DFT based DNA Splicing Algorithms for Prediction of Protein Coding Regions

Suprakash Datta, Amir Asif
Department of Computer Science and Engineering
York University, Toronto, Canada
{datta, asif}@cs.yorku.ca

Abstract

Identifying protein coding regions in DNA sequences is a fundamental step in computational recognition of genes. Traditional Discrete Fourier transform (DFT) based approaches [9, 2, 3, 7] exploit the empirical observation that the spectrum of protein coding DNA of length N nucleotides has a peak at frequency $k = N/3$ corresponding to the length of a DNA codon. In this paper, we prove the aforementioned and several other empirical observations attributed to DNA sequences. Our analytical results lead to faster and more accurate DFT-based algorithms for predicting coding regions. Further, our algorithm requires no prior training.

1. Introduction

An essential step in annotating genomes is differentiating protein coding regions from non-coding regions; this is often referred to as *splicing*. DFT-based splicing algorithms rely on empirically observed differences between coding and non-coding regions, especially that the magnitude spectrum of a protein coding region of length N , exhibits a relatively large value at discrete frequency $k = N/3$ but near-zero coefficients at $N/6, N/9$ etc. No such behavior is observed in non-coding regions. This observation is commonly referred to as the “3-periodicity” property.

In this paper, we investigate the use of signal processing techniques for developing *de novo* or *ab initio* techniques for splicing – i.e. those which require no prior training or learning. Specifically, we provide analytical reasons for 3-periodicity and other observed differences in the Fourier spectra of coding and non-coding regions. Also, we improve the DFT-based splicing algorithm in [2, 3] in many ways. First, we compute the magnitude of the frequency component at $k = N/3$ in $O(N)$ time, without using any complex number arithmetic, and avoid the use of the fast Fourier transform (FFT) which takes $\Omega(N \log N)$ time. Second, instead of using the magnitude of the spectral coefficients, we

use the signal-to-noise ratio (SNR) of the DNA sequence which provides a more robust prediction of coding regions. Third, we use a two-character alphabet to represent DNA sequences, which makes our algorithm faster and deterministic. Finally, we show that using the Bartlett window instead of a rectangular window in the DFT computations gives better results.

Related Work: There is a considerable body of work that propose algorithms for splicing. We refer the interested reader to [4, 6, 8], which survey and compare existing algorithms. In this section, we list only papers that employ Genomic Signal Processing techniques, viz., [2, 3, 9, 7, 10]. In [9], the authors propose a DFT-based splicing algorithm that uses the squared sum of the magnitude of the frequency components at $k = N/3$ of the binary indicator sequences corresponding to the four nucleotides. References [2, 3] use prior knowledge of at least a few coding regions and involve solving an optimization problem. Further, due to its randomized nature, the algorithm may predict different boundaries and even different numbers of coding regions for the same DNA sequence in different runs. It is not clear how the outputs from different runs can be combined to yield the optimal result. In [7], the authors make novel use of the phase of Fourier coefficients to improve the accuracy of prediction of coding regions. Vaidyanathan et al [10] proposed using digital filters to clean and enhance the frequency components at $k = N/3$ of the binary indicator sequences corresponding to the four nucleotides.

The rest of this paper is organized as follows. Section 2 derives several properties of DFT coefficients of a binary sequence including an alternate expression for computing the DFT coefficient at discrete frequency $k = N/3$. In section 2, we exploit the results derived in section 2 to prove the 3-periodicity property in the protein coding region of a DNA sequence. Section 4 introduces the proposed splicing algorithms and experimental results are presented in section 5. Finally, we outline future work and open problems in section 6.

2. Properties of Binary Indicator Sequences

In this section, we derive several properties of Fourier coefficients of a binary sequence $X[n]$, $0 \leq n < N$, $X[n] \in [0, 1]$. Later, we will use these properties to explain some interesting features observed in DNA sequences.

We parse a binary sequence $X[n]$ into *words* (subsequences) of length w , ($3 \leq w \leq N$). The *position count function* counts the number of 1's present at location s in the parsed words of length w and is given by $C_w^X(s) = \sum_{n=0}^{\lfloor \frac{N-1}{w} \rfloor} X[wn + s]$, for $0 \leq s < w$. The discrete Fourier transform (DFT) of the sequence $X[n]$ is given by $\tilde{X}[k] = \sum_{n=0}^{N-1} X[n]e^{-\frac{j2\pi kn}{N}}$. It can be shown that

$$\begin{aligned} \tilde{X}[N/w] &= \sum_{m=0}^{\lfloor \frac{N-1}{w} \rfloor} \left(X[wm] + X[wm+1]e^{-\frac{j2\pi}{w}} + \right. \\ &\quad \left. \dots + X[wm+w-1]e^{-\frac{j2\pi(w-1)}{w}} \right) \quad (1) \\ &= C_w^X(0) + \dots + e^{-\frac{j2(w-1)\pi}{w}} C_w^X(w-1). \quad (2) \end{aligned}$$

Equation (2) provides an $O(N)$ -time algorithm to compute $\tilde{X}[N/3]$, instead of $\Omega(N \log N)$ -time as with the fast Fourier transform (FFT). Due to space limitations, we omit all proofs in this paper. Proofs of all our results can be found in [1]. The following properties hold.

