Analysis of the Predictability of Time Series Obtained from Genomic Sequences by Using Several Predictors

Horia-Nicolai Teodorescu[*,**], Lucian-Iulian Fira[**]
(*)Technical University of Iasi, (**) Institute for Theoretical Computer Science of the Romanian Academy

Abstract **– *In previous papers, we used one-step-ahead predictors for the genomic sequence recognition scores computation. The genomic sequences are coded as distances between successive bases. The recognition scores were then used as inputs for a hierarchical decision system. The relevance of these scores might be affected by the prediction quality. It is necessary to appreciate the prediction performance in a framework based on the analyzed time series predictability. The aim of this paper is to determine which predictors are most suitable for genomic sequence identification. We analyze linear predictors (like linear combiner), neuronal predictors (RBF or MLP type), and neuro-fuzzy predictors (Yamakawa model based). Several methods to appreciate the predictability of time series are used, like Hurst exponent, self-correlation function, and eta metric. All predictors were tested and compared for prediction quality using sequences from HIV-1 genome. The mean square prediction error (MSPE), direction test, and Theil coefficient were used as prediction performance measures. The prediction results obtained with the predictors are contrasted and discussed.***

Keywords **– *Distance series, genomic sequences, predictability, prediction performances, recognition scores.***

I. INTRODUCTION

Genomic sequences are represented as sequences of letters (A, C, G or T / U), that are the initials of nitric bases from ADN / ARN structures (Adenine, Cytosine, Guanine, Thymine / Uracil). This literal representation is improper for the computing systems used for the genomic sequence analysis. Numerous authors (see [1]) proposed different methods for representing the nitric bases or nucleotide sequences. The original representation used in this paper, proposed in [2], consists in the coding of genomic sequences by for time series, one four each basis type. Each series contains the distances between successive occurrences of the corresponding basis. Details about above mentioned coding can be found in [2], together with the methodology proposed by the first author, methodology that stand at the basis of this paper.

According to the methodology [2], if the slowly varying component (including the trend) and the fast varying component (including random component) are separated, a better prediction for the genomic time series is obtained. In this paper, these two components will be named trend and random component. The separation might be done using a moving average filter. Elements of the methodology were presented in [3-7], together with several results.

According to the methodology from [2], a hierarchical hybrid system for recognition of genomic sequence was designed and implemented [3-7]. We used one-step-ahead predictors for the genomic sequence recognition scores computation. These recognition scores are then used as inputs for a hierarchically superior decision system. The relevance of these scores might be affected by the prediction quality. It is necessary to appreciate the prediction performance framework according to the analyzed time series predictability.

The aim of this paper is to determine the predictors that are most suitable to use as genomic sequence identifiers. The choice is made amongst linear predictors (like linear combiner), neuronal predictors (RBF or MLP type), and neuro-fuzzy predictors (Yamakawa model based).

Hurst exponent, self-correlation function, and eta (η) metrics were used to appreciate the predictability of time series. All predictors were tested and compared for prediction quality using sequences from the HIV-1 genome. As prediction performances measures were used the following: mean square prediction error (MSPE), direction test, and Theil coefficient.

The paper is structured as follows: the next section is devoted to the description of the methodology. The third section briefly presents the predicting systems used. In the fourth section, we show several simulation results. In the last section, conclusions are outlined.

II. METHODOLOGY

A. Identifiers for genomic sequences

The aim of the design of the identifiers is to obtain tools able to scan genomic sequences and to identify the known (learned) patterns. The basic methodology to identify genomic sequence, using one-step-ahead predictors, was published in [2]. An already learned sequence will give a small prediction error at a subsequently testing. A foreign sequence might be rejected due of high prediction error. To verify the methodology, we tested linear predictors (linear

Copyright MATERIAL Lucian Fira

combiner), neuronal predictors (RBF or MLP type), and neuro-fuzzy predictors (Yamakawa model based).

The class of a predictor is given by the input-output function or by the characteristic function of the predicting system. We tried several predictors, including linear predictors based on linear combiners, MLP predictors, RBF predictors, and neuro-fuzzy (NF) predictors. In case of adaptive linear combiner (ALC) predictors (see Fig. 1), the characteristic function is a linear weighted sum of the delayed inputs:

$$f_1(x_n, x_{n-1}, x_{n-2}, \ldots, x_{n-k+2}, x_{n-k+1}) = w_0 + \sum_{j=1}^{k} w_j x_{n-j+1} \quad (1)$$

where k represents the predictor order, w_0 is the bias, and w_j are the weights of the linear combiner.

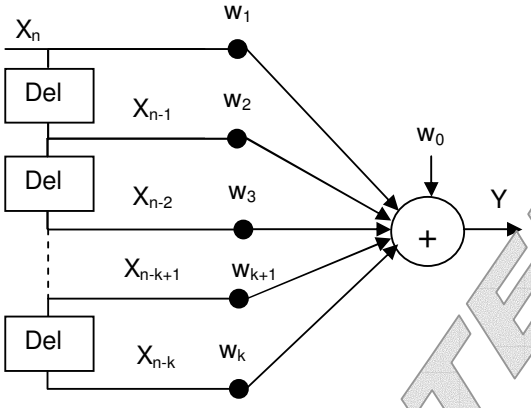


Fig.1 – One-step-ahead adaptive linear combiner based predictor

The *Del* symbol represents the delay operator, which ensures one-iteration-delay between the samples of consecutive inputs.

In case of single hidden layer perceptron, the characteristic function is:

$$f_2(x_n, x_{n-1}, x_{n-2}, \ldots, x_{n-k+2}, x_{n-k+1}) = w_0 + \sum_{i=1}^{H} w_i \frac{1}{1+\exp\left(-\left(w_{i0}^1 + \sum_{j=1}^{k} w_{ij}^1 x_{n-j+1}\right)\right)}, \quad (2)$$

where k represents the predictor order, w_0 is the bias and w_i are the weights of the output neuron, w_{i0}^1 and w_{ij}^1 are the bias and the weights, respectively, of the neuron $\#i$ from the hidden layer. The PE blocks are processing elements with sigmoid activation function.

