International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.5 (2016), pp.243-252
http://dx.doi.org/10.14257/ijsip.2016.9.5.21

Efficient Exonic Regions Prediction in DNA Sequence Using Fast Converged Adaptive Filter

Y. Murali Krishna[1], K. Murali Krishna[2] and Ch. Amaranatha Sarma[3]

*[1]Asst. Professor, Dept. of Electronics & Comminication Engineering
Kallam Haranatha Reddy Institute of Technology, A.P, India.
[2,3] Asst.Professor, Dept. of Electronics & Comminication Engineering,
Gudlavalleru Engineering College, Gudlavalleru, A.P, India.
[1]muralikhit123@gmail.com, [2]muralikondaveeti1@gmail.com,*

Abstract

Signal processing takes an important role in genomic with enormous data available in public domain. Generally digital filters are applied to predict the protein and genes, but it needs to be redesigned when the characteristic frequency and periodic behavior is changed. In this paper proposed the novel adaptive algorithm which can identify the genes and proteins effectively from unified framework. First using the electron ion potential method the symbolic DNA sequences are converted in to digital signal. Secondly the filtering scheme for genomic signal processing with periodic behavior in biological sequence is introduced, which can predict and analyze the biological region that are interested in. finally the proposed adaptive filtering method is applied to recognize the exons of protein coding regions according to periodic-3 property. The exons prediction curves are obtained with Discrete Fourier Transform (DFT), Least mean square (LMS), and proposed Fast Recursive least Mean Square (F-RLS) algorithms. It is shown that proposed method shows efficiency in convergence of identification and precise prediction of exons regions compared to existed methods.

***Keywords**: Exons, DNA, Genomic Signal Processing, LMS, F-RLS*

1. Introduction

Genomic Signal Processing (GSP) is the engineering discipline that studies the processing of genomic signals. It encompasses several methodologies for expression profiles: detection, prediction, classification, control and statistical and dynamic models of gene networks [1]. It can be a very useful tool for processing large genomic and proteomic data [2-6] and [9]. These data contain deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and protein sequences. It is necessary to process DNA, RNA and protein sequences to identify segments of special biological, such as exons, introns and hotspots. A DNA sequence comprises four different nuclides (or bases) named as A, T, C and G. By mapping the alphabetic sequence of a DNA chain into a set of digital signal processing (DSP) techniques based on digital signals can be applied to analyze the DNA sequence.

At present, traditional and modern signal processing techniques play an important role in these fields. For example, the Fourier transform has been applied to identify exons in genes. To improve the traditional Fourier analysis performance in distinguishing coding from non-coding regions in a DNA sequence an optimization procedure has been used. The Digital filters have been applied to the gene prediction. The multi-stage filter is designed to process the pointer sequences, in order to reduce locale noise and to obtain the curve prediction coding regions. DSP other methods have been applied to identify protein coding regions .In hot spots on proteins they have been predicted according to the characteristic frequency of the protein sequence.

2. Numerical Mapping of DNA Sequence

For the identification of protein-coding regions we have to apply suitable signal processing methods, the character string of the DNA sequence is changed to a suitable numerical sequence. This is achieved by assigning a numeral to each nucleotide that forms the DNA sequence. Therefore, various techniques have been suggested to achieve this particular conversion. The goal of each encoding method is to improve the hidden information for further analysis. One of the most extensively used methods is adaptive filters [1], where the character string of DNA is converted to four binary pointer sequences for each base (A, T, C and G). It assigns a digit “1” when a particular symbol is found in the sequence, otherwise a “0”. Anastassiou [2] has proposed a complex number mapping by conveying a particular complex number to each base. Silverman and Linsker [4] have used a tetrahedron mapping, in which each nucleotide is assigned to one of the four corners of a regular tetrahedron. Chakravarthy [3] have proposed a real number mapping of the DNA sequence. Zhang [5-6] presented a Z-curve mapping, which is a three-dimensional curve representation for the DNA sequence. Recently, Nair [7-8] have used an EIIP indicator sequence to map the character string of DNA to numeric form. The EIIP is defined as the average energy of delocalized electrons of the nucleotide. Assigning the EIIP values to the nucleotides, a numerical sequence is obtained to represent the distribution of the free electrons’ energies along the DNA sequence. This has been effectively used to identify hot spots in proteins, for peptide design and also for identification of coding regions [7-9]. The EIIP sequence is a better choice for numerically representing DNA when compared to indicator sequences for the following reasons. First, it involves only a single sequence instead of four in the case of binary indicator sequences, thereby reduces the computational complexity. Second, it is biologically more meaningful as it represents a physical property when compared to the indicator values, which represent just the presence or absence of a nucleotide. Hence in this paper, we have also used the EIIP representation method of numerical mapping of DNA sequence. The DNA sequence can be converted to the numerical sequence by replacing each nucleotide with the corresponding EIIP value. The EIIP values for the nucleotides are given in Table 1. For example, if x[n]=AATGCATCA, then using the values from Table 1, the corresponding EIIP numerical sequence is given as: x[n]=[0.1260 0.1260 0.1335 0.0806 0.1340 0.1260 0.1335 0.1340 0.1260].

