A Fast Algorithm for Detecting Frame Shifts in DNA sequences

Hassan Masoom*, Suprakash Datta*, Amir Asif*, Lesley Cunningham[†] and Gillian Wu[†]

*Department of Computer Science and Engineering, York University, Toronto, ON, Canada

Email: {cs242003, datta, asif}@cs.yorku.ca

[†]Department of Biology, York University, Toronto, ON, Canada

Email: {lesleyc, gillwu}@yorku.ca

Abstract—Sequencing technologies used to generate long strands of DNA are susceptible to laboratory errors that may result in several DNA nucleotides being deleted from the genome. Detecting such deletions in the protein coding regions is of utmost importance. Missing even a single nucleotide may lead to frame shifts with all the following codons (and consequently the encoded amino acids) being identified incorrectly. In addition to the deletion of nucleotides during sequencing, frame shifts can occur because of a variety of other reasons including mutations. In this paper, we present a fast computational technique to identify frame shifts in protein coding regions in DNA sequences. Our technique is based on Fourier spectral characteristics of coding regions in DNA sequences. We provide two applications of our technique – detecting deletions in DNA sequences in coding regions and also detecting frame shifts in viral DNA.

I. INTRODUCTION

Over the last decade, there has been a great increase in the number of genomes that have been sequenced completely. More genomes are being sequenced each day. The sequencing technologies used to determine long DNA sequences are susceptible to laboratory errors. Thus it is not uncommon to have one or more nucleotides deleted from the genome. Detecting such deletions in the protein coding regions are of utmost importance. Indeed, the effect of missing even a single nucleotide may lead to frame shifts with all the following codons (and consequently the encoded amino acids) being identified incorrectly. It has also been observed that frame shifts happen for a variety of reasons other than the deletion of nucleotides, including mutations.

The traditional approaches to detecting frame shifts and detecting sequencing errors use laboratory techniques and detailed knowledge about the organism being studied. Such techniques are very accurate, but they are not automated and require considerable time and resources. Computational techniques, on the other hand, are fast and often require less knowledge about the organism. The output of computational techniques may be less accurate or precise but they are still very valuable aids for focusing the Biologist's attention to the right areas in a genome.

In this paper, we present a fast computational technique to identify frame shifts in protein coding regions of newly sequenced genomes. Our technique assumes a sequenced genome with the coding regions identified accurately. It uti-

lizes a well-known property called 3-periodicity observed in protein coding regions [1], [2], [3]. Broadly speaking, this property distinguishes coding regions and non-coding regions using structural features that are observed in terms of codon bias. One implication of this property is that coding regions possess very specific Fourier spectral characteristics. This phenomenon has been used for solving the problem of *ab initio* prediction of protein coding regions in DNA sequences [2], [3], [4], [5], [6]. Our algorithm uses the same spectral characteristics (more specifically the aspect reported in [6]) to detect frame shifts in a given coding region using a novel extension of those ideas. We provide two applications of our technique – detecting deletions in protein coding regions and detecting frame shifts in viral DNA.

The algorithm proposed in this paper predicts frame shifts without requiring any a priori knowledge of the organism being studied. However, knowing whether the frame shift indicates a deletion is not possible without some knowledge about the organism. Such knowledge regarding the genome being studied may come from the source of the data; e.g., the organism being studied is a strain of a flu virus. In cases where this information is not available, the genome can be compared to a database of already sequenced genomes.

Our algorithm (described in Section IV) has running time $O(N)$ where N is the size of the sequence. A very attractive feature of our technique is that our frame shift detection algorithm does not require any computationally intensive sequence alignment or machine learning algorithms.

A. Overview of Results

We used coding regions from several viral and non-viral genomes to test the performance of our algorithm in detecting deletions of nucleotides. The set of viruses that we used include: Human Coronavirus (HCV229E), HIV1, HIV2, Herpes viruses HSV1 and HSV2, Measles, Rabies, SARS, Polio, Smallpox, Sudanese Ebolavirus and Yellow Fever. The non-viral genomes we used were human chromosomes 21 and 22 and the bacteria *E. Coli*.

We found that our algorithm detects deletions for all cases we tested and estimates the location of missing nucleotides to within a few dozen base pairs.

For the frame shift detection problem we are able to detect the frame shift that takes place in the Human Coronavirus

(HCV 229E) and estimates the location of the frame shift to within a few dozen base pairs.

B. Organization of the paper

We survey related work in Section II. Section III provides the Genomic signal processing background required for describing our algorithm proposed in this paper. Our proposed algorithm is presented in Section IV. Section V presents our experimental results. Finally, Section VI concludes the paper.