An RBF network with Gaussian neurons in the hidden layer (see Fig. 3) has the characteristic function as a linear combination of Gauss functions:

$$f_2(x_n, x_{n-1}, x_{n-2}, \ldots, x_{n-k+2}, x_{n-k+1}) = w_0 + \sum_{i=1}^{H} w_i \cdot \exp\left(-\sum_{j=1}^{k} (x_{n-j+1} - c_{ij})^2 / \sigma^2\right) \quad (3)$$

where k represents the predictor order, w_0 is the bias, and w_j are the weights of the output neuron. H is the hidden Gauss type neurons number. Here, σ denote the spreading of the Gauss type functions (all functions are assumed to have the same spreading) and c_{ij} are the centers.

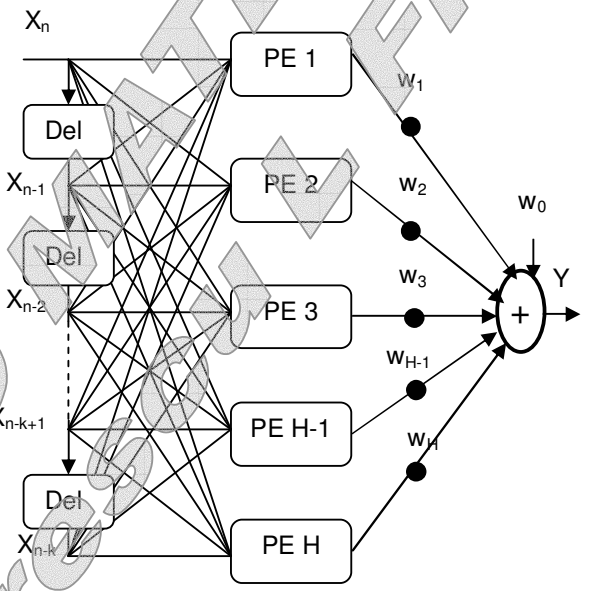


Fig. 2 – Architecture of a MLP based predictor

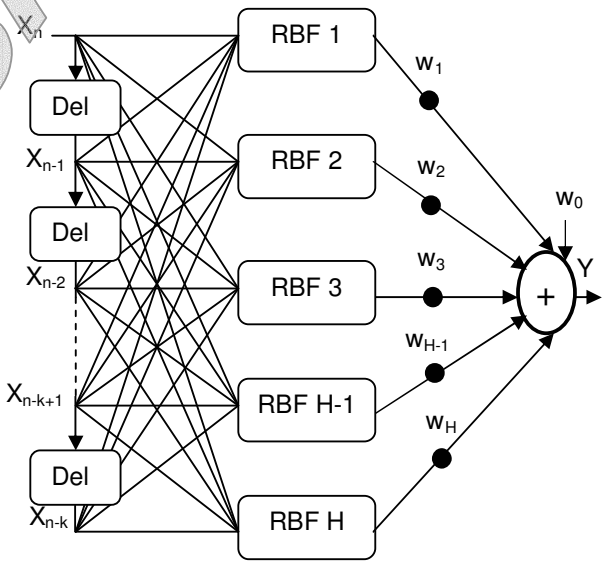


Fig. 3 – Architecture of a RBF based predictor

COPYRIGHTED MATERIAL Teodor Iliu-Fira

The RBF X blocks are Gauss type RBF functions.

In case of the neuro-fuzzy (NF) predictor (see Fig. 4), the architecture is a multi-fuzzy system network with inputs represented by the delayed samples. The fuzzy cells acting as multipliers of inputs are Sugeno type 0 systems, with Gauss input membership functions. The input-output function is a ratio with sums of exponentials at the nominator and the denominator.

$$f_3(x_n, x_{n-1}, x_{n-2}, \ldots, x_{n-k+2}, x_{n-k+1}) = \sum_{j=1}^{k} w_j \frac{\sum_{i=1}^{N} \beta_{ij} \cdot \exp\left(-\left(x_{n-j+1} - c_{ij}\right)^2 / \sigma^2\right)}{\sum_{i=1}^{N} \exp\left(-\left(x_{n-j+1} - c_{ij}\right)^2 / \sigma^2\right)} \quad (4)$$

where k is the predictor order, N is the input membership function number for each Sugeno fuzzy system, c_{ij} are the centers of the Gauss type input membership function, and β_{ij} are the output singletons $\#i$ of the fuzzy system $\#j$. The spreads of the Gauss type functions are assumed again equal and are denoted by σ; the weights associated to the output of the system $\#j$ are denoted by w_j.

In Fig. 4, the SFS X blocks are single-input-single-output 0-type Sugeno fuzzy systems. The input and the output membership functions are Gauss type and singletons type, respectively.

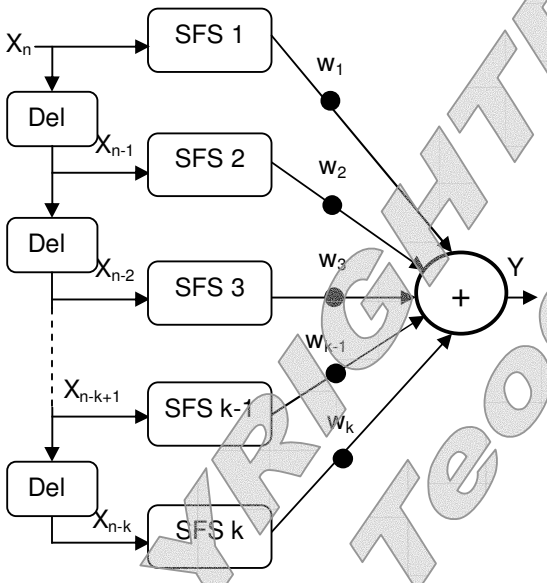


Fig. 4 – Architecture of a neuro-fuzzy predictor [20]