Table 1. The EIIP Values of Nucleotides

Nucleotide	EIIP value
A	0.1260
T	0.1335
G	0.0806
C	0.1340

2.1. Spectral Content Measure Method

In this frequency domain method, the discrete Fourier transform (DFT) of the EIIP indicator sequence is employed. Let *X(k)* represents the DFT of the corresponding EIIP numerical sequence and is given by:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}$$

For $k = 0, 1, \ldots, N$-1.
Then the spectral content at k^{th} instant is:

$$S[k] = |X[k]|^2 \quad (2)$$

$S[k]$ acts as a preface indicator of a coding region giving a peak at the $N/3$ frequency. This procedure is used to detect the probable coding regions in the DNA sequence. Hence the coding regions are identified by evaluating $S[N/3]$ over a window of N samples, then sliding the window by one or more samples and recalculating $S[N/3]$. This process is carried out over the entire DNA sequence. The peaks in the spectra obtained by the sliding window DFT correspond to the protein-coding regions. It is essential that the window length N be adequately large (typical sizes are a few hundred to a few thousand), so that the periodicity effect dominates the background noise spectrum. This approach increases the computational complexity as it computes the spectrum within a window and is also constrained by the frequency resolution and spectral leakage effects of the windowed data record.

3. The Adaptive Filtering Approach for Genomic Signal Processing

The adaptive filtering method for predicting biological function segments is shown in Figure 1. Let the input signal of the filter be denoted by *x(n)* , the output be *y(n)*, the desire response be *d(n)* and the error be *e(n)*. We choose the desired signal *d(n)* in terms of the period-3 behavior. The error *e(n)* between output *y(n)* and desired response signal *d(n)* is used to regulate the weights vector of the filter.

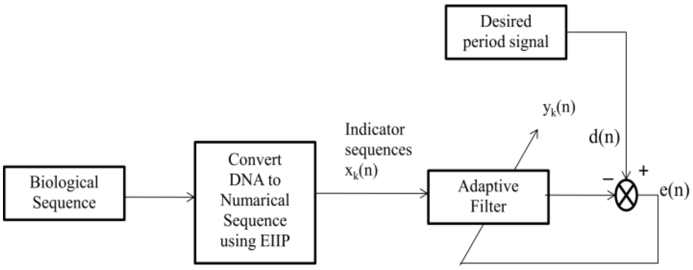

Figure 1. Adaptive Filtering Method for Predicting Biological Function Segments

The four indicator sequences are used as the input signals $x_k(n)$ of the adaptive filter $k \in \{A, G, T, C\}$.The output signals $y_k(n)$ are obtained. The four output signals all have contribution to the prediction, so we define the sum output *y(n)* as

$$y(n) = |y_A(n)|^2 + |y_G(n)|^2 + |y_T(n)|^2 + |y_C(n)|^2 \quad (3)$$

It has been noticed that period-3 property exists within the exons (coding regions inside the genes) for eukaryotes (cells with nucleus) and does not exist within the introns (non-coding regions in the genes) because of coding biases in the translation of codons into amino acids. In this paper, period-3 property is applied to find the protein coding segments (exons) in a DNA sequence. According to theperiod-3 property of protein coding regions, the desired signal is generated by sinusoidal functions in *f(k)* with the frequency f=2Π/3, k=0,1,2,3… which is desired periodic signal.