II. RELATED WORK

Several algorithms have been proposed in the literature for detecting errors in DNA sequences. We do not consider laboratory techniques, and instead, survey only computational techniques for solving these problems.

Existing computational approaches take as input a query sequence and sometimes (depending on the algorithm) a protein database, or a database of annotated genomes. These techniques can be classified as homology-based methods and learning-based methods. Homology-based methods involve comparisons and/or sequence alignments to existing sequenced genomes. Learning-based methods use the genome itself, and sometimes the knowledge of another genome to estimate the parameters of a computational model, which is then used to detect frame shifts.

A. Homology-based methods

An early paper proposing a homology-based method was [7]. This paper used similarity to known protein sequences to detect frame shift. Specifically, it proposed an algorithm that compares translations corresponding to all six reading frames with every sequence in a database of protein sequences. The algorithm displays a list of possible frame shift sites to the user. In [8], the authors solved a related but slightly different problem – they analyzed the accuracy of a specific statistical technique for reconstructing a large DNA sequence from a set of smaller sequenced fragments. These results were used to predict sequencing errors. Claverie [9] used sequence comparison with sequences in the Swissprot database to detect frame shifts. A later paper [10] provided a tool for detecting frame shift in coding regions in intron-free DNA sequences. Their technique assigned open reading frames using homology (like [7]) and used this information to detect frame shifts.

B. Learning-based methods

In [11], the authors proposed an algorithm that computes the distribution of non-overlapping 3-tuples (codons) and 6-tuples (two consecutive codons) in the three frames of an ORF. The algorithm depends on the observation that the distribution of k -tuples ($k = 3$ or $k = 6$) is distinct in the three frames in a coding region. The algorithm assumes that a coding region in only one reading frame (frame 1) is given. It uses this sequence to learn the distributions of the k -tuples, and then uses these distributions to assign reading frames to the remainder of the sequence, thereby predicting probable frame shifts. The algorithm is shown to work well on mammalian (including

human) genes. Xu et al [12] described an algorithm to detect and corrects errors by discovering changes in “statistically preferred reading frames”. They predicted 76% of the insertion/deletion (indel) errors and their predictions were 9.4 bases away from the actual indels on average. Medigue et al [13] proposed combining a coding region prediction algorithm and a search algorithm for translational initiation and termination sites. The algorithm outputs probable locations; sequences of about 500 nucleotides each centered around these sites are identified and resequenced for comparison to the previous version of the same sequence. Shah et al [14] used a simple Markov chain model to predict frame shift sites. The transition probabilities of the Markov chain are learned using some training data. Schiex et al [15] also used a Markovian model for predicting coding regions and predicting frame shifts. The Markov model was optimized for bacterial GC-rich genomes. More recently, Moon et al [16] proposed an algorithm for the computational detection of frame shift. Their algorithm uses a model of frame shift signals built from a number of empirical observations about the sites at which frame shifts take place.

III. GENOMIC SIGNAL PROCESSING ALGORITHMS

There has been a lot of work in recent times in the application of signal processing techniques in Genomics. Genomic signal processing makes use of transforms used in traditional signal processing for the analysis of genomic sequences. In this paper, we focus on the use of Fourier Transforms in Genomics. Several Genomic signal processing algorithms have been proposed for gene prediction [1]-[6]. The prediction of genes is an important problem in Genomics. Although it is not directly related to the problem being addressed in this paper, we borrow a lot of terminology and techniques from gene prediction algorithms.

In this section, we will provide some background in Genomic signal processing techniques that we utilize in this paper. We will also survey some work in this area, mainly in the area of coding region prediction. We borrow from [5] some results and terminology used below.

A. Frequency Transforms and 3-periodicity

The discrete Fourier transform (DFT) of a sequence $X[n]$ is defined as

$$\tilde{X}[k] = \sum_{n=0}^{N-1} X[n] e^{-j \frac{2\pi k n}{N}}. \quad (1)$$

A DNA sequence can be transformed into four binary indicator functions, one for each nucleotide. The binary indicator function takes the value of 1 or 0 depending on the presence or absence of the corresponding nucleotide at each position. For example, the DNA sequence 5'-ATGACTAAGAGATCCGG-3' is translated into the following four binary indicator functions

$$\left. \begin{aligned} A[n] &= [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0] \\ T[n] &= [0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0] \\ C[n] &= [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0] \\ \text{and } G[n] &= [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \end{aligned} \right\} \quad (2)$$