B. Predictability analysis tools

Predictability definition

In [8], predictability at a time in the future is defined by

$$\frac{R\left(x(t), x(t+\tau)\right)}{H(x(t))} \quad (5)$$

and linear predictability by

$$\frac{L\left(x(t), x(t+\tau)\right)}{H(x(t))} \quad (6)$$

where R and L are the redundancy and linear redundancy, and H is the entropy. Redundancy is

$$R(X_1, X_2, \ldots, X_n) = \sum_{k=1}^{n} H(X_i) - H(X_1, X_2, \ldots, X_n) \quad (7)$$

where $H(X_i)$ is the entropy and $H(X_1, X_2, \ldots, X_n)$ is the joint entropy. Linear redundancy is

$$L(X_1, X_2, \ldots, X_n) = -\frac{1}{2} \sum \log_2 \sigma_i, \quad (8)$$

where σ_i are the eigenvalues of the correlation matrix [9]. In [10], the (Shannon) entropy of a variable X is defined as

$$H(X) = -\sum_x P(x) \log_2 \left[\left(P(x)\right)\right] \text{ bits}, \quad (9)$$

where $P(x)$ is the probability that X is in the state x, and $P \log P$ is defined as 0 if $P = 0$. The joint entropy of variables $x_1, \ldots, x_n$ is then defined by

$$H(X_1, \ldots, X_n) = -\sum_{x_1} \ldots \sum_{x_n} P(x_1, \ldots, x_n) \log_2 \left[P(x_1, \ldots, x_n)\right] \quad (10)$$

Correlation analysis

For the time series obtained as we described above, to fast identify the dynamics associated with the genomic sequences, we made the correlation analysis. It is well known that the self-correlation function is a tool able to show that a time series is constant, periodical or random.

For a time series $y = \{y_t\}_{t=1..N}$, the correlation function is obtained by multiplying each y_t by $y_{t+\tau}$ and summing the result over all the data points. The average is then plotted as a function of lags τ. This gives a measure of how dependent data points are on their neighbors [11].

The correlation time (also named coherence time or correlation interval) is the number of lags for that the correlation values are still high, the consecutive samples being correlated [12].

A small correlation time is obtained for highly random data. White noise has no correlation, i.e., the correlation

COPYRIGHTED MATERIAL © Teodorescu Fira

function drops abruptly to zero. For highly correlated data, the correlation function slowly decreases.

Hurst exponent

It is known that the Hurst exponent (H) measures the fractal dimension of a data series [13]. A Hurst exponent of 0.5 indicates no long-term memory effect; this is the case of the random data. If $H > 0.5$, we have an indication of an increasing presence of long-term memory effect. In this case, data series reverse signs less frequently than would be true for white noise [14].

If $H < 0.5$, the data series is called anti-persistent. Namely, each data value is more likely to have a negative correlation with preceding values. Such data series reverse signs more frequently than would a white noise series [14].

Eta metrics

Introduced by Kabudan [15], the η-metrics measures the predictability level for the time series, using models based on genetic programming. The η-metrics consists in outputs comparison for two systems: the first system is the best one-step-ahead predictor for a specified time series, and the second is the best predictor for the time series obtained by shuffling the original series.

For a given time series $y = \{y_t\}_{t=1..N}$, the squared step prediction error, denoted by SSE_y, is:

$$SSE_y = \sum_{t=1}^{N} \left(y_t - \hat{y}_t\right)^2, \qquad (11)$$

where $\hat{y}_t$ is the predicted value of y_t.

For the shuffled time series, the prediction error is:

$$SSE_S = \sum_{t=1}^{N} \left(S_t - \hat{S}_t\right)^2, \qquad (12)$$

where S is y shuffled series.

According to Kabudan, the evaluation is given by [15]:

$$\eta_1 = 1 - \frac{SSE_y}{SSE_S} \qquad (13)$$

If the y series is totally determinist, it can be perfectly modeled. Then, $SSE_y = 0$, and $\eta_1 = 1$. If the y series is totally unpredictable, the shuffling has no influence for prediction, $SSE_y = SSE_S$, and $\eta_1 = 0$.

According to Duan [16], the η-metrics has two disadvantages. The first is that the value given by the metrics depends on the time series length. The second disadvantage consists in the small resolution of the metrics, i.e. a great number of series (financial) may have η values belonging to a small interval, namely (0.9, 1) versus (0,1).

If $MSE_x = SSE_x / N$, then the ratio MSE_y / SSE_S is identical to the ratio SSE_y / SSE_S. Therefore, one can use the mean square error for the η_1 metrics evaluation.

An improvement of the η_1 metrics, proposed by Duan [16], is

$$\eta_2 = 1 - \sqrt{\frac{SSE_y}{SSE_S}}. \qquad (14)$$

We propose two improved variants for the η-metrics:

$$\eta_{31} = \left(1 - 10^k \cdot MSE_y\right)\left(1 - \frac{SSE_y}{SSE_S}\right) \qquad (15)$$

$$\eta_{32} = \left(1 - 10^k \cdot MSE_y\right)\left(1 - \sqrt{\frac{SSE_y}{SSE_S}}\right) \qquad (16)$$

The proposed metrics improves the advantages of previous metrics by tacking into account of MSE obtained at the test. A series with small testing MSE is more predictable than in case of high testing error. Here, k is a scaling exponent.

C. Prediction error performances evaluation

For the one-step-ahead prediction, a common error is the repetition phenomenon known as naive prediction, or trivial prediction. The repetition consists in output of a one-sample delayed series versus the desired series. Namely, the input that is represented by the current sample is transmitted unchanged to the output. The repetition is a local optimum for the training of neuronal systems.