3.1. The DFT

The Discrete Fourier Transform (DFT) is the equivalent of the continuous Fourier Transform for signals known only at N instants separated by sample times T(*i.e.* a finite sequence of data)[10-15].

Let *f(t)* be the continuous signal which is the source of the data. Let N samples be denoted *f(0), f(1), f(2), f(3),…,f(k),…,f(N-1).*

The Fourier Transform [16] of the original signal, f(t), would be

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{4}$$

We could regard each sample *f(k)* as an impulse having area *f(k)*. Then, since the integrand exists only at the sample points:

$$\mathrm{F(j\omega)} = \int_{0}^{(\mathrm{N-1})\mathrm{T}} \int_{-\infty}^{\infty} \mathrm{f(t)e^{-j\omega t}dt} \tag{5}$$

$$= \mathrm{f(0)e^{-j0} + f(1)e^{-j\omega T} + \cdots + f(2)e^{-j\omega kT} + \cdots f(N-1)e^{-j\omega(N-1)T}}$$

i.e.,

$$F(j\omega) = \sum_{k=0}^{N-1} f(k)e^{-j\omega kT} \tag{6}$$

We could in principle evaluate this for anyω, but with only N data points to start with, only N final outputs will be significant. The continuous Fourier transform could be evaluated over a finite interval (usually the fundamental period T_0) rather than from $-\infty$ to $+\infty$ if the waveform was periodic. Similarly, since there are only a finite number of input data points, the DFT treats the data as if it were periodic (*i.e. f(N)* to *f(2N-1)* is the same as *f(0)* to *f(N-1)*.

3.2. Least-Mean Square (LMS) Algorithm

Generally adaptive filter algorithms are implemented in an iterative or recursive manner. With each iteration, they improve the performance of the filter according to the criteria of performance. Specifically if we let $\mathrm{w_n}$ to be a vector of length N whose elements are the time-varying coefficients of an FIR filter at time index n, then the adaptation mechanism of the vector can be given by

$$\mathrm{W_{n+1} = W_n - \mu[\nabla_W J_W]} \tag{7}$$

where μ is the learning parameter called the step-size that controls the amount of correction applied to the weight vector in each iteration and ensures the convergence of the adaptive algorithm. The term $\nabla_{\mathbf{W}}\mathrm{J}_{\mathbf{W}}$ is the gradient of the cost function with respect to the weight vector of the algorithm. (7) represents a special approach in adaptive algorithms called the steepest descent [1]. In this approach, the solution to the problem, *i.e.*, the filter that best suits the application is sought in the direction where the cost function is minimized. Hence the term $\nabla_{\mathbf{W}}\mathrm{J}_{\mathbf{W}}$ is given as

$$\nabla_{\mathbf{W}}\mathrm{J}_{\mathbf{W}} = \frac{\partial \mathrm{J}}{\partial \mathbf{W}} = \frac{\partial \mathrm{J}}{\partial \mathbf{W}}\mathrm{E[g(e_n)]} \tag{8}$$

where $\mathrm{g(e_n)}$ is a function of the error and consequently

$$\mathrm{J_W = E[g(e_n)]} \tag{9}$$

is the cost function. According to the steepest descent approach, the gradient of the cost function can be written as

$$\mathrm{\nabla_W J_W = R_x W_n - R_{dx}} \tag{10}$$

where $\mathrm{x_n}$ is the tap input regressors taken from the input sequence $\mathrm{\{x_n\}}$, $\mathrm{R_x = E[X_n X_n^T]}$ is the autocorrelation of the input regressor and $\mathrm{R_{dx} = E[X_n d_n]}$ is the cross-correlation of the input regressor and the desired response. The expectation that appears in (8) requires the knowledge of the statistics of the tap input vector that, in practice, is not available. Hence the gradient has to be estimated by dropping the expectation and taking the sample value of the tap input. This introduces a randomness or stochastic behavior to all such adaptive algorithms and is hence termed stochastic gradient algorithms. The approximation made for (8) can be further explored where

