F.S.A.I., Vol. 9, Nos. 1–3, pp. 23-33, 2003

Predicting the Genome Bases Sequences by means of distance sequences and a Neuro-Fuzzy Predictor

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

* Faculty of Electronics and Telecommunications, Technical University of Iasi, Iasi, Romania
** Institute for Computer Science, Romanian Academy, Iasi, Romania

Abstract: The prediction of the structure of the genes is addressed using a new method and tools, involving the sequence of distances between bases and neuro-fuzzy predictors. The method is tested on the genome of the HIV virus and the results look promising compared to other methods.

1. Introduction

Life needs a blueprint to organize the matter and generate functions. The set of chromosomes includes the blueprint recording of the organism written in the DNA. DNA is a huge molecule basically constituted out of four chained aminic bases. Genes are sections of the DNA that serve in the synthesis of proteins. DNA and consequently the genetic sequences looks like {…cgcaacgt. ccgcctgtggtc…}, where the symbols represent the codes of the aminic bases. For example, the first section of the sequence of the NC_004718 SARS coronavirus, at [1] sequence reads: {atggagagcc ttgttcttgg tgtcaacgag aaaacacacg tccaactcag tttgcctgtc …}, with a base count of 6034 a / 4128 c / 4521 g / 6538 t [2].

A large effort has been made during the last decades to unveil the genome for humans, some species of mammals, viruses, and bacteria. Many genome databases exist today, for example the NCBI *Entrez* databases, covering, beyond genome databases, nucleotide databases (dbEST MGC, dbGSS PopSet, dbSNP RefSeq, dbSTS TPA, Nucleotide Trace Archive, GenBank, UniGene, HomoloGene UniSTS), protein databases, structure databases, taxonomy databases, and expression databases (see [3]). As in May 2003, the whole genomes of over 1000 viruses and over 100 microbes can be found in the *Entrez* databases. According to *Entrez*, "All three main domains of life – bacteria, archaea, and eukaryota, as well as many viruses and organelles" are included in the database.

Because of the tremendous implications, during the last decade, genomics became both a leading edge and a highly demanding science, asking for and imposing advances in many fields, including biomedical engineering and computer science. Since 1990, the U.S. Human Genome Project aimed to determine all the (about) 30,000 genes in human DNA and the sequences of about 3 billion chemical

base pairs that make up human DNA. Moreover, beyond storing and distributing this information, the project aims to develop appropriate tools for data analysis, a task not yet fulfilled. (That project also aims to "transfer related technologies to the private sector, and [to] address the ethical, legal, and social issues (ELSI) that may arise from the project", according to [4]). Similar challenges face any other similar project in genomics. While determining the base sequence is the obvious first step, analyzing the sequence is a much more work-intensive goal and the reward for determining the base sequences is measured in the ability to interpret these sequences and use the derived information.

Determining the base sequence is only the initial step in using the genetic information. Most (90%) of the genetic information does not relate to genes. Finding the gene sections (about 10% of the complete base sequence), and other sections of interest and determining their function, that is the protein they help synthesize ('express") is the next step. This task is huge and can not be performed manually. Automatically determining what regions of the sequence represent genes and what their functions are is named genetic prediction.

Currently, several methods are used to make predictions, including repeated elements searching, functional signals prediction, and dicodon statistics (see for example the "GeneBuilder" description, [5]). Other methods proposed to analyze and predict the structure of genes are based on formal grammars and syntactic pattern recognition. The grammars are organism-specific.

The prediction methods rely on knowledge of the genes for a specified class; thus, methods and results are organism specific. Recall that functional proteomics describes the proteins and protein networks that underlie the basic biological processes.

Among others, essential tasks for the computer scientist is to develop programs able to find specific patterns in the sequence and to predict a sequence. The prediction, at the current stage, is used to help the analysis of the structure, taking into account the huge amount of data that is to be analyzed.