Prediction error

The commonly used measure for the prediction performances is the mean square prediction error (MSPE). Other prediction error measures are the normalized mean square prediction error (NMSPE) and the root mean square prediction error (RMSPE).

Theil coefficient

The Theil coefficient compares the RMSE error for the obtained prediction and for the naive prediction. If the current value of a time series is y_t, then the naive prediction would be $y'_{t+1} = y_t$. If we have a desired series $\{y_{t,\, t=1,N}\}$ and a predicted series $\{y'_{t,\, t=1,N}\}$, then the Theil coefficient is defined as ([17] citing [18]):

$$T = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y_t - y'_t\right)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y_t - y_{t-1}\right)^2}}. \qquad (17)$$

COPYRIGHTED MATERIAL
HN Teodorescu L Fira

For the Theil coefficient the following interpretations are possible:

- if $T = 0$, then the prediction error is zero and the time series was perfectly predicted.
- if $T = 1$, then the prediction error is equal to the naive prediction and the predictor is the same with the naive predictor.
- if $T < 1$, then the obtained prediction is better than the naive prediction, the prediction quality increasing when T goes to zero.
- if $T > 1$, then the obtained prediction is weaker than the naive prediction, the prediction quality decreasing when T goes to infinity.

Direction test

To compare the prediction of direction changes the so-called "direction" test is used. A variant is to compute the number of correctly predicted changes, namely the number of positive products $y_t \cdot y'_t$, (with the same notations as in the previous paragraph).

Given a time series $\{y_{t,t=1,N}\}$ and the predicted time series $\{y'_{t,t=1,N}\}$, compute the directional prediction hit rate H [19] as

$$H = \frac{\left|\{t \mid y_t \cdot y'_t > 0, t = 1, N\}\right|}{\left|\{t \mid y_t \cdot y'_t \neq 0, t = 1, N\}\right|} \quad (18)$$

The same computation is made for the naive prediction, obtaining H_N,

$$H_N = \frac{\left|\{t \mid y_t \cdot y_{t-1} > 0, t = 1, N\}\right|}{\left|\{t \mid y_t \cdot y_{t-1} \neq 0, t = 1, N\}\right|} \quad (19)$$

The normalized hit rate is $H_0 = \frac{H}{H_N}$ [19]. A value $H_0 < 1$ indicates a real predictive power.

III. SYSTEMS USED FOR THE PREDICTION

A. Identifiers based on a neuro-fuzzy predictor

The neuro-fuzzy predictor we used is based on the Yamakawa neuronal model (see [20]). The model has a transversal filter with external delays, with cells represented by Sugeno fuzzy systems instead of weights. Each fuzzy cell consists in a type-0 Sugeno fuzzy system, with single input and single output. The fuzzy systems have seven Gauss-type input membership functions. The fuzzy systems act as multipliers of the delayed samples of the inputs, $\{x_n, x_{n-1}, \ldots, x_{n-M-1}\}$. By adaptation, the parameters of the systems are modified to obtain at the output a better approximation of the sample x_{n+1}. We used variants with different number of inputs (3-7,10,11). Details about the methodology were presented in [7]. The predictor training involves the adaptation of two sets of parameters: the output singletons β_i of the Sugeno fuzzy systems, and the weights w_i.

B. Identifiers based on neuronal predictors

For the neuronal identifies, two models were tested: RBF and MLP. The variant that involves RBF includes, in the neurons with radial basis functions layer, a number between 1 and 30 Gauss type neurons. The number of inputs is between 1 and 30, respectively, and the number of outputs is 1. The MLP network has one hidden layer, with 5 or 10 neurons with hyperbolic tangent activation function. Similarly, the number of inputs is between 1 and 30, respectively, and there is a single output.

C. Identifiers based on linear predictors

The linear combiner has the number of inputs between 1 and 30, and a single output.

IV. RESULTS

A. Predictability analysis

For the distance series between the A bases from the ENV gene, HIV-1 virus [21], we performed two analyses:

- the correlation analysis for the original time series;
- the correlation analysis for the trend and random component.

In the upper panel of Fig. 5, the original time series is shown, after normalization to the [-1,1] interval. In the lower panel, the graphic of the self-correlation function is shown. A strong continuous component is observed.

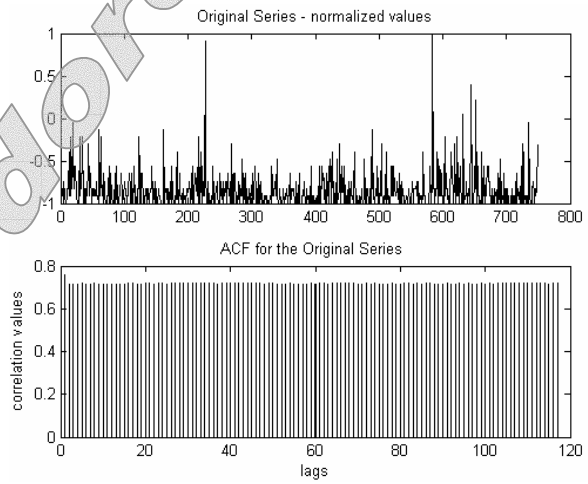


Fig. 5 – The original time series after normalization (top). The self-correlation function (down)

A causal MA filter having the equation

COPYRIGHTED MATERIAL Fără drepturi

$$y[n]=\frac{1}{3}\left(x[n]+x[n-1]+x[n-2]\right) \quad (20)$$

splits the two components, trend and random, as it is shown in Fig. 6 and Fig. 7.

The continuous component from the original time series is found in the trend component, as it is shown on the self-correlation function graphics in Fig. 1 and 2.

For the random component, the self-correlation function graphics (see Fig. 7, down) shows very small values, under 0.03, confirming the opinion of non-existence of a periodical component [2].

