Detection of Coding Regions in Large DNA Sequences Using the Short Time Fourier Transform with Reduced Computational Load

Aníbal Rodríguez Fuentes[1], Juan V. Lorenzo Ginori[1], and Ricardo Grau Ábalo[2]

[1] Center for Studies on Electronics and Information Technologies
[2] Center for Studies on Informatics, Universidad Central "Marta Abreu" de Las Villas,
Carretera a Camajuaní, km 5 ½, Santa Clara, VC, CP 54830, Cuba
anibalr@uclv.edu.cu, juanl@uclv.edu.cu, rgrau@uclv.edu.cu
http://www.fie.uclv.edu.cu

Abstract. Due to the non-uniform distribution of codons in coding regions, a three-periodicity is present in most of genome coding regions which, after a previous numeric conversion, show a notable peak at frequency component N/3 when calculating the Fourier Transform. Taking into account the veracity of this result, the Short Time Fourier Transform has been applied to large DNA sequences to predict coding regions. This paper presents a new approach to reduce the computational burden associated with STFT computation, for coding regions detection purposes. Experimental results show significant savings in computation time when the proposed algorithm is employed.

1 Introduction

Bioinformatics has become one of the most exciting areas of research in science today. The whole description of the human genome is about three billion characters in length. In the last years, scientists have completed the sequencing of a few other organisms, which are very useful in the study of general features and the architecture of entire genomes. The speed at which data is currently being acquired is growing at very high rates. Interpreting the meaning of these genome sequences is a big challenge for scientists today.

The standard approach used to represent genome sequences consists in representing the genomic information by sequences of nucleotide symbols in the strands of DNA and RNA molecules, by symbolic codons (triplets of nucleotides), or by symbolic sequences of amino acids in the corresponding polypeptide chains (for the genes). This approach limits the methodology for handling the genomic information to mere pattern matching or statistical procedures. However, numerical assignments can also be made as an alternative for analysis purposes. Many approaches [1-6] have been used in order to transform a DNA sequence into a numerical signal. For example, one of the most used approaches is the computation of four binary sequences (one per each base A, T, C and G), called binary indicator sequences, where 1 at position *k* indicates the presence of the base at that position, and 0 its absence. Another approach consists in assigning numerical values to each one of the nucleotide bases, as is used in this work and explained in detail below.

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 902 – 909, 2006.

It is known that due to the non-uniform distribution of codons in coding regions, a three-periodicity is present in most of genome coding regions, which show a peak at frequency N/3 when calculating their Discrete Fourier Transform (DFT). Many authors have used this result to propose approaches to detect coding regions in large DNA sequences. In [1], the Short Time Fourier Transform (STFT) is used to detect five coding regions in an 8000 base pairs DNA stretch of *C. elegans*, and in [7] a new measure, based on the DFT phase at a frequency *N*/3, is presented. It is important to notice that when applying the STFT in these cases, usually the frequency component that corresponds to the periodicity three is the only one to be calculated.

In this paper a new approach to reduce the computational load when calculating the STFT for coding regions detection purposes is presented. It is based on a computational simplification obtained when calculating the Fourier Transform (FT) for a data window centered in a certain point, knowing the FT of a data window of the same size, but shifted one point backwards in the sequence. To complement this result, the Goertzel algorithm was used to calculate the frequency content in the first window of the entire sequence.

2 Materials and Methods

In the following paragraphs there is a presentation of the method introduced in this work to perform the computations associated to the spectral analysis of a genomic sequence. For this purpose, the nucleotide bases are previously mapped into a sequence of complex numbers according to

$$A=1+j; \quad T=1-j; \quad C=-1-j; \quad G=-1+j\,.$$

As a result a discrete sequence of complex numbers is obtained, that can be analyzed through standard techniques like the Discrete Fourier Transform [1].

2.1 Reducing the Computational Load When Calculating the Discrete Fourier Transform for Sliding Windows

The STFT uses a sliding window along the sequence and calculates the Fourier Transform of each subsequence. We consider here the case where a rectangular window is used, so that we do not have to multiply the data samples by any coefficient, as occurs when using a weighting window.