The HIV viruses have several subtypes, which are differentiate based on the sequences of the ENV gene. The major group (with 8 subtypes denoted with letters from A to H) and the group O (outgroup) with 3 subtypes (O.1, O.2 and O.3) constitute the HIV1. HIV2 several subtypes, denoted with letters from A to E. The viral RNA is constituted of three genes, that are common to all the retroviruses, moreover of specific genes. The three typical genes are named GAG, POL, and ENV. The specific genes are named TAT, REV, VIF, NEF and VPX. There is a significant genomic variability of the HIV, during the infection, and from one subject to another. The highest variability for the typical genes is seen for the ENV [6].

The sequence data sets used here are the primary input. We used the nucleotide sequence from the region ENV from HIV-1 "B.FR.83.HXB2", available at [7].

2. Methodology

2.1. Coding of the sequence

In a previous paper [8], we have presented the underlying principles for the system presented in this paper. Here, we detail the method used. Because the codes of the genes are letters and have no numerical representation, in order to use a numerical prediction algorithm, we have to produce a "translation" from the symbol to numbers for the sequence. The method of numerical coding is important, as the prediction results may heavily depend on it. We have first tried a direct numerical representation of the bases, using the rank in the alphabet of the corresponding letters, with normalization: A=1/26, G= 7/26 etc. This coding yielded very poor prediction results. Therefore, we have applied a new method, first used by the first author in the prediction of words in natural languages [8]. Namely, we coded the sequence as a set of four sequences, each constituted by the distances between successive occurrences of the basis. For example, the sequence

{atggagagcc ttgttcttgg tgtcaacgag aaaacacacg tccaactcag tttgcctgtc …}

is coded as:

A = {4,2,18, …}; , C={1,6,8,…}, G={1,2,2,…}, T={9,1,2,…}

Then, a predictor is developed to generate each of the four sequences A,C,G, and T.

2.2. The predictor

We have used a neuro-fuzzy predictor, who has been developed and tested, in our group, in previous researches [9]. The architecture for one step predictive system is shown in Fig. 1.

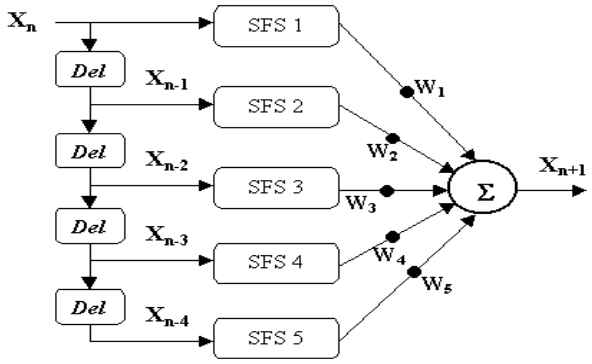

Fig. 1 – The topology of transversal type filter with Sugeno fuzzy system network [9]

The *Del* symbol stands for the delay operator and ensures one iteration delay between the samples delivered towards the input of the consecutive SFS1...SFS5 Sugeno systems inputs. The Sugeno fuzzy system with single input and single output was chosen for the elementary cells of the network of systems. The inputs of the fuzzy systems are characterized by seven Gaussian type membership functions, presented in Fig. 2.

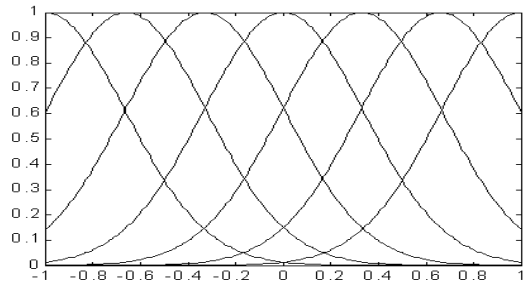

Fig. 2 – The membership functions for the input of Sugeno fuzzy system.