where $A[n]$, $T[n]$, $C[n]$, and $G[n]$ represent the binary indicator functions for nucleotides A, C, G, and T respectively. The binary indicator sequence $A[n]$, for example, takes the value of 1 at index n if the nucleotide at location i is A. Otherwise, it is assigned a value of 0. Likewise, for the other three functions. Since the indicator functions $X[n]$ ($X \in \{A, C, G, T\}$) add to 1, i.e.,

$$A[n] + C[n] + G[n] + T[n] = 1 \quad \text{for } i = 0, 1, \dots, N-1, \quad (3)$$

hence, any three of this four binary indicator functions are sufficient for characterizing the DNA sequence. The DFT of each of the binary indicator sequences

$$\begin{aligned} A[n] &\xrightarrow{\text{DFT}} \tilde{A}[k], & C[n] &\xrightarrow{\text{DFT}} \tilde{C}[k], & G[n] &\xrightarrow{\text{DFT}} \tilde{G}[k], & \text{and} \\ T[n] &\xrightarrow{\text{DFT}} \tilde{T}[k], \end{aligned} \quad (4)$$

can be combined into one term [2] as follows.

$$\tilde{S}[k] = |\tilde{A}[k]|^2 + |\tilde{C}[k]|^2 + |\tilde{G}[k]|^2 + |\tilde{T}[k]|^2. \quad (5)$$

The quantity $\tilde{S}[k]$ is called the spectral content of the DNA sequence at discrete frequency k .

It has long been known [1] (at least empirically) that any coding region of length N produces a peak in $\tilde{S}[k]$ at discrete frequency $k = N/3$. No such peak is observed within the non-coding region. This property has been called 3-periodicity in the literature [1]. It was believed to be a consequence of the fact that codons are used with unequal frequencies in coding regions. This has been called codon bias in the literature. Several papers attempted to provide explanations for this phenomenon. Recently, we provided a complete explanation of the properties of DFT coefficients [5] and used these results to improve existing coding region prediction algorithms.

Kotlar et al [6] observed that for a given genome, if we take different fragments from coding regions and compute the phases of the Fourier coefficients $\tilde{A}[N/3]$, $\tilde{C}[N/3]$, $\tilde{T}[N/3]$, $\tilde{G}[N/3]$, then the distribution of the phases thus obtained is tightly clustered around a mean value. However, the same distribution computed from non-coding regions is flat (uniform).

B. DFT-based coding region prediction

We now describe existing approaches for predicting coding regions (also known as *splicing*). In general, the DFT based algorithms consist of the following steps.

- 1) Compute the binary indicator functions for the first W nucleotides of the query DNA sequence. This step is equivalent to applying a rectangular window of length W to the DNA sequence.
- 2) Calculate the DFT's of the binary indicator sequences obtained in step 1, which are used to compute the spectral content $\tilde{S}[k]$ at discrete frequency $k = W/3$.
- 3) The rectangular window is moved forward by 3 nucleotides. For the next subsequence, steps 1 and 2 are repeated till the entire sequence is parsed.
- 4) The spectral content $\tilde{S}[W/3]$ is plotted as a function of the position of the rectangular window. The peaks in the spectral plot identify the coding regions.

References [3], [4] use an alternative spectral parameter

$$\tilde{S}[k] = |a\tilde{A}[k]| + c\tilde{C}[k]^2 + g\tilde{G}[k]^2 + t\tilde{T}[k]^2. \quad (6)$$

for $k = N/3$ to generate the spectral plots. To use (6), prior knowledge of coding regions from the same organism is assumed; this is used to compute (complex-valued) weights $\{a, t, c, g\}$ by 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. Several modifications of the DFT-based splicing approaches have been suggested. Vaidyanathan et al [17] used digital filters to clean and enhance the frequency components at $k = N/3$ of the binary indicator sequences corresponding to the four nucleotides. Kotlar et al [6] exploited the phase of the DFT coefficients to improve the accuracy of prediction of the protein coding regions. Specifically, they eliminated the need for solving the optimization problem proposed in [3], [4]. However, the algorithms in both [3], [4] and [6] need some training data before they can be used. More recently, in [5] we showed that by combining two of the binary indicator sequences, and using non-rectangular window functions, we eliminate the need for any training data. The running time of our algorithm was orders of magnitude faster than existing DFT-based splicing algorithms.

C. Fast computation of the spectra of DNA sequences

In this section, we focus on the computational aspects of the DFT-based splicing approach. Following [5], we define a parameter called position count functions. We note that Kotlar et al [6] defined a similar parameter called position frequencies.