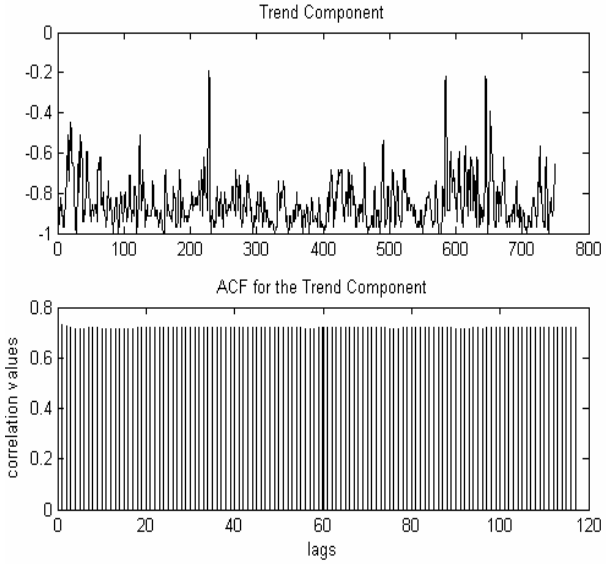


Fig. 6 - The trend component (top).
The self-correlation function (down)

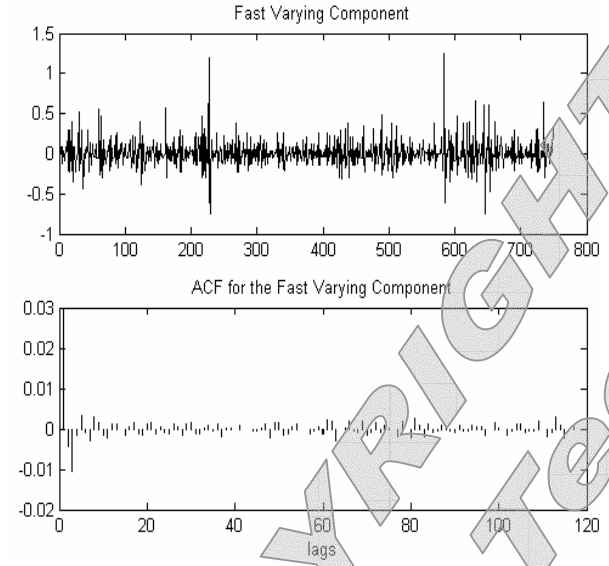


Fig. 7 - The random component (top).
The self-correlation function (down)

In Table 1, the values of the Hurst exponent are presented for an A-ENV original series and for its components.

Table 1. The Hurst exponent for the distance series between A bases

Hurst Exponent	Original series	Trend component	Random component
ENV A	0.0015	0.150	-0.0139

The computed Hurst exponent is less than 0.5 for the original series and its components. Thus, the series have negative self-correlation.

The measures of the predictability using eta metrics are summarized in Tables 2 and 3.

Table 2. Genomic series predictability measuring using eta metrics for the random component

Random component, ENV gene, A basis						
System	MSEy	MSEs	η_1	η_2	η_{31} k=10	η_{32} k=10
ALC	0.0271	0.0419	0.3523	0.1952	0.2567	0.1422
RBF	0.0233	0.0416	0.4407	0.2521	0.3382	0.1935
MLP	0.0238	0.0371	0.3583	0.1989	0.2729	0.1515
NF	0.0279	0.0423	0.3420	0.1888	0.2467	0.1362

Table 3. Genomic series predictability measuring using eta metrics for the trend component

Trend component, ENV gene, A basis						
System	MSEy	MSEs	η_1	η_2	η_{31} k=10	η_{32} k=10
ALC	0.0075	0.0192	0.6097	0.3752	0.5641	0.3472
RBF	0.0086	0.0188	0.5438	0.3246	0.4973	0.2968
MLP	0.0077	0.0190	0.5973	0.3654	0.5515	0.3374
NF	0.0115	0.0193	0.4039	0.2279	0.3574	0.2017

As we expected, the predictability of the trend component is better than the predictability of the random component. Indeed, all metrics (labeled η_1, η_2, η_{31}, and η_{32} in Tables 2 and 3) computed for the all predictors (ALC, RBF, MLP, and NF based) indicate greater values for the trend component than for the random component. For the scaling exponent k, a value of 10 was chosen because this values performed better than 1 and 100. We have an indication that the η_{31} and η_{32} metrics perform better than η_1 and η_2 metrics, because the proposed metrics increase the resolution of η_{31} and η_{32} metrics for the distances between bases series. This indication must be through tested for another time series type, like financial data or biological data, also. The metrics improved is obtained at the down of the measuring scale: all η_{31} and η_{32} metrics have values less than values for the η_1 and η_2 metrics, respectively.

B. Prediction performances

A comparison between the performances of different types of predictors is presented below.

COPYRIGHTED MATERIAL
HN Technologies Ltd

As Tables 2 and 3 show, the best predictor for the random component is a MLP with 0.0371 MSE for the test period. For the trend component, the best performance is obtained for a RBF, with MSE about 0.0188, followed by a MLP with MSE about 0.0190.

The performances of different predictors with different order are presented in the figures 8 to 11. The Test MSE series are plotted on the second (right) axis.

The performances of the predictors must be interpreted by cumulating of the indication given by the MSE on the TEST period, direction test, and Theil coefficient.

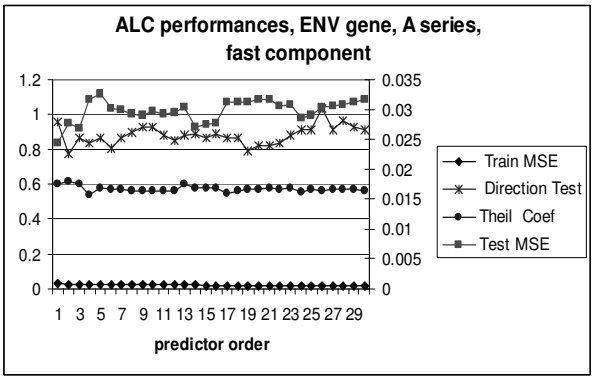


Fig. 8a – The prediction performances for different Adaptive Linear Combiner predictor orders; ENV gene, A series, fast varying component

In Fig. 8a, for ALC predictors trained on the fast varying component of A series from ENV gene, on the MSE series for the test period, a local minimum can be seen at the order 14. The direction test and the Theil coefficient indicate average values for the predictor with order 14. Another minimum can be seen at order 1, but this case can not be considered an optimum predictor because a real prediction can not be made using only the current sample.