These seven Gauss-type membership functions, identical for all Sugeno systems, are defined by two parameters, namely "center" and "sigma", with the following values: *center* ∈ {-1.0, -0.66, -0.33, 0, 0.33, 0.66, 1.0}, *sigma* = 0.23. For an input x, the membership function values are:

$$h_{kl}(x) = e^{-\frac{(x-a_{kl})^2}{\sigma}} \tag{1}$$

Although the graphics for the membership functions are represented only for the interval [-1, 1], the input domain is (-∞, ∞).

2.3. The equations for the neuro-fuzzy predictor

We denote by M the number of Sugeno fuzzy systems, N the number of membership functions for each Sugeno fuzzy system, index $k = 0 \div M$, index $l = 1 \div N$, a_{kl} – the centers of the Gauss functions[z1], β_{kl} – the singletones, h_{kl} – the degree of belief, and w_k – the weights.

If the input is x, then the output for the Sugeno fuzzy system # k is:

$$Y_k(x) = \frac{\sum_{l=1}^{N} h_{kl} \cdot \beta_{kl}}{\sum_{l=1}^{N} h_{kl}} \tag{2}$$

The characteristic function of the predictor is:

$$Y = \sum_{k=0}^{M} w_k \cdot Y_k\left(x_{n-k}\right) \tag{3}$$

After straightforward computations, we obtain:

$$Y = \sum_{k=0}^{M} w_k \cdot \frac{\sum_{l=1}^{N} h_{kl}\left(x_{n-k}\right) \cdot \beta_{kl}}{\sum_{l=1}^{N} h_{kl}\left(x_{n-k}\right)} \tag{4}$$

and after straightforward computations,

$$Y = \sum_{k=0}^{M} w_k \cdot \frac{\sum_{l=1}^{N} \beta_{kl} \cdot e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}}{\sum_{l=1}^{N} e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}} \tag{5}$$

Equation (5) stands for the input-output function of the neuro fuzzy predictor.

2.3. Training the predictor by error minimization

By definition, the error is:

$$\varepsilon_n = \hat{x}_{n+1} - x_{n+1} \tag{6}$$

The mean square error (L_2 criterion) is:

$$L_2 = \frac{1}{T} \sum_{n=1}^{T} \varepsilon_n{}^2 = \frac{1}{T} \sum_{n=1}^{T} \left(\hat{x}_{n+1} - x_{n+1}\right)^2 \tag{7}$$

The optimum is obtained by minimizing the error,

$$\frac{\partial L_2}{\partial \beta_{kl}} = 0, \ \frac{\partial L_2}{\partial w_k} = 0. \tag{8}$$

The complete formula for L_2 is:

$$L_2 = \frac{1}{T} \sum_{n=1}^{T} \left[x_{n+1} - \sum_{k=0}^{M} w_k \cdot \frac{\sum_{l=1}^{N} \beta_{kl} \cdot e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}}{\sum_{l=1}^{N} e^{-\frac{(x_{n-k}-a_{kl})^2}{\sigma}}} \right]^2 \tag{9}$$

For example, the second equation is:

$$\frac{\partial L_2}{\partial w_k} = -\frac{2}{T}\sum_{n=1}^{T}\left\{Y_k\left(x_{n-k}\right).\left[x_{n+1} - \sum_{k=0}^{M} w_k \cdot Y_k\left(x_{n-k}\right)\right]^2\right\} \tag{10}$$

The above equations are used in the gradient algorithm for adaptation of the neuro-fuzzy predictor. The prediction is one-step ahead.

3. Results

3.1. Statistics

The basic statistic of the distances for the basis *A* is shown in Table 1, and the corresponding histograms are shown in Fig. 3.

Table 1

Statistical properties of the sequence of distances between subsequent occurrences of the basis A, C, G, T.