$$-\hat{\nabla}_W J_w = \hat{R}_{dx} - \hat{R}_x W_n$$

$$= X_n [d_n - X_n^T W_n]$$

$$-\hat{\nabla}_W J_w = X_n e_n \tag{11}$$

or in general it can be setup as

$$-\hat{\nabla}_W J_w = X_n g(e_n) \tag{12}$$

Where $g(e_n)$ is a function of the error, $\hat{\nabla}_W J_w$ can be viewed as the gradient applied to the instantaneous error function, and

$$\hat{R}_{dx} = d_n X_n \ , \hat{R}_x = X_n X_n^T$$

are the instantaneous cross-correlation between the desired and tap input vector and the autocorrelation of the tap input vector, respectively. Substituting (12) in (7) will give us the generic adaptation equation for stochastic gradient algorithms as

$$W_{n+1} = W_n + \mu X_n g(e_n) \tag{13}$$

where μ is the step-size of the algorithm. Selecting the appropriate error function, g (en), will yield different algorithms that behave totally differently and require extensive analysis for their proper behavior to be characterized.

By far the most popular stochastic gradient algorithm is the Least-Mean Square (LMS) algorithm [17-19] developed by Widrow and Hopf. The name signifies its cost function as the minimization of the mean squared error. The error function for LMS is given by $g(e_n) = e_n$, the LMS weight update recursion will be

$$w_{n+1} = w_n + \mu X_n e_n \tag{14}$$

Due to its simple yet elegant mathematics, this algorithm has been extensively used in various applications. The most motivating factor of its usage is its simplicity of implementation. The LMS algorithm achieves good performance characteristics when the conditions of operation are rightly suited for it. This performance is seen where the noise environment is Gaussian. Even then there are limitations to its performance that has seen its fair share of research. The motivation behind improving the LMS algorithm is its slow convergence rate and higher steady-state error. The slow convergence of the LMS algorithm is due the fact that it is based on the minimization of the mean-squared error and is only dependent upon the second order moment of the noise; this results in identical convergence rates in various noise environments.

Also the steady-state error of the LMS algorithm is dependent on the second order moment of noise which, as we shall see later, compared to algorithms based on higher order moments of error, results in higher steady-state error. Moreover, the steady state error and rate of convergence are highly dependent upon the step-size of the algorithm. In fact the steady-state error is inversely proportional to the step-size parameter. This highlights a compromise that has to be made in every design of fixed step-size algorithms.

3.3. Fast RLS Algorithm

In general the problem of system identification involves constructing an estimate of an unknown system given only two signals, the input signal and a reference signal. Typically the unknown system is modeled linearly with a finite impulse response (FIR), and adaptive filtering algorithms are employed to iteratively converge upon an estimate of the response. If the system is time-varying, then the problem expands to include tracking the unknown system as it changes over time. The system identification problem has numerous applications in control systems, digital communications, and signal processing, and a recent survey of adaptive filtering algorithms highlights the rich diversity of techniques available in the literature [1]. Adaptive filtering has been, and still is, an area of active research, playing important roles in an ever increasing number of applications [1-20]. Numerous algorithms for the solution of the adaptive filtering problem have been proposed over the years. The recursive least squares (RLS) algorithms are used in a broad class of applications. The RLS algorithm solves this problem, but at the expense of increased computational complexity. A large number of fast RLS (FRLS) algorithms have been developed over the years, but, unfortunately, it seems that the better a FRLS algorithm is in terms of computational efficiency, the more severe is its problems related to numerical stability [21]. Several numerical solutions of stabilization, with stationary signals, are proposed in the literature [22-27].

The main identification block diagram of a linear system with finite impulse response (FIR), by adaptive filtering using an adaptation algorithm, is represented in Figure 2.