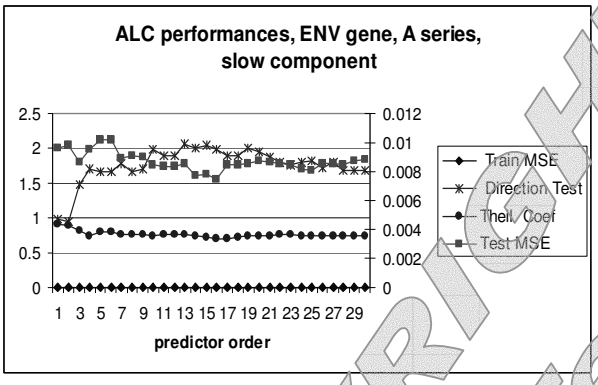


Fig. 8b - The prediction performances for different Adaptive Linear Combiner predictor orders; ENV gene, A series, slow varying component

In Fig. 8b, the case of the ALC predictors trained on the trend component of A series from ENV gene is considered. The optimum predictor might have order 16, due to global minimum observed at this order. The high value of the direction test throws susceptibility for this optimum.

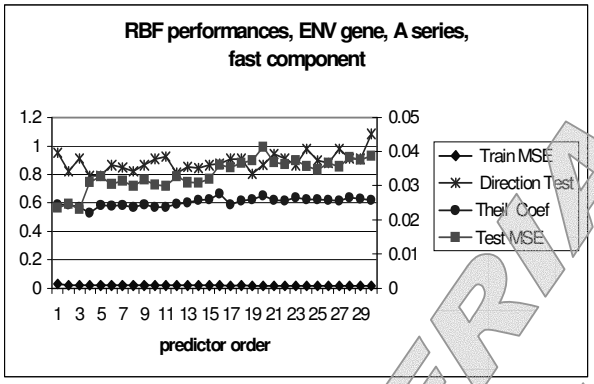


Fig. 9a - The prediction performances for different RBF predictor orders; ENV gene, A series, fast varying component

For the RBF predictors, in case of the fast component, the optimum is given at the order 3, where the global minimum is present on the test MSE series. Even if the direction test series shows a local maximum, this value is less than 1, as shown in Fig. 9a.

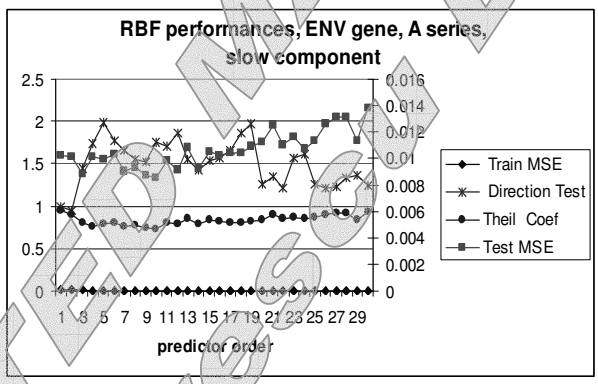
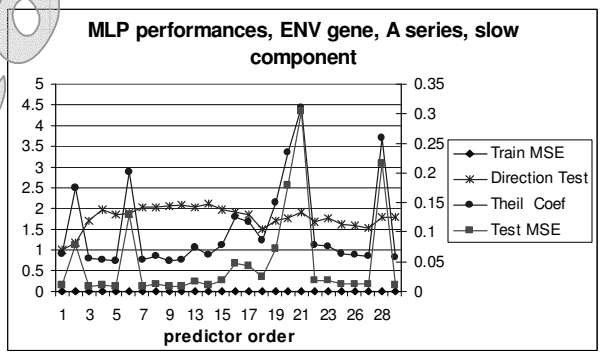


Fig. 9b - The prediction performances for different RBF predictor orders; ENV gene, A series, slow varying component

In Fig. 9b the case of RBF predictors for the trend components is presented. The optimum predictor has order 10, indicated by the global optimum existing on the test MSE series. The Theil coefficient series indicates a global minimum, too.



COPYRIGHTED MATERIAL
HNT Tecnologies Lda

Fig. 10a - The prediction performances for different MLP predictor orders; ENV gene, A series, slow varying component

In Fig. 10a, the case of MLP predictors trained on the trend component of A series from ENV gene is shown. The optimum is given for order 3, but several predictors have, like predictor with orders 5, 7, 9, and 10 have similar performances from the test MSE, direction test, and Theil coefficient point of view. The parsimony principle indicates the 3-order predictor as optimum. Notice that several predictors, corresponding to the orders 2, 6, 21, and 28, are not well trained. Also, the predictors by orders 11, 25, and 30 are not well trained and they are removed from the graphics.

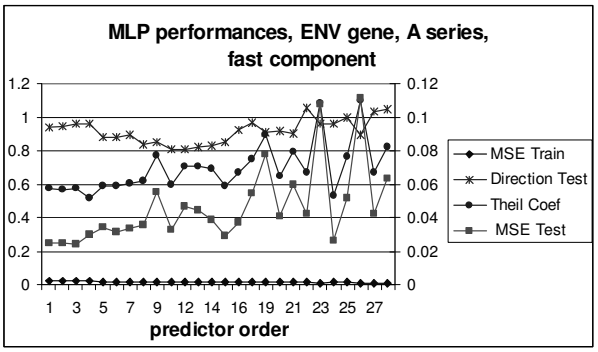


Fig. 10b - The prediction performances for different MLP predictor orders; ENV gene, A series, fast varying component

In Fig. 10b, the optimum MLP-type predictor, for the fast varying component, has order 3, due to the global minimum from the test MSE series. Several predictors, like predictors with orders 11, 18, 29, and 30, are not well trained and they are removed from the plot from Fig. 10b.