	A Basis series	C Basis series	G Basis series	T Basis series
Average	2.776	6.026	4.327	4.107
Spreading	2.445	6.201	4.117	3.871
Mode	1	1	1	1
Median	2	4	3	3
Skewness	3.160	2.200	2.021	2.418
Kurtoisis	16.902	7.103	4.639	8.213
Max	24	46	25	27
Sum	2079	2073	2077	2078

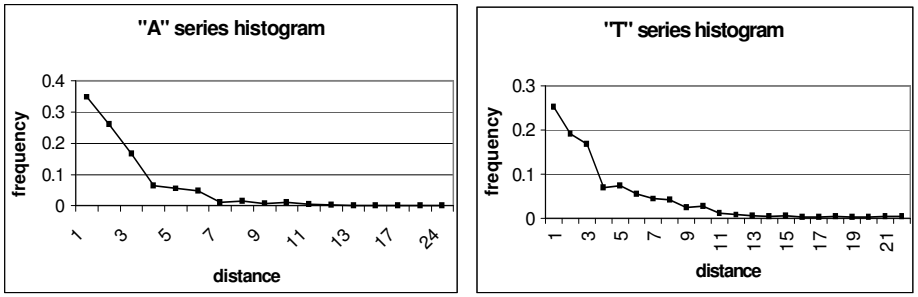

Fig. 3 – Statistics of distances for the Adenine and Tymine basis, in the use sequence

Notice that the statistics of the four bases are highly asymmetric, with averages spanning a range larger than 1:2. The distributions broadly follow the "Zipf law".

3.2. Component decomposition

Any time series may include a slowly changing component, usually named the trend component, an almost periodical component, usually named cyclic or seasonal component, and a non-periodic, fast variable component, originating from a stochastic or nonlinear dynamic process. Using a predictor for each of the components is known to significantly enhance the prediction outcome. Therefore, we have first decomposed the time series into the trend component $y_t[n]$, the cyclic component $y_c[n]$, and the fast varying component $y_a[n]$,

$$y[n] = y_t[n] + y_c[n] + y_a[n] \tag{11}$$

The trend component is obtained by a moving average procedure:

$$y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1]) \tag{12}$$

We tested for the cyclic component by applying the self-correlation procedure to the series $y[n] - y_t[n]$; the self-correlation is well known to evidence periodicity. Because the cyclic component has been found insignificant, the result of the subtraction $y[n] - y_t[n]$ has been dealt with as a random component. After normalization, these two series have been separately predicted using the same number of samples for the train and test periods. The values are then de-normalized, such that the results obtained from the two predictions are compatible. The denormalized results of the two independent predictions have been added to obtain the original series prediction. An example of results is shown in Fig. 4 – 7. The average error (normalized mean square error – NMSE) in distance prediction has been 0.513 for the *A* basis, 0.927 for the *T* basis, 0.488 for the *G* basis.

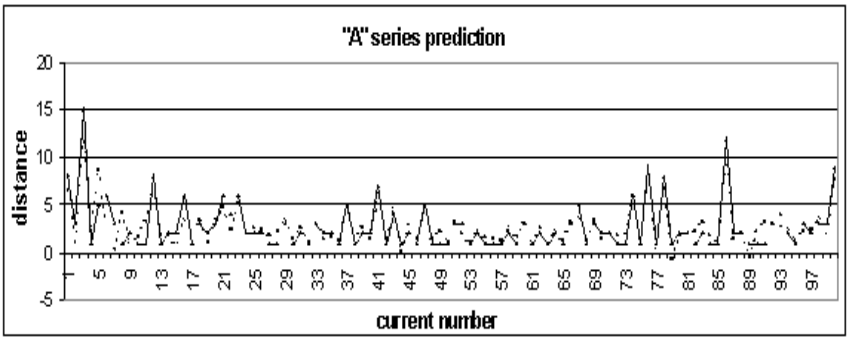

Fig. 4 – The time series corresponding to the distances for the A basis and the result of its one-step ahead prediction