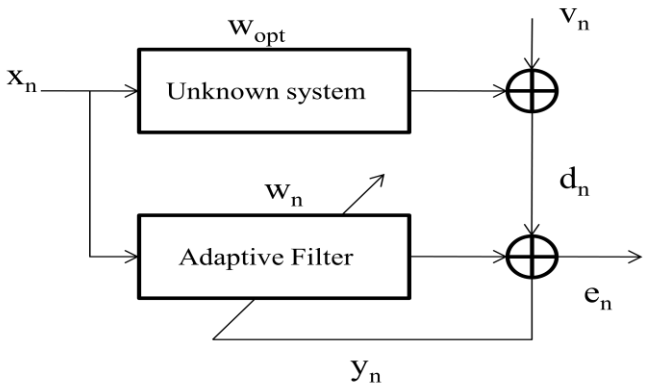

Figure 2. System Identification Block Diagram

The output a priori error e_n of this system at time n is:

$$e_n = d_n - y_n \tag{15}$$

Where

$$y_n = w_{n-1}^T x_n \tag{16}$$

is the model filter output, $x_n = [x_{n-1}, x_{n-2}, \dots, x_{n-L+1}]^T$ is a vector containing the last L samples of the input signal x_n, $w_{n-1} = [w_{1,n-1}, w_{2,n-1}, \dots, w_{L,n-1}]^T$ is the coefficient vector of the adaptive filter and L is the filter length. We assume that the desired signal from the model is:

$$d_n = v_n + w_{opt}^T x_n \tag{17}$$

Where $w_{opt} = [w_{opt,1}, w_{opt,2}, \dots, w_{opt,L}]^T$ is the unknown system impulse response vector and v_n is a stationary, zero-mean, and independent noise sequence that is uncorrelated with any other signal. The superscript $(.)^T$ describes transposition.

The filter w_n is calculated by minimizing the weighted least squares criterion according to [1]:

$$J_n(w) = \sum_{i=1}^{n} \lambda^{n-1} (d_i - w_n^T x_i)^2 \tag{18}$$

where λ denotes the exponential forgetting factor ($0 < \lambda \leq 1$). The recursive solution is written as follows:

$$w_n = w_{n-1} + g_n e_n \tag{19}$$

Where g_n represents the adaptation gain, given by:

$$g_n = R_n^{-1} x_n = \gamma_n \bar{k}_n \tag{20}$$

With

$$R_n = \sum_{i=1}^{n} \lambda^{n-i} x_i x_i^T = \lambda R_{n-1} + x_n x_n^T \tag{21}$$

The quantity R_n is the L-by-L sample covariance matrix of the input signalx_n. The variables γ_n and $\bar{k}_n$ respectively indicate the likelihood variable and normalized Kalman gain vector. This latter is calculated, independently of the filtering part w_n, by a FRLS algorithm using forward/backward linear prediction analysis over the signalx_n. The calculation complexity of a FRLS algorithm is of order L. This reduction in computation complexity.

4. Results

In order to demonstrate the performance of the methods, we apply them on some gene sequences. AF019074, AJ223321.1 and AF009962 from Gen bank database. AF009962 is the accession number for single exon, which has one coding region at position 3934-4581. The gene sequence AF019074.1 has the length of 6350, which has three distinct exons, 3187-3500, 3761-4574 and 5832-6007. There is one coding region existed in AJ223321.1 gene sequence, which its location is 1196-2764. All mentioned sequences are converted to numerical sequences using EIIP method.

In this paper, to compare the performance of the proposed adaptive algorithm plotted in Figure 3, Figure 4 and Figure 5. From the figures, the exons can be seen and the background noise is removed. Compared to the real locations of the exons , there exists some error to predict the actual exons locations using DFT method. Figure 4, 5 and 6 shows the predictive results of the DFT, LMS and FAST-RLS adaptive algorithm. The peaks of the exons can be seen clearly using LMS, FAST-RLS the background noise is removed largely Compared to the DFT method. FAST-RLS algorithm shows the better prediction over LMS algorithm slightly but the rate of estimating prediction by filter coefficients is very fast that has shown in Figure 2.All the predictive results of different sequences at different segments results specified in Table 2. From all these contents it has to justifies that our algorithm predicts exons regions some what more accurately than previous nethods.