1) *Position count functions*: Given a binary sequence $X[n]$, we parse $X[n]$ into non-overlapping *words* (subsequences) of length w , ($3 \leq w \leq N$). The term

$$\mathcal{C}_w^X(s) = \sum_{n=0}^{\lfloor \frac{N}{w} \rfloor - 1} X[wn + s], \quad \text{for } (0 \leq s < w), \quad (7)$$

counts the number of 1's present at location s in the parsed words of length w and is referred to as the *position count function*. For the DNA sequence 5'-ATG ACT AAG AGA TCC GGA-3', the position count functions for $s = 0$ are given by $\mathcal{C}_3^A(0) = 4$, $\mathcal{C}_3^T(0) = 1$, $\mathcal{C}_3^C(0) = 0$, and $\mathcal{C}_3^G(0) = 1$.

2) *Computing DFT Coefficients from Position count functions*: In this section, we derive several properties of the DFT coefficients of a binary sequence $X[n]$, $0 \leq n < N$, $X[n] \in [0, 1]$. Substituting $k = N/w$ in equation (1), we get

$$\tilde{X}[N/w] = \sum_{n=0}^{N-1} X[n] e^{-j \frac{2\pi n}{w}}. \quad (8)$$

Rearranging the summation in equation (8) as

$$\tilde{X}[N/w] = \underbrace{\sum_{n=0, w, \dots} X[n] e^{-j \frac{2\pi n}{w}}}_{\text{Term 1}} + \underbrace{\sum_{n=1, w+1, \dots} X[n] e^{-j \frac{2\pi n}{w}}}_{\text{Term 2}}$$

$$+ \dots + \underbrace{\sum_{n=w-1, 2w-1, \dots} X[n] e^{-\frac{j2\pi n}{w}}}_{\text{Term } (w-1)} \quad (9) \text{ Defining a new variable } \tilde{Y} \text{ as follows:}$$

$$\begin{aligned} \tilde{Y}[N/3] &= \left(e^{\frac{j2\pi}{3}} \tilde{X}[N/3] \right) \\ &= e^{\frac{j2\pi}{3}} \mathcal{C}_3^X(0) + \mathcal{C}_3^X(1) + e^{-\frac{j2\pi}{3}} \mathcal{C}_3^X(2), \\ &= \mathcal{C}_3^X(1) + e^{-\frac{j2\pi}{3}} \mathcal{C}_3^X(2) + e^{\frac{j2\pi}{3}} \mathcal{C}_3^X(0), \end{aligned}$$

it follows that

$$\angle \tilde{Y}[N/3] = 2\pi/3 + \angle \tilde{X}[N/3].$$

$$\tilde{X}[N/w] = \sum_{m=0}^{\lfloor \frac{N-1}{w} \rfloor} \left(X[wm] + X[wm+1] e^{-\frac{j2\pi}{w}} + \dots + X[wm+w-1] e^{-\frac{j2\pi(w-1)}{w}} \right) \quad (10)$$

where we have used the Euler property, $e^{-j2\pi m} = 1$. Equation (10) can be written as

$$\begin{aligned} \tilde{X}[N/w] &= \sum_{m=0}^{\lfloor \frac{N-1}{w} \rfloor} X[wm] + e^{-\frac{j2\pi}{w}} \sum_{m=0}^{\lfloor \frac{N-1}{w} \rfloor} X[wm+1] + \dots \\ &\quad + e^{-\frac{j2(w-1)\pi}{w}} \sum_{m=0}^{\lfloor \frac{N-1}{w} \rfloor} X[wm+w-1]. \quad (11) \\ &= \mathcal{C}_w^X(0) + e^{-\frac{j2\pi}{w}} \mathcal{C}_w^X(1) + \dots \\ &\quad + e^{-\frac{j2(w-1)\pi}{w}} \mathcal{C}_w^X(w-1). \end{aligned}$$

Implication for 3-bit Words: In this paper we are interested in 3-bit words. Substituting $w = 3$ in equation (10), we get

$$\tilde{X}[N/3] = \sum_{m=0}^{\lfloor \frac{N-1}{3} \rfloor} \left(X[3m] + X[3m+1] e^{-\frac{j2\pi}{3}} + X[3m+2] e^{-\frac{j4\pi}{3}} \right)$$

or,

$$\tilde{X}[N/3] = \mathcal{C}_3^X(0) + e^{-\frac{j2\pi}{3}} \mathcal{C}_3^X(1) + e^{-\frac{j4\pi}{3}} \mathcal{C}_3^X(2), \quad (12)$$

which provides a fast algorithm for computing $\tilde{X}[N/3]$. Equation (12) provides an $O(N)$ -time algorithm to compute $\tilde{X}[N/3]$ for binary sequences of length N . In contrast, the fast Fourier transform (FFT) requires $\Omega(N \log N)$ time.