Property 1 *If the position count functions for w -bit parsed words are equal, i.e., $C_w^X(0) = C_w^X(1) = C_w^X(2) = \dots = C_w^X(w-1)$, then the frequency component $\tilde{X}[N/w]$ of the DFT of a binary sequence is zero.*

We restate equation (1) for 3-bit words, i.e., for $w = 3$. $\tilde{X}[N/3] \triangleq \sum_{m=0}^{\lfloor \frac{N-1}{3} \rfloor} f(X[3m], X[3m+1], X[3m+2])$, where $f(a, b, c) = a + be^{-\frac{j2\pi}{3}} + ce^{-\frac{j4\pi}{3}}$.

Property 2 *The squared-magnitude of the frequency component $N/3$, $|\tilde{X}[N/3]|^2$ of a binary sequence $X[n]$ is given by $\frac{1}{2} [(C_3^X(0) - C_3^X(1))^2 + (C_3^X(1) - C_3^X(2))^2 + (C_3^X(2) - C_3^X(0))^2]$.*

Property 2 provides a fast algorithm for computing $\tilde{X}[N/3]$ without using complex arithmetic as required in most DFT implementations.

Property 3 *If the position count functions for 3-bit parsed words are equal, i.e., $C_3^X(0) = C_3^X(1) = C_3^X(2)$, then the frequency component $\tilde{X}[N/3]$ is zero. If the $C_3^X(i), i = 0, 1, 2$ are not all equal, then $\tilde{X}[N/3] \neq 0$.*

Property 4 *Any permutation of the 3-bit words in a binary sequence produces a Fourier spectra whose frequency component $\tilde{X}[k]$ at $k = N/3$ is identical in magnitude to the value obtained from the original sequence at frequency $k = N/3$ without any permutation.*

Define N_j to be the number of occurrences of string S_j in the binary sequence. We can show that $\tilde{X}[N/3] = \sum_{j=1}^6 N_j f(S_j[0], S_j[1], S_j[2])$.

Property 5 (a) *If the number of occurrences of 3-bit strings S_j , ($1 \leq j \leq 6$), obtained by parsing a binary sequence are equal, i.e., $N_1 = N_2 = \dots = N_6 = k$, then $\tilde{X}[N/3] = 0$.* (b) *If the number of occurrences $N_1, N_2, \dots, N_6$ in a 3-bit string S_j obtained by parsing a binary sequence are not equal to each other, then $\tilde{X}[N/3] > 0$.*

This property can be generalized as follows.

Property 6 *If the position count functions satisfy the 3-periodicity condition, $C_{3m}^X(s) = C_{3m}^X(s+3) = \dots = C_{3m}^X(s+3k)$, for ($0 \leq s \leq 2$) and for any $k > 0$ at which the position count functions exist, then $\tilde{X}[\frac{N}{3m}] = 0$.*

Property 7 *The average value $|\tilde{X}_{av}^{(1)}|^2$ of the squared magnitude, $|\tilde{X}[k]|^2$, ($1 \leq k \leq N-1$) of the DFT of a binary sequence $X[n]$, ($0 \leq n \leq N-1$), is computed from the position count functions using the following*

$$|\tilde{X}_{av}^{(1)}|^2 = \frac{1}{(N-1)} \left(N - \sum_{s=0}^{w-1} C_w^X(s) \right) \sum_{s=0}^{w-1} C_w^X(s).$$

3. Properties of DNA sequences

A DNA sequence $D[n]$ of length N is represented in terms of four binary indicator sequences $A[n], T[n], C[n]$, and $G[n]$, respectively for each nucleotide A, T, C, and G [2]. The indicator sequence for nucleotide A is a vector $A[n]$, ($1 \leq n \leq N$), such that $A[n] = 1$ if and only if $D[n] = 'A'$.

In section 3.1, we list a set of properties of coding regions observed in our experiments and/or reported in the literature. In sections 3.2 and 3.3, we use these properties and results from section 2 to derive several results for the coding and non-coding regions.