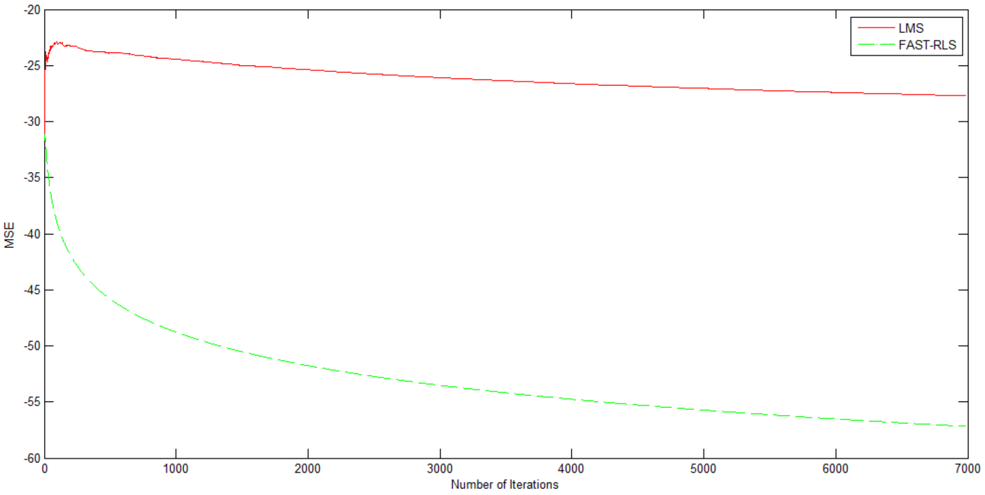

Figure 3. Convergence Plot for LMS and FAST-RLS Algorithms

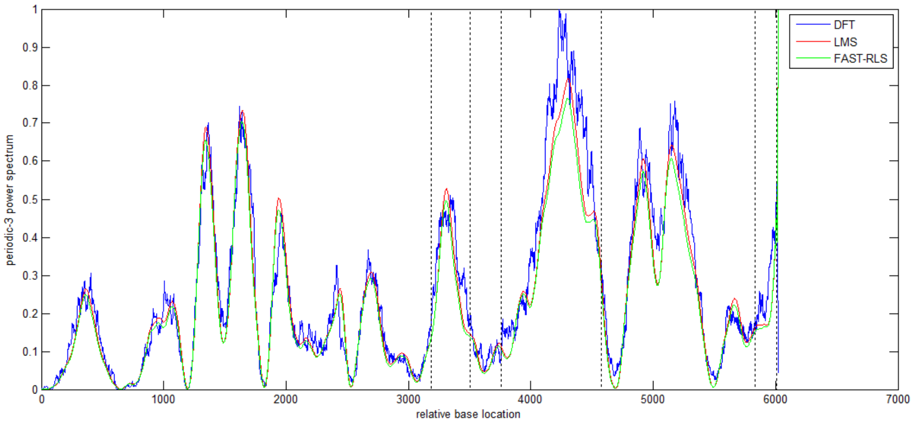

Figure 4. Power Spectra of the Gene AF019074 Obtained By DFT, LMS and FAST-RLS Algorithms

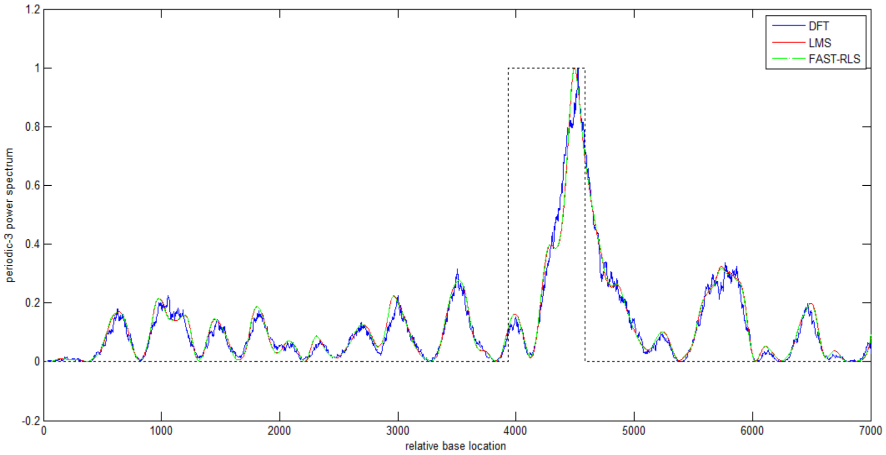

Figure 5. Power Spectra of the Gene AF009962 Obtained By DFT, LMS and FAST-RLS Algorithms