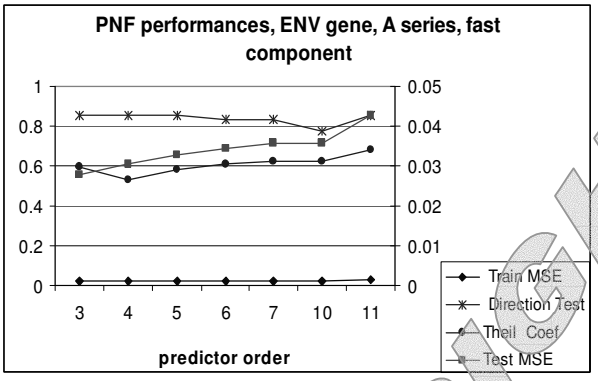


Fig. 11a - The prediction performances for different NF predictor orders; ENV gene, A series, fast varying component

In Fig. 11a, the case of NF predictors for the fast varying component is shown. The optimum predictor has order 3 and, by increasing of the predictor order, the performances do not improve.

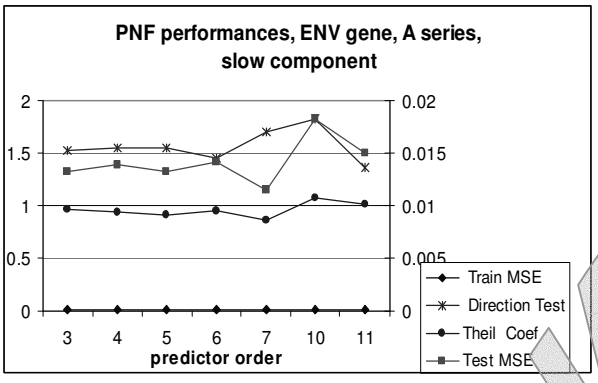


Fig. 11b - The prediction performances for different PNF predictor orders; ENV gene, A series, slow varying component

In Fig. 11b, the case of NF predictors for the slow varying component is illustrated. The optimum predictor has order 7. The predictors with order 8 and 9 are not well trained.

C. Testing the recognition power

To test the recognizing ability of the identifiers, we used a set of genomic sequence either HIV-1 (ENV, POL, REV, GAG, LTR, NEF, TAT, VIF, VPR, VPU-VPX) or other entity [22] (2L526 - Hepatitis C, CP – Mosaic Virus, GltP - Escherichia Coli, ORF3 - Hepatitis E, PyrB - Salmonella Enterica). The obtained results are presented in the Table 4.

Table 4 summarizes the recognition scores obtained by testing the MLP predictor on a set of genomic sequences with the parameters resulted from the training on the ENV gene, A series. Notice that the MSE obtained at training on the ENV gene is greater than all MSE obtained at testing on all foreign sequences.

In Table 5, we present the results obtained by training several predictors on the ENV gene. Recall that the ENV gene has been coded, according to the methodology, in four series, one for each of the bases A, C, G, T.

Table 4. The genomic sequences recognizing using MLP

		TEST: MSE	TEST: MSE	TEST: MSE
	GENE	RANDOM	TREND	CUMULATED

COPYRIGHT MATERIAL
HN Teodorescu Fira

1	ENV	0.0195	0.005	0.041
2	POL	0.0584	0.0171	0.1243
3	REV	0.0728	0.024	0.1548
4	GAG	0.0704	0.0183	0.1434
5	LTR	0.0615	0.0196	0.1368
6	NEF	0.0459	0.0122	0.0926
7	TAT	0.1226	0.0308	0.2438
8	VIF	0.0761	0.0233	0.167
9	VPR	0.1279	0.0342	0.2598
10	VPU-VPX	0.0609	0.0182	0.1378
11	2L526 HC	0.0848	0.0244	0.1839
12	CP Mosaic	0.0621	0.0178	0.1294
13	GltP E Coli	0.0286	0.0083	0.0612
14	ORF3 HE	0.0683	0.0161	0.1307
15	PyrB Salmonella	0.1234	0.034	0.2553

Table 5. Comparison of recognition ability for genomic sequences in cases of ALC, RBF, and MLP

		ALC	RBF	MLP
Random	A	0	0	**0**
	C	1	1	**0**
	G	5	3	**1**
	T	**5**	**4**	**2**
Trend	A	0	0	**0**
	C	1	1	**0**
	G	5	5	**0**
	T	**5**	**5**	**3**
Cumulated	A	0	0	**0**
	C	1	1	**0**
	G	5	4	**0**
	T	**5**	**4**	**2**

In Table 5 are counted the fails (false-positive cases) for the rejection of foreign sequences. The best results are obtained using a MLP predictor. Notice that the base T series generate the weakest indications for all predictors. Surprisingly, the NF predictors do not provide good results, despite their complexity. We have no explanation for this experimental finding.

V. CONCLUSIONS

The main goal of this paper was to determine suitable predictors for genomic sequence identifiers. We compared adaptive linear combiners, neuronal predictors (RBF or MLP), and neuro-fuzzy predictors.

Hurst exponent, self-correlation function, and eta metrics were used to appreciate the predictability of time series.

All predictors were tested and compared for prediction quality using sequences from HIV-1 genome. As prediction performance measures, we used the mean square prediction error (MSPE), direction test, and Theil coefficient.

The continuous component from the original time series is found in the trend component. For the random component, the self-correlation function graphics shown very small values, under 0.03, confirming the opinion of non-existence of a periodical component.

The computed Hurst exponent is less than 0.5 for al time series. In that case, we have series with negative self-correlation.