3.1. Properties Observed in Coding Regions

We include only results for *C. Elegans* chromosome III, (GenBank Acc. no. NC_003281, complete sequence, total length of 13783268 nucleotides, cumulative length of coding regions of about 4 million nucleotides).

Observation 1 *When the length w of the subsequence is a multiple of 3, i.e., $w = 3m$, a 3-sample periodicity is observed in the nucleotide count functions in the coding regions. For nucleotide A, we observe that*

$$C_w^A(s) \approx C_w^A(s+3) \approx \dots \approx C_w^A(s+3k) \quad (3)$$

for $k = 0, 1, 2, \dots$ such that $(s+3k) < w$. However, $C_w^A(s), C_w^A(s+1), C_w^A(s+2)$ are not equal, for $1 \leq s \leq w-3$. The same holds true for the count functions $C_w^T(s), C_w^G(s)$, and $C_w^C(s)$.

Position	A	C	G	T
0	0.3189	0.3291	0.4545	0.2488
1	0.3642	0.3664	0.2382	0.3523
2	0.3168	0.3044	0.3073	0.3989

Table 1. Fraction of nucleotides (A, C, G, and T) at locations 0, 1, and 2 in DNA words of length ($w = 3$) for coding regions.

Position	A	C	G	T
0	0.1603	0.1647	0.2271	0.1241
3	0.1587	0.1645	0.2274	0.1247
1	0.1825	0.1824	0.1191	0.1771
4	0.1818	0.1840	0.1192	0.1751
2	0.1583	0.1517	0.1551	0.1994
5	0.1585	0.1527	0.1522	0.1995

Table 2. Same as Table 1 but for $w = 6$.

Tables 1 and 2 record each of the four nucleotide counts as fractions of the total number of nucleotides of that particular type. The row entries in each column of table 1 are significantly different from each other. To show the similarity between positions 0 and 3, 1 and 4, and 2 and 5, we have rearranged the order of the six rows in Table 2. Table 2 clearly illustrates 3-periodicity for the four nucleotides.

Position	A	C	G	T
0	0.2515	0.2514	0.2491	0.2489
1	0.2495	0.2500	0.2488	0.2520
2	0.2489	0.2491	0.2523	0.2497
3	0.2501	0.2496	0.2498	0.2493

Table 3. Same as Table 1 except for $w = 4$.

Observation 2 When the length w of the subsequence is not a multiple of 3, i.e., $w = 3m + r$, m and r being integers with $r = 1, 2$, the nucleotide count functions in coding regions satisfy $C_w^A(0) \approx C_w^A(1) \approx \dots \approx C_w^A(w-1)$.

Table 3 shows the position count functions as a fraction of the total number of nucleotides of the particular type under consideration, for $w = 4$. We see that the nucleotide counts in each position are very close to each other.

3.2. Reason for a peak at $\tilde{X}[N/3]$ in coding regions

In [2, 3, 5, 9], it has been observed that the magnitude spectrum $|\tilde{S}[k]|^2 = |\tilde{A}[k]|^2 + |\tilde{T}[k]|^2 + |\tilde{C}[k]|^2 + |\tilde{G}[k]|^2$

of the protein coding region exhibits a peak at frequency $k = N/3$. This is not caused by a repeating pattern of length 3; there is a much deeper structure. It can be shown that $|\tilde{A}[N/3]|^2 = \frac{1}{2} [(C_3^A(0) - C_3^A(1))^2 + (C_3^A(1) - C_3^A(2))^2 + (C_3^A(2) - C_3^A(0))^2]$. Similar results hold for $\tilde{C}, \tilde{G}, \tilde{T}$ also. This yields the following results.

Result 1: Effect of codon permutations: As reported in [9], the magnitudes of $\tilde{A}[N/3]$, $\tilde{T}[N/3]$, $\tilde{C}[N/3]$, and $\tilde{G}[N/3]$ are not affected by codon permutations.

Result 2: Positional frequencies of nucleotides in codons: Since $C_3^X(0)$, $C_3^X(1)$, and $C_3^X(2)$ are not equal to each other in coding regions for $X = A, C, T, G$, from Property 3 we know that $|\tilde{A}[N/3]| > 0$ and so $|\tilde{S}[N/3]|^2 > 0$.

Result 3: Reason for $\tilde{X}[N/3m] \approx 0$ when $m > 1$: From Observation 1, we note that for $w = 3m$, $m > 1$, $k > 0$, and $s = 0, 1, \dots$ such that $s + 3k \leq w - 1$, the position count functions $C_w^A(s) \approx C_w^A(s+3) \approx \dots \approx C_w^A(s+3k)$. Coupling Observation 1 with Property 6, we conclude that $\tilde{A}[\frac{N}{3m}] \approx 0$ within a coding region.