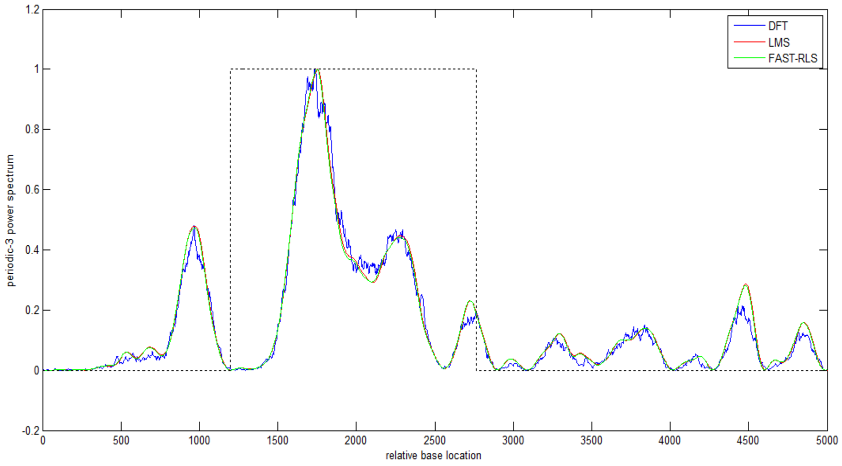

Figure 6. Power Spectra of the Gene AJ223321.1 Obtained By DFT, LMS and FAST-RLS Algorithms.

Table 2. Comparison of the Proposed Algorithm and the Other Methods in Determining Protein Coding Regions in Three Gene Sequence

Sequence	NCBI DATA	DFT	LMS	FAST-RLS
AF019074	3187-3500(313)	3093-3618(525)	3101-3599(490)	3102-3580(478)
	3761-4574(813)	3630-4653(1023)	3636-4650(1014)	3636-4649(1013)
	5832-6007(175)	5752-6545(793)	3745-6540(2795)	3745-6538(2793)
AF009962	3934-4581(647)	3735-5210(1475)	3740-5200(1460)	3742-5198(1456)
AJ223321.1	1196-2764(1568)	1195-2960(1765)	1195-2955(1760)	1196-2953(1957)

5. Conclusion

A novel adaptive filtering approach is presented to predict the biological function segments. Finally, we illustrate this method by application to identify the protein coding regions from a real DNA sequences. The predictive location curve of the exons is obtained by simulation experiments. It is shown that the presented adaptive filtering approach is valid. We construct a unified framework of adaptive filtering for genomic and proteomic signal processing. The main advantage of this method is that it can be easily utilized to identify some biological function segments very fast by adjusting filter weights.

References

[1] S. Haykin, “Adaptive filter theory”, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 4th ed., **(2002)**.

[2] D. Anastassiou, “Genomic signal processing”, IEEE Sign. Proc. Mag. vol. 18, **(2001)**, pp. 8-20.

[3] N.Chakravarthy, “Autoregressive modelingand feature analysis of DNA sequence”, EURASIP J. Appl. Sign. Proc., **(2004)**, pp. 13-28.

[4] B. D. Silverman and R. Linsker, “A measure of DNA periodicity”, J. Theor. Biol., vol. 118, **(1986),** pp. 295-300.

[5] R. Zhang and C.T. Zhang, “Z curves, an intuitive tool for visualizing and analyzing the DNA sequences”, J.Biomol. Struct. Dyn., vol. 11, **(1994)**, pp. 767-782.

[6] C.T. Zhang and J. Wang, “Recognition of protein coding genes in the yeast genome at better than 95% accuracy based on the Z curve”, Nucleic Acids Res. vol.28, **(2000)**, pp. 2804-2814.

[7] A. S. Nair and S. Sreenadhan, “A coding measure scheme employing electron-ion interaction pseudopotential (EIIP). Bioinformation, vol. 1, **(2006)**, pp. 197-202.

[8] K. D. Rao and M. N. S. Swamy, “Analysis of genomics and proteomics using DSP techniques. IEEE Trans”, Circuits Syst., vol. 55, **(2008)**, pp. 370-378.