As we expected, the predictability of the trend component is better than the predictability of the random component for all metrics and for all prediction systems used.

The best predictor for the random component was a MLP with 0.0371 MSE for the test period. For the trend component, the maximum performance was obtained for a RBF, with MSE about 0.0188, followed by a MLP with MSE about 0.0190.

To test the recognizing ability of the identifiers, we used a set of genomic sequence either HIV or other entity. The MSE obtained at training on the ENV gene was greater than all MSE obtained at testing on all foreign sequences.

Several predictors were trained on the ENV gene, coded, according to the methodology, in four series, one for each of bases A, C, G, T. We counted the fails (false-positive cases) for the reject of foreign sequences. The best results are obtained using MLP. The bases T series generate the weakest indications.

Open questions remain:

Is the same level of predictability a good way to classify the "complexity" of sequences?

If two sequences have good prediction scores by using the same predictor, does this mean that they have some common "hidden characteristics"? If yes, what are these common characteristics?

We are not able to satisfactory answer how these questions and further thorough studies on several genetic sequences have to be performed. Also, the results presented in this paper have to be contrasted with results obtained with other prediction methods.

ACKNOWLEDGEMENTS

The CNCSIS Grant 149/2005, "System for the analysis and prediction of genomic sequences based on neuro-fuzzy data-mining methods" 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. The first author has received no grant or other financial support for the research reported here; consequently he reserves all the rights on this research.

REFERENCES

[1] C.H. Wu, J.W. Mclarty, *Neural Networks and Genome Informatics*, Elsevier, SUA, 2000.

[2] H.N. Teodorescu, "Genetics, Gene Prediction, and Neuro-Fuzzy Systems-The Context and a Program Proposal", *Fuzzy Systems & A.I. - Reports and Letters*, vol. 9, nos. 1-3, pp. 15-22, 2003.

[3] 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*, nos. 1-3, pp. 23-29, 2003.

COPYRIGHTED MATERIAL Firs

[4] L.I. Fira, H.N. Teodorescu, "Genome Bases Sequences Characterization by a Neuro-Fuzzy Predictor", In *Proc. of the IEEE-EMBS 2003 Conference*, 17-21 September, Cancun, Mexico.

[5] H.N. Teodorescu, L.I. Fira, "A Hybrid Data-Mining Approach in Genomics and Text Structures", In *Proc. of the Third IEEE International Conference on Data Mining ICDM '03*, Melbourne, Florida, USA, November 19 - 22, 2003.

[6] L.I. Fira, H.N. Teodorescu, "Analiza unor secvente de baze din genom cu un predictor neuro-fuzzy", In *Proc. Simpozionul National de Sisteme Inteligente si Aplicatii, SIA'2003*, 19-20 Septembrie 2003, Iasi.

[7] H.N. Teodorescu, L.I. Fira, "DNA Sequence Pattern Identification using A Combination of Neuro-Fuzzy Predictors", *In Proc. of the 11th International Conference on Neural Information Processing*, ICONIP2004, November 22-25, 2004 Science City, Calcutta.

[8] E.W. Weisstein. "Predictability". From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Predictability.html. © 1999 CRC Press LLC, ©1999-2005 Wolfram Research, Inc. Accessed: 31/MAR/2005

[9] E.W. Weisstein. "Redundancy". From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Redundancy.html, © 1999 CRC Press LLC, ©1999-2005 Wolfram Research, Inc. Accessed: 31/MAR/2005

[10] E.W. Weisstein. "Entropy". From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Entropy.html, ©1999 CRC Press LLC, ©1999-2005 Wolfram Research, Inc. Accessed: 31/MAR/2005

[11] J.C. Sprott, Time-Series Analysis; http://sprott.physics.wisc.edu/lectures/tsa.ppt Accessed: 31/MAR/2005

[12] H.R. Madala and A.G. Ivakhnenko, *Inductive Learning Algorithms for Complex System Modeling*, 1994, CRC Press, ISBN: 0-8493-4438-7, p. 52.

[13] E. W. Weisstein. "Hurst Exponent". From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/HurstExponent.html, ©1999 CRC Press LLC, ©1999-2005 Wolfram Research, Inc. Accessed: 31/MAR/2005

[14] SYSTAT Software Inc., AutoSignal - HTML Help http://www.systat.com/products/AutoSignal/help/?sec=1138. Accessed: 31/MAR/2005

[15] M. Kaboudan, "A Measure of Time-Series' Predictability Using Genetic Programming Applied to Stock Returns", *Journal of Forecasting*, vol. 18, pp. 345-357, 1999.

[16] M. Duan, „Time Series Predictability", Master of Science Thesis, Faculty of the Graduate School, Marquette University, Milwaukee, Wisconsin, April 5, 2002.

[17] A. Foka, "Time Series Prediction Using Evolving Polynomial Neural Networks", A dissertation submitted to the University of Manchester Institute of Science and Technology for the degree of MSc, 1999.

[18] N.R. Farnum, L.W. Stanton, *Quantitative forecasting methods*. PWS-Kent, Boston, USA, 1989.

[19] T. Hellström, "Predicting Stock Prices", *Workshop on the ML2000 Project*, Riga, University of Latvia, Latvia, November 28-29, 1997. http://www.cs.umu.se/~thomash/reports/riga-97.pdf. Accessed: 31/Mar/2005.

[20] H.N. Teodorescu, T. Yamakawa, "Neuro-Fuzzy Systems: Hybrid configurations". In vol. M.J. Patyra, D. Mlynek (Eds.): Fuzzy Logic. Implementation and applications. Wiley & Teubner, 1996, pp. 267-298.

[21] Los Alamos National Laboratory. http://hiv-web.lanl.gov/cgi-bin/ALIGN_CURRENT/ALIGN-INDEX.cgi. Accessed: 01/Mar/2005.

[22] National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=genome. Accessed: 01/Mar/2005.