3.3. Reason for $\tilde{X}[N/3] \approx 0$ in noncoding regions

Observation 3 For any word size w , the nucleotide count functions in a non-coding region of a DNA sequence satisfy $C_w^A(0) \approx C_w^A(1) \approx \dots \approx C_w^A(w-1)$ for $3 \leq w \leq N$.

We constructed tables similar to Table 1 for non-coding regions for different values of w (see [1]). They show that the nucleotide counts are fairly close to each other, and confirm Observation 3. From property 3, it follows that $\tilde{A}[\frac{N}{3}] \approx 0$. The same reasoning can be extended to prove that $\tilde{T}[\frac{N}{3}]$, $\tilde{C}[\frac{N}{3}]$, $\tilde{G}[\frac{N}{3}] \approx 0$, which implies that $\tilde{S}[\frac{N}{3}] \approx 0$ within non-coding regions.

4. Splicing Algorithm

We exploit the 3-periodicity property in this section to develop STFT-based algorithms for the prediction of protein coding regions. The algorithm uses two binary indicator sequences, $R[n]$ and $W[n]$. Sequence $R[n]$ is 1 if the nucleotide at location n in the DNA sequence is either G or C. Similarly, sequence $W[n]$ is 1 if the nucleotide at location n is either A or T.

Initialization: Set the number of window $\ell_1 = 0$.

Step 1: Apply a rectangular window of length $N_1 = 351$ to select the first N_1 nucleotides of the DNA sequence.

Step 2: For the ℓ_1^{th} DNA subsequence obtained from step 1, form the sequences $R[n]$ and $W[n]$.

Step 3: Compute $C_3^R(s)$, ($0 \leq s \leq 2$), within the selected window. The magnitude of $\tilde{R}[N_1/3]$ is computed directly from the position count functions using Property 2.

Step 4: The squared magnitude $|\tilde{R}[N_1/3]|^2$ contains peaks corresponding to coding regions and relatively low values

in non-coding regions. However, the values of these peaks vary significantly even for different DNA sequences derived from the same organisms. We use an alternative function to solve this problem: $\text{SNR}[\ell_1] = \frac{|\tilde{R}[N_1/3]|^2}{2|\tilde{R}_{\text{av}}^{(1)}|^2}$.

Step 5: The rectangular window is moved forward by 3 nucleotides, the value of ℓ_1 is incremented by 1, and starting from step 2, the procedure is repeated till the entire DNA sequence is scanned.

Step 6: Plot $\text{SNR}[\ell_1]$ as a function of ℓ_1 , and predict a coding region wherever $\text{SNR}[\ell_1] \geq \eta = 2$.

Alternate Windows: The aforementioned algorithm uses rectangular windows to partition the DNA sequence into subsequences. Rectangular windows cause the power of the STFT to *leak* over into adjacent frequencies. This leakage can be minimized by using a differently shaped window function. While several windowing functions (e.g Hamming, Hanning, or von-Hann) are possible, we use the Bartlett window, which is given by

$$w[n] = \begin{cases} \frac{2n}{N-1} & 0 \leq n \leq \frac{1}{2}(N-1) \\ 2 - \frac{2n}{N-1} & \frac{1}{2}(N-1) \leq n \leq (N-1) \end{cases} \quad (4)$$

The alternate algorithm with the Bartlett window involves the same steps as the original algorithm described in the section 4 except that the signal $R[n]$ is multiplied with the Bartlett window and the resulting signal $R_b[n] = w[n]R[n]$ is processed in step 3. Although the modified signal $R_b[n]$ is no longer binary, it still has integral values if we scale the signal by a factor $(N-1)$, and hence magnitude of $\tilde{R}[N_1/3]$ is computed directly from the position count functions using Property 2.