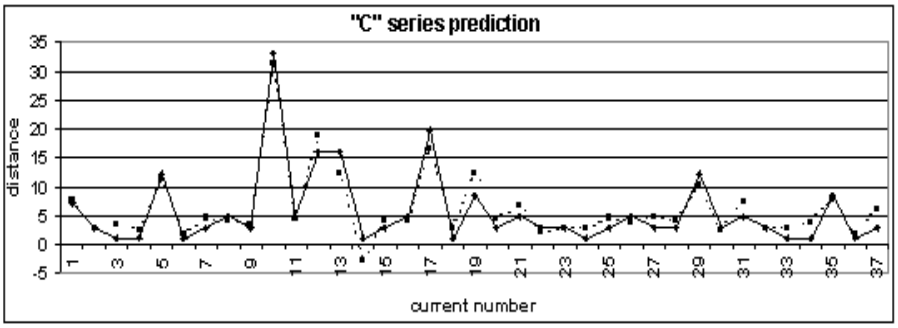

Fig. 5 – The time series corresponding to the distances for the C basis and the result of its one-step ahead prediction

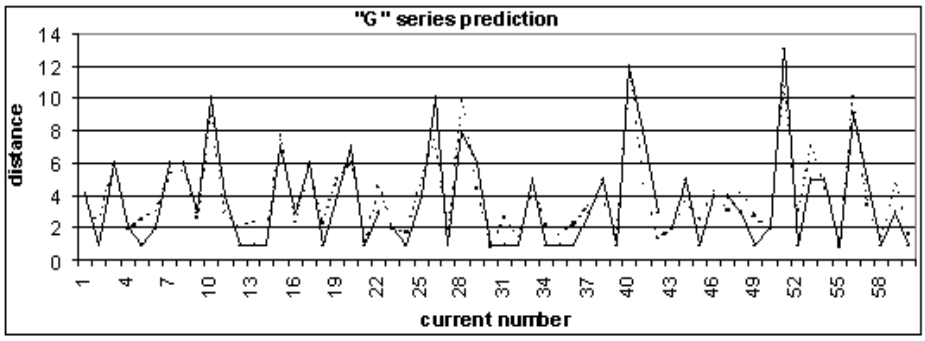

Fig. 6 – The time series corresponding to the distances for the G basis and the result of its one-step ahead prediction

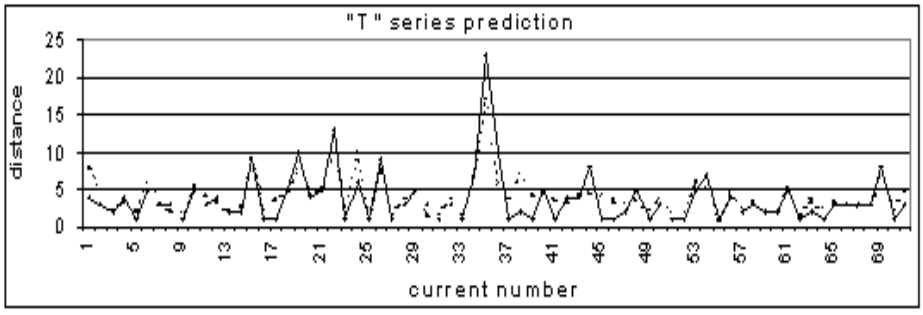

Fig. 7 – The time series corresponding to the distances for the T basis and the result of its one-step ahead prediction

3.3. Comparison

For comparison, we have used the prediction software package [10], freely available on the Internet. [E2]