[9] I. Cosic, “Macromolecular bioactivity: is it resonant interaction between macromolecules”?—Theory and applications.IEEE Trans. Biomed. Eng., vol. 41, **(1994)**, pp. 1101-1114.

[10] D. Anastassiou, “DSP in genomics: processing and frequency-domain analysis of character strings”, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. (ICASSP ’01), Salt Lake City, (IEEE, Piscataway,0-7803-7041-2001, vol. 2, **(2001)** May 7–11, pp 1053–1056.

[11] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya and R. Ramaswamy, “Prediction of probable genes by Fourier analysis of genomic sequences”, CABIOS, vol. 3, no. 3, **(1997)**, pp. 263–270.

[12] E. Jacobson and R. Lyons, “ The sliding DFT”, IEEE signal processing magazine, **(2003)**, pp. 74–80.

[13] E. Jacobson and R. Lyons, “An update to the sliding DFT”, IEEE signal processing magazine, **(2003)**, pp. 110–111.

[14] S. Datta and A. Asif, “A Fast DFT-Based Gene Prediction Algorithm for Identification of Protein Coding Regions,” Proceedings of the 30th International Conference on Acoustics, Speech, and Signal Processing, **(2005)**.

[15] I. Koval and G. Gara, “Digital MF receiver using discrete Fourier transform,” IEEE transactions on Commiunication, vol. 21, no. 12, **(1973)**, pp. 1331-1335.

[16] A.V. Oppenheim and R.W.Schafer, “Discrete Time Signal Processing,” Prentice Hall, Inc, NJ, **(1999)**.

[17] F. Q. Braun, “Nonrecursive digital filters for detecting multifrequency code dignals,” IEEE Transactions on Acoustic, Speech and Signal Processing, vol. 23, no. 3, **(1975)**, pp. 250-256.

[18] S. Haykin, “Adaptive filter theory”, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 4th ed., **(2002)**.

[19] B. Widrow, J. Glover Jr, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, E. Dong Jr and R. Goodlin, “Adaptive noise cancelling: Principles and applications,” Proceedings of the IEEE, vol. 63, no. 12, **(1975)**, pp. 1692–1716.

[20] J. Makhoul, “Linear prediction: A tutorial review,” Proceedings of the IEEE, vol. 63, no. 4, **(1975)**, pp. 561–580.

[21] A. H. “Sayed Fundamentals of Adaptive Filtering”, JohnWiley &Sons, New York, NJ, USA, **(2003)**.

[22] J.R. Treichler, C.R. Johnson nd M.G. Larimore, “Theory and Design of Adaptive Filters”, PrenticeHall, **(2001)**.

[23] A.BENALLAL and A.GILLOIRE, "A New method to stabilize fast RLS algorithms based on a firstorder model of the propagation of numerical errors" Proc. ICASSP88, New York, **(1988)**.

[24] D.T.M.Slock and T.Kailath, “Numerically stable fast transversal filters for recursive least squaresadaptive filtering” , IEEE transactions on signal processing, vol-39, no.1, **(1991)** January.

[25] A.P.Liavas and P.A.Regalia, “Numerical stability issues of the conventional recursive leastsquares algorithm”, in Proc. ICASSP, Seattle, WA, **(1998)** May.

[26] M. AREZKI, A. BENALLAL, A. GUESSOUM and D. BERKANI, “Three New Versions of Numerically Stable Fast Least Squares Algorithms (NS-FRLS) for Adaptive Filtering,” Proceedings of the 4th international symposium on Communication Systems, Networks and Digital Signal Processing 2004, University of Newcastle, U.K, **(2004)** July, pp. 528-532.

[27] M. AREZKI, F. Ykhlef, P. Meyrueis and D. Berkani, “Numerically Stable Fast Least Squares Algorithms in Adaptive Filtering” Proceedings vol. 5, of the 10th World Multi-Conference on Systemics, Cybernetics and Informatics –Orlando, Florida, USA, **(2006)** July, pp. 221-224.

[28] S. Blind, A numerically stable fast transversal filter with leakage correction,” IEEESignal Processing Letters, vol. 2, no. 6, **(1995)**, June. pp.114-116.