5. Experiments

First, we show that the use of the Bartlett window improves the performance of the DFT based algorithm and removes any extraneous peaks caused by abrupt truncations of the rectangular window. Figure 1 illustrates the differences between the magnitude of $|\tilde{R}[N_1/3]|^2$ for the DFT based splicing algorithm using the rectangular and Bartlett windows. The spectrum are more smooth and the extraneous peaks observed with the rectangular window are absent in the spectrum obtained using the Bartlett window. Next, we determine the value of threshold η used in our algorithm. We plot the cumulative distribution of SNR for coding regions and the complement of the cumulative distribution of SNR for non-coding regions for three organisms: Chromosome III of *C. Elegans* (Acc. no. NC_003281), complete genome of *E. Coli* (Acc. no. NC_002695) and complete genome of *Pirellula* sp. (Acc. no. NC_005027). The solid curve in figure 3 shows the fraction of coding regions with SNR less than the abscissa while the dotted curve shows the fraction of non-coding regions with SNR greater than

Figure 1. Comparison of our splicing algorithms with the rectangular windows ((a),(c) and (e)) and Bartlett windows ((b),(d) and (f)). The vertical lines indicate the boundaries of the coding regions

the abscissa. Based on figure 2, η should be chosen in the range $[1.75, 2.0]$. Lower values of η provide us with a better detection rate for coding regions but a higher false alarm rate for noncoding regions and vice versa. Figure 2 also provides an upper bound of around 15% on the capability of the DFT based algorithm for detection of both coding and non-coding regions.

Finally, we quantify the performance of our algorithm with the Bartlett window. We ran our algorithm on chromosome III of *C. Elegans* (Acc. no. NC_003281) comprising 13783268 nucleotides in a total of 8172 coding regions with lengths between 2 and 7204 nucleotides. For this paper, we regard a coding region as being successfully detected if our algorithm identifies any part of it as a coding region. Table 4 lists the number of coding regions successfully detected, arranged in order of increasing length of coding regions. The performance of the algorithm is better for coding regions with lengths (> 250) comparable to the window size with

Figure 2. Cumulative distribution of SNR for coding regions (solid curve) and complement of the cumulative distribution for non-coding regions dotted curve) obtained from the C. Elegans, E. Coli, and Pirellula sp. datasets.

the detection rate higher than 80%. With larger coding regions, the detection rate improves even further. However, when the length of coding regions is much smaller (< 150) than the window size N , the DFT based splicing algorithm does not perform as well. In such cases, the data extracted by the window contains both coding and non-coding nucleotides. The “3-periodicity” condition is no longer valid and the DFT based algorithms are relatively inaccurate. In terms of the detection of non-coding regions, 86% are correctly identified by our algorithm.

Exons with length	Total Number	Total Detected
$L \geq 100$	7157	3004 (42%)
$L \geq 150$	4177	2513 (60%)
$L \geq 200$	2949	2080 (71%)
$L \geq 250$	2099	1648 (79%)
$L \geq 300$	1534	1270 (83%)
$L \geq 350$	1177	1010 (86%)
$L \geq 400$	919	826 (90%)

Table 4. Number of coding region successfully detected by the STFT based algorithm.

6. Future work

In this paper, we have provided theoretical reasons for several empirical observations used by existing splicing algorithms. We have also proposed fast algorithms for splic-

ing. Our experiments show that the algorithms perform better than existing STFT-based algorithms. We are currently augmenting our algorithm by adding schemes for refinement of the predicted coding regions to predict the actual boundaries between coding and non-coding regions.

One problem that seems unlikely to be solved by our approach is the detection of very small coding regions. It would be interesting to see if genomic signal processing ideas can be used to solve this problem.

Acknowledgments

The authors thank Professor P.P. Vaidyanathan of Cal-Tech for pointing out that the computation of the magnitude of $\hat{R}[N_1/3]$ can be done using the position count functions using Property 2.

References

- [1] Full version of this paper. Available at <http://www.cs.yorku.ca/~datta>.
- [2] D. Anastassiou. DSP in genomics. In *Proceedings of the IEEE International Conference ICASSP 2001*, May 2001.
- [3] D. Anastassiou. Genomic signal processing. *IEEE Signal Processing Magazine*, July 2001.
- [4] M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, July 1996.
- [5] V. Chechetkin and A. Turygin. Size-dependence of three-periodicity and long-range correlations in DNA sequences. *Physics Letters A*, 199(1-2):75–80, 1995.
- [6] J. W. Fickett. Finding genes by computer: the state of the art. *Trends in Genetics*, 12(8):316–320, 1996.
- [7] D. Kotlar and Y. Lavner. Gene prediction by spectral rotation measure: a new method for identifying protein-coding regions. *Genome research*, 13(8):1930–1937, 2003.
- [8] S. Rogic, A. K. Mackworth, and B. F. F. Ouellette. Evaluation of gene finding programs. *Genome Research*, 11:817–832, 2001.
- [9] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy. Prediction of probable genes by fourier analysis of genomic sequences. *Computer Applications in Biosciences*, 13:263–270, 1997.
- [10] P. P. Vaidyanathan and B.-J. Yoon. Digital filters for gene prediction applications. In *Proceedings of the 36th Asilomar Conference on Signals, Systems, and Computers*, November 2002.