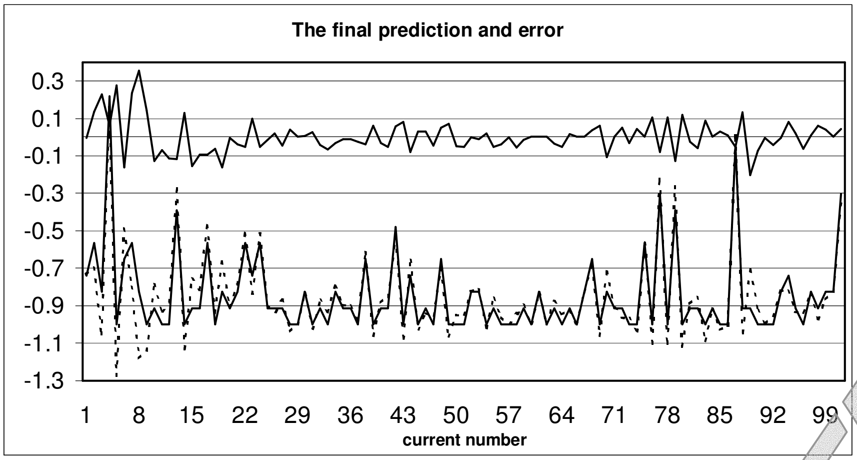

Fig. 9 - The true and the predicted A time series, as trained with the system in Fig. 2 on the HIV1 genomic segment of the ENV gene. Zoom for the samples [615, 715]. No filtering of aberrant errors.

The signals represented in Fig. 8 correspond to the TEST period. The accuracy of prediction is notable (see Fig. 9); the maximum difference, of about 0.37, is reached by a single sample (see Fig. 10).

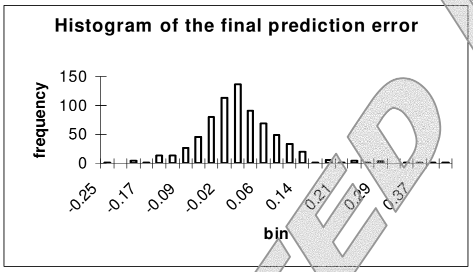

Fig. 10 - The histogram of the final prediction error

The histograms of the errors, illustrated in Fig. 7 and Fig. 10, are close to Gauss functions. That shows that most of the relevant information in the data has been used by the predictor. The Gauss-like histogram is not, of course, a guarantee that the signal (remaining error) is pure white noise, but it is a necessary condition for the error being white noise. Moreover, it is an indication that there is no strong global correlation in the remaining signal. Therefore, we need other tools to determine the degree of information extraction from the signal. Such a tool is the self-correlation function.

4. Discussion

The prediction of the error time series shows that: the prediction applied to ε_n does not change the general appearance of the error time series, as the error ε_n and the error η_n look quite similar; the correlation functions of the two errors show very little correlation meaning that the errors are mainly white noise – a fact confirmed by the histogram of the errors.

The remarks above lead to some consequences on the genome time series:

- There is a certain correlation in the distribution of the A, C, G, T bases, yet a significant part of the distribution along the genomic series of the bases looks random.
- Coherent information is not equally distributed on the four bases, because the errors look rather different.
- A significant part of the coherence is not quite specific to genes, but to the overall genome. Indeed, training on a gene the predictor does not guarantee in general. Removing the part of the coherence that is not gene-specific might improve the identification of the gene segments and splice positions – a hypothesis that must be tested yet.

5. Conclusions and further work

We conclude that the two-step NF predictor has extracted the available information in the time series and that further prediction should be based on models of white noise rather on correlation in the signal. In this respect, a HMM-based system might work better.

We have not yet studied the "word"-level distributions, i.e., the time series of the distances between "di-bases", like AC, AG, AT, … sequences. This task remains to be fulfilled. Adding di-bases prediction may improve the selectivity of the predictor in recognizing a specific sequence.

To improve the prediction after the occurrence of the aberrant values, while preserving the predictability of the aberrant values, during the training stage the aberrant values are taken into account in the section preceding them, while they are replaced by the value σ in the sections succeeding them.

Acknowledgments. The Romanian Academy Grant 155/2003-153/2004 has supported part of the research for this paper. This research is party performed for the Romanian Academy prioritary grant "Cognitive systems and applications" ("Sisteme cognitive şi aplicaţii"); however, there was no financial support from this Grant for this research. Also, part of the research has been done as preparatory research for the CNCSIS Grant "System for the analysis and prediction of genomic sequences".

References

[1] E. A. WAN: *Application of FIR Neural Networks to Time Series Prediction*. http://www.cse.ogi.edu/~ericwan/FIR/timeseries.html, accessed June 22, 2004.

[2] G. DANGELMAYR, S. GADALETA, D. HUNDLEY, M. KIRBY: *Time series prediction by estimating Markov probabilities through topology preserving maps*, Proc. SPIE Vol. 3812, Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation, II, pages 86-93, Eds. B. Bosacchi and D. B. Fogel and J.C. Bezdek (1999).

[3] X. LIU, B. KWAN, S. FOO: *Time Series Prediction Based on Fuzzy Principles*, http://devils.eng.fsu.edu/download/Time_Series_Prediction_Based_on_Fuzzy_Principles.pdf, accessed June 22, 2004.

[4] A. RODRIGO: *Editorial foreword*, Applied Bioinformatics, 2002:1(**1**) 1-2.

[5] P. KELLAM, M. ALBA: Virus bioinformatics: databases and recent applications", *Applied Bioinformatics*, 2002:1(**1**) 37-42.

[6] NORTHWESTERN UNIVERSITY, THE DEPARTMENT OF BIOCHEMISTRY, MOLECULAR BIOLOGY, & CELL BIOLOGY: http://bip.weizmann.ac.il/mb/functional_genomics.html, accessed June 21, 2004

[7] H.N. TEODORESCU: *Genetics, Gene Prediction, and Neuro-Fuzzy Systems – The Context and A Program Proposal*, Fuzzy Systems & A.I. – Reports and Letters, vol. **7** (2003), pp. 15–22.

[8] H.N. TEODORESCU, L.I. FIRA: *A Hybrid Data-Mining Approach in Genomics and Text Structures*, The Third IEEE International Conference on Data Mining ICDM '03, Melbourne, Florida, USA, November 19 – 22, 2003, pp. 649-652.

[9] L.I. FIRA, H.N. TEODORESCU: *Genome Bases Sequences Characterization by a Neuro-Fuzzy Predictor*, Proceedings IEEE-EMBS 2003 Conference, 17-21 September, Cancun, Mexico, pp. 3555-3558.

[10] H.N. TEODORESCU, L.I. FIRA: *Predicting the Genome Bases Sequences by means of distance sequences and a Neuro-Fuzzy Predictor*, Fuzzy Systems & A.I. – Reports and Letters, vol. **7** (2003), pp. 23-33.

[11] H.N. TEODORESCU: *The Dynamics of the Words*, Invited Plenary Lecture, The 11th Conference on Applied and Industrial Mathematics (CAIM 2003), 29-31 May, 2003. University of Oradea, Romania.

[12] T. YAMAKAWA, H.N. TEODORESCU: *Neuro-Fuzzy Systems: Hybrid configurations*. In vol. M.J. Patyra, D. Mlynek (Eds.): Fuzzy Logic. Implementation and applications. Wiley & Teubner, Chichester, 1996, pp. 267-298.

[13] LOS ALAMOS NATIONAL LABORATORY: http://hiv-web.lanl.gov/content/hiv-db/align_current/ align-index.html, accessed 6/20/04.