The k-th coefficient of the Discrete Fourier Transform of a signal x[*n*] of length N can be computed as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \quad . \tag{1}$$

Assume now that $X_p[k]$ is the *k*-th coefficient of the DFT for the subsequence starting at position p within the sequence, and with a window of length N:

$$X_p[k] = \sum_{n=0}^{N-1} x[p+n] e^{-j2\pi nk/N} \tag{2}$$

So $X_{p+1}[k]$ can be computed as:

$$X_{p+1}[k] = \sum_{n=0}^{N-1} x[p+n+1] e^{-j2\pi nk/N} \tag{3}$$

Making the change of variables m = n + 1 (n = m – 1) we can show that

$$X_{p+1}[k] = e^{j2\pi k/N} \sum_{m=1}^{N} x[p+m] e^{-j2\pi mk/N} \tag{4}$$

Now it is possible to compute $X_{p+1}[k]$ using the previous result (2) for $X_p[k]$:

$$X_{p+1}[k] = e^{j2\pi k/N}(X_p[k] - x[p] + x[p+N])$$

In the particular case of interest in this study, for the periodicity three when k = N/3 we have:

$$X_{p+1}[N/3] = e^{j2\pi/3}(X_p[N/3] - x[p] + x[p+N]) \tag{5}$$

Using the previous equation, we can just calculate the DFT for the first subsequence (window) instead of calculating a Fourier Transform for each window, and then use its N/3 frequency component to calculate the same component in the next subsequence. This procedure can be continued for all the window displacements until the whole DNA sequence is analyzed. Notice that with this approach, only one DFT, corresponding to the first subsequence, is to be fully calculated. This is the basis for the significant reduction in the computational cost that was obtained. The same savings in computational load can be obtained for every frequency component of the STFT, which makes the proposed method very suitable to detect any periodicity that could appear in the DNA sequence [8].

2.2 The Goertzel Algorithm

To calculate the DFT frequency component at N/3 for the first subsequence, we can use the Goertzel algorithm [9] instead of using the DFT in order to increase the computational efficiency. The Goertzel algorithm computes the DFT for specific indexes in a vector or matrix by using the periodicity of the sequence $e^{-j2\pi kn/N}$ to reduce the computational load. It computes the k-th DFT coefficient of the input signal $x[n]$ using a second-order filter. The algorithm can be implemented as:

$$v_k[n] = x_e[n] + 2\cos(2\pi k/N) v_k[n-1] - v_k[n-2] \tag{6}$$

where

$$v_k[-2] = v_k[-1] = 0$$

$$x_e[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ 0, & n < 0, n \ge N \end{cases} \tag{7}$$

and

$$X[k] = v_k[N] - W_N^k v_k[N-1] \tag{8}$$

It can be seen that this method uses recursion to compute $\cos(2\pi k/N)$ and $W_N^k = e^{-j2\pi k/N}$, which are evaluated only at $n = N$. The direct DFT does not use recursion and must compute each complex term separately.

The real cost of this algorithm is 2*N*+4 real multiplications, and 4*N* real sums, for the general case of a complex signal.

Notice that although there is a computational saving by using the Goertzel algorithm when it is desired to compute only one DFT coefficient, the main savings result from the application of equation (5). This makes useful the proposed method even for the case in which more than one frequency component is to be evaluated and DFT calculations could be more efficient than the Goertzel algorithm.

3 Results

Using the Goertzel algorithm to compute the N/3 frequency component of the first subsequence and then the algorithm developed previously to compute the same frequency component for the next subsequences, we reduced considerably the computation complexity of the application of the STFT to detect coding regions in large DNA sequences.

In Table 1 it is shown a detailed comparison between the direct method and the algorithm we propose, when a complex signal is used to describe the DNA sequence, assuming that the length of the DNA sequences is *L* and the sliding rectangular window has length *N*.

Table 1. Comparison between the direct method and the proposed algorithm when calculating the STFT for a complex signal

	Direct method		Proposed algorithm		
	Per point	Total	Goertzel	Per point	Total
Real multiplications	4*N*	4*L***N*	2*N*+4	4	4*L*+2*N*
Real sums	4*N*-2	*L**(4*N*-2*)*	4*N*	6	6*L*+4*N*-6
Computational load	Order *L***N*		Order *L*+*N*		

As Table 1 shows, the proposed algorithm can reduce significantly the computational burden for the calculation of the Short Time Fourier Transform to detect coding regions in DNA sequences. If we use binary indicator sequences to obtain the power spectrum of the signal like in [1] the reduction is more significant because we need to compute 4 times the DFT per point, although in this case the signals are real. The use of the STFT involves completing the sequence with 2N/3 ceros at the beginning and with N/3 ceros at the end, and this reduces the amount of operations when computing the N/3 frequency component in the first 2N/3 windows and last N/3 windows of the cero-padded sequence. However, this reduction is not shown in Table 1 because it is not significant for large sequences.