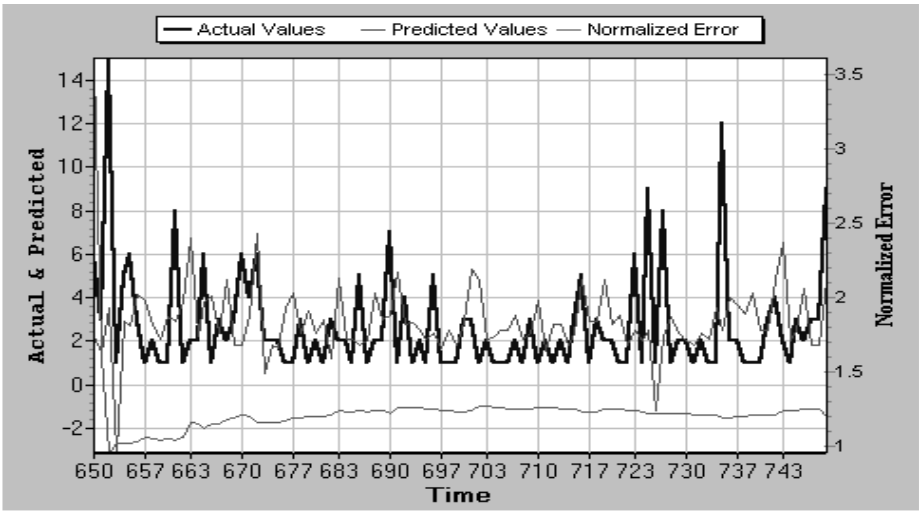

Fig. 8 – The time series corresponding to the distances for the *A* basis and the result of its one-step ahead prediction using the predictor in [10].

The results obtained with several prediction methods available in this software are significantly poorer than the results obtained with our predictor. In Fig. 8, a sample of prediction results obtained with the VRA package is shown. The method used in the VRA package is based on the nonparametric modeling, which consists in directly deriving the model from given data. In VRA, this is done using local polynomial models.

The parameters chosen to construct the model to generate predictions are: type is one-step ahead, predictor is radial basis, RBF is Gaussian, the distance is Euclidean, and 10 neighbors are used for the train period, we use samples between 1 and 650 and for test sequence, samples between 651 and 749.

A normalized error about 1.195 for test period in case of basis *A*, 1.107 in case of basis T and 0.965 for basis *G*. That was the best performance which we obtained with the VRA package. A comparison between the performance of the neuro-fuzzy predictor and prediction with VRA is presented in Table 2, where RMSE denoted root mean square error.

Table 2

Comparison between neuro-fuzzy predictor and prediction performed with VRA

Basis	Error type	NFP	VRA
A	NMSE	0.513	1.195
	RMSE	1.127	2.69
C	NMSE	0.566	1.057
	RMSE	1.930	7.201
T	NMSE	0.927	1.107
	RMSE	1.803	3.72
G	NMSE	0.488	0.965
	RMSE	1.214	4.602

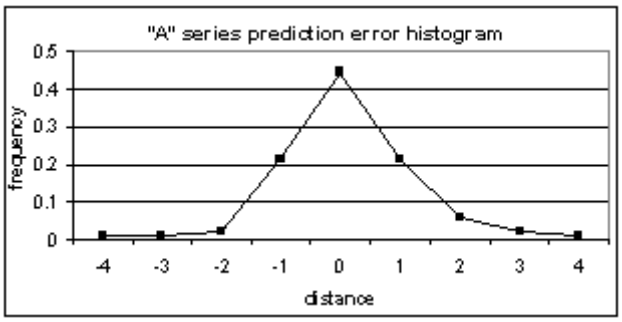

Fig. 9 – Histogram of the prediction error for A basis

The histogram of the errors as show in Fig. 9 is close to a Gauss function. That indicates that most of the relevant information in the data has been used by the predictor.

4. Discussion and conclusions

We have proposed a modified method to deal with the prediction of bases sequences, converting the sequence of bases in several sequences, each for a single basis, according to [11]. Moreover, we use a neuro-fuzzy predictor to perform the prediction; then, the distances are converted back to current positions of the individual bases, and the complete sequence is reconstructed. The method yields better results than those obtained with several other predictors.

We have found that the conversion to distances much enhances the results. Actually, we also tried the method of first converting the bases symbols into numerical values like $A\rightarrow$ 1, $C\rightarrow$2, $G\rightarrow$3, $T\rightarrow$4 moreover $A\rightarrow$1, $C\rightarrow$2, $G\rightarrow$4, $T\rightarrow$8, but the results obtained were poor.