It is also important to compare the computation load between the Fast Fourier Transform and the proposed algorithm when it is necessary to compute all frequency components per window, which is the case when calculating complete spectrograms. The FFT computation load is $N*\log_2 N$ for a window, which is totalized as $L*N*\log_2 N$, for L windows, while the computation load for the proposed algorithm when computing all frequency components is $L*N*(1+(\log_2 N)/L))$. Here an FFT is used instead the Goertzel algorithm to compute all frequency components of the first window. Comparing $\log_2 N$ and $1+(\log_2 N)/L$ it is possible to realize that a noticeable saving in computational load is obtained when using the proposed algorithm.

In Table 2 it is shown the average execution time, in seconds, of the expression: $|aA + tT + cC + gG|^2$, where A, T, C and G are respectively the DFT values at N/3 for each subsequence of length N corresponding to binary indicator sequences Xa, Xt, Xc and Xg, and a, t, c and g are complex constants obtained in [1] as the solution of an optimization problem to maximize the discriminatory capability between protein coding regions and random DNA regions. The values of these complex constants are:

$$a = 0.10 + 0.12\mathrm{j} \qquad t = -0.30 - 0.20\mathrm{j}$$
$$c = 0 \qquad g = 0.45 - 0.19\mathrm{j} \tag{9}$$

In the computer experiments, different DNA strings contained in Chromosome III of *C. elegans* were analyzed, and both approaches, the direct method and the algorithm we propose, were used for this purpose. The 8000 base pairs DNA stretch was the same used by Anastassiou in [1] for a sliding window of length 351.

Table 2. Computation time comparison, in seconds, between the direct method and the algorithm we propose for different DNA strings contained in Chromosome III of *C. elegans*

DNA stretch	8000 bp DNA stretch		15100 bp DNA stretch		42799 bp DNA stretch	
Window Length	351	702	351	702	351	702
Direct Method (DM)	4.9370	8.6130	8.8630	15.6730	26.1270	46.4470
Proposed Algorithm	0.0550	0.0551	0.0800	0.0800	0.2300	0.2310
% of DM time	1.11%	0.64%	0.90%	0.51%	0.88%	0.50%

Practical results confirm the computational orders shown in Table 1. Notice that when we increase the length of the window, the execution time using direct method is increased at about the same rate, whereas the time spent by the proposed algorithm is almost invariable. In general sense, our algorithm employs less that 1% of the time required when the direct method is used.

The graph shown in Figure 1, obtained using our algorithm, coincides with the one obtained by Anastassiou in [1]. Figure 2 shows the graph for the first 42799 bp of the same Chromosome. Most of the notable peaks correspond to coding regions.

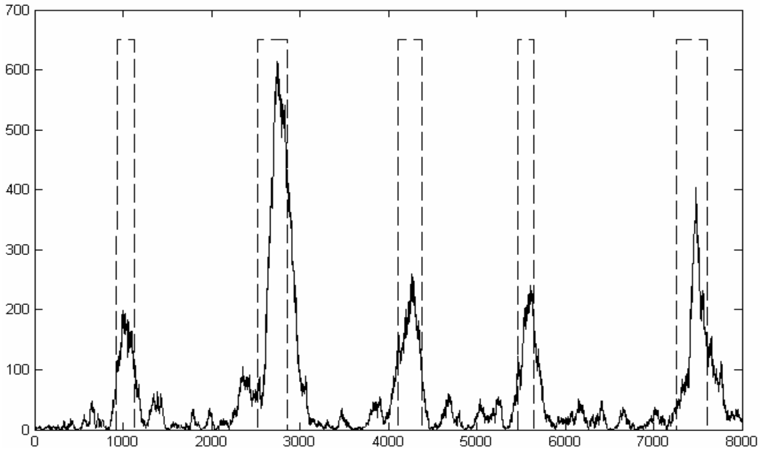

Fig. 1. Plot of $|aA + tT + cC + gG|^2$ for the 8000 bp stretch starting at position 7020 inside Chromosome III of *C. elegans.* using a sliding window of length 351. Noticeable peaks correspond to coding regions, which are represented using dashed lines.

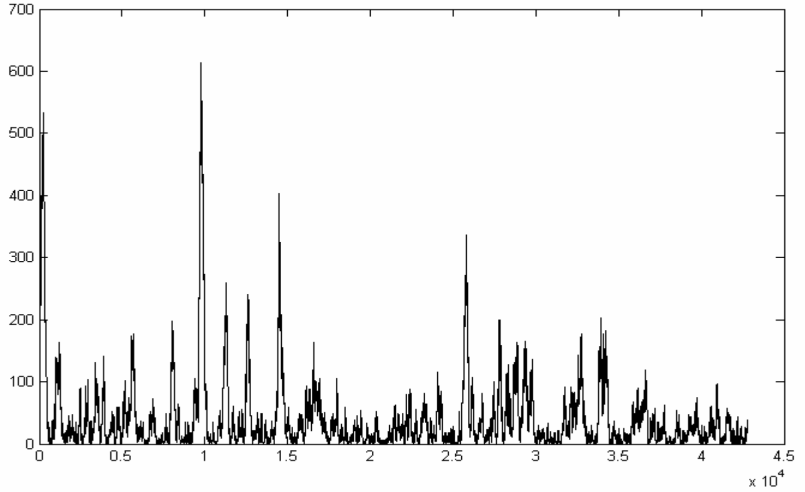

Fig. 2. Plot of $|aA + tT + cC + gG|^2$ for the first 42799 bp inside Chromosome III of *C. elegans*

4 Discussion and Conclusions

Various Digital Signal Processing based methods are being used currently to detect coding regions in large DNA strings. Among them, there are several frequency-domain techniques that make extensive use of the Discrete Fourier Transform. Some of these techniques use the amplitudes of the DFT coefficients and others their phase angles. When using these methods, a typical situation arises in which it is necessary to calculate only the frequency component at *N/*3 in a repetitive process, which involves the calculation of the DFT for many subsequences obtained from a sliding window applied to the whole sequence. This could mean a high computational load when dealing with large DNA databases.

In this work, an algorithm was introduced to reduce the computational burden associated to the calculation of the *N/*3 frequency component of the STFT for a sliding window with a one-sample step. The proposed algorithm was derived from the properties of the DFT, and combined with the well-known Goertzel algorithm, which allowed further improvements when calculating only one DFT coefficient for a particular frequency, that is precisely the situation in this application.

The computational experiments performed consisted in the calculation of the STFT for long sequences using both the proposed algorithm and the conventional method, and using the computation time as the basis to evaluate the computational efficiency. The results showed that the application of the algorithm introduced in this work, reduced at great extent (typically less than 1% for the computed cases) the computational load associated to this task. It is worth to mention that the precision of the results is not affected significantly, given that this is only influenced by the different way in which roundoff errors propagate. These errors are usually negligible when using floating point as is usual in modern computers.

A very significant reduction in computation time can be also expected when calculating more frequency components, or even the complete STFT (with all the frequency components). The results obtained suggest that the proposed algorithm can be used efficiently in analyzing long DNA strings, even in large studies involving many sequences. Practical comparison results demonstrated the good performance of the proposed algorithm.

Acknowledgements

This work was supported partially by the Canadian International Development Agency Project Tier II-394-TT02-00 and by the Flemish VLIR-UOS Programme for Institutional University Co-operation (IUC). The authors wish to thank Dr. Julián L. Cárdenas-Barrera for the revision of the manuscript.

References

1. D. Anastassiou, "Genomic signal processing," *IEEE Signal Processing Magazine*, vol. 18, pp. 8-20, 2001.
2. P. D. Cristea, "Conversion of nucleotides sequences into genomic signals," *J. Cell. Mol. Med.*, vol. 6, pp. 279-303, 2002.
3. J. A. Berger, S. K. Mitra, and J. Astola, "Power spectrum analysis for DNA sequences," *University of California*, 2002.
4. G. Dodin, P. Vanderghenynst, P. Levoir, C. Cordier, and L. Marcourt, "Fourier and Wavelet Transform Analysis, a Tool for Visualizing Regular Patterns in DNA Sequences," *J. Theor. Biol*, vol. 206, pp. 323-326, 2000.
5. J. A. Berger, S. K. Mitra, M. Carli, and A. Neri, "New approaches to genome sequence analysis based on digital signal processing," *University of California*, 2002.

6. S.-C. Su, C. H. Yeh, and C. J. Kuo, "Structural Analysis of Genomic Sequences with Matched Filtering," *IEEE Signal Proccessing Magazine*, vol. 3, pp. 2893-2896, 2003.
7. D. Kotlar and Y. Lavner, "Gene Prediction by Spectral Rotation Measure: A New Method for Identifying Protein-Coding Regions," *Genome Research*, vol. 13, pp. 1930-1937, 2003.
8. J. A. Berger, S. K. Mitra, and J. Astola, "Power spectrum analysis for DNA sequences," *Proceedings of the International Symposium on Signal Processing and its Applications (ISSPA 2003), Paris, France*, pp. 29-32, 2003.
9. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*: Prentice-Hall, 1989.