Next we derive expressions for the phase of the DFT coefficient at frequency $N/3$. From equation (12), we have

$$\begin{aligned} \tilde{X}[N/3] &= \mathcal{C}_3^X(0) + e^{-\frac{j2\pi}{3}} \mathcal{C}_3^X(1) + e^{-\frac{j4\pi}{3}} \mathcal{C}_3^X(2), \\ &= \mathcal{C}_3^X(0) + \left(-\frac{1}{2} - j\frac{\sqrt{3}}{2} \right) \mathcal{C}_3^X(1) \\ &\quad + \left(-\frac{1}{2} + j\frac{\sqrt{3}}{2} \right) \mathcal{C}_3^X(2) \\ &= \frac{1}{2} (2\mathcal{C}_3^X(0) - \mathcal{C}_3^X(1) - \mathcal{C}_3^X(2)) \\ &\quad + j\frac{\sqrt{3}}{2} (\mathcal{C}_3^X(2) - \mathcal{C}_3^X(1)) \quad (13) \end{aligned}$$

So the phase of $\tilde{X}[N/3]$ is given by

$$\angle \tilde{X}[N/3] = \tan^{-1} \left(\frac{\sqrt{3}(\mathcal{C}_3^X(2) - \mathcal{C}_3^X(1))}{2\mathcal{C}_3^X(0) - \mathcal{C}_3^X(1) - \mathcal{C}_3^X(2)} \right).$$

Notice that the coefficient $\tilde{Y}[N/3]$ is obtained by permuting (cyclic left shifting) $\mathcal{C}_3^X(0), \mathcal{C}_3^X(1), \mathcal{C}_3^X(2)$. The above equation thus implies that the deletion of a nucleotide would shift the phase of every window starting from the point of deletion onwards by $2\pi/3$. This observation generalizes to multiple deletions in the expected way – each deleted nucleotide effectively shifts the phase by $2\pi/3$. Using identical reasoning, we can show that the insertion of a nucleotide would shift the phase of every window starting from the point of deletion onwards by $-2\pi/3$. This observation also generalizes to multiple insertions in the same way – each inserted nucleotide effectively shifts the phase by $-2\pi/3$.

Our algorithm uses this observation to detect insertions and deletions in nucleotide sequences. The details of our algorithm as well as its limitations are described in Section IV.

IV. OUR ALGORITHM

In this section, we describe our algorithm for detecting frame shifts in DNA sequences. The main property utilized by our algorithm was the characteristic of the phase of the reported in [6] – this is the observation that the phase of the Fourier spectral component at $N/3$ in each window from within a protein-coding region is tightly clustered around a single value, whereas the same quantity is uniformly distributed between 0 and 2π in the non-coding regions.

Our algorithm has some similarities with coding region prediction algorithms and has the following steps.

- 1) Compute the binary indicator functions for the first W nucleotides of the query DNA sequence. This step is equivalent to applying a rectangular window of length W to the DNA sequence.
- 2) Calculate the DFT's of the binary indicator sequences obtained in step 1, which are used to compute the DFT coefficients $\tilde{A}[W/3], \tilde{C}[W/3], \tilde{T}[W/3], \tilde{G}[W/3]$.
- 3) The rectangular window is moved forward by 3 nucleotides. For the next subsequence, steps 1 and 2 are repeated till the entire sequence is parsed.
- 4) Histograms are computed from the phases of each of $\tilde{A}[W/3], \tilde{C}[W/3], \tilde{T}[W/3], \tilde{G}[W/3]$ are computed. The largest peak in these histograms is labeled as ϕ .
- 5) The phase in each window is classified into 3 classes – $C_{\phi-2\pi/3}, C_{\phi}, C_{\phi+2\pi/3}$, and labeled -1,0,1. This classification is done using the nearest-neighbor heuristic, and the class assigned to a window is henceforth referred to as its *phase label*. The phase label thus defines a function $f : [1, n_W] \rightarrow \{-1, 0, +1\}$, where n_W is the number of windows used by the algorithm.

- 6) We then define the cumulative function $F : [1, N - n + 1] \rightarrow [-N, +N]$ as

$$F(j) = \sum_{k=1}^j f(k)$$

The function $F(j)$, when plotted against j , gives a very good visualization of the changes in phase. Note that a sequence of windows with phases in the same class will yield a straight line. Any change in phase results in a change of slope.

Fig. 1. The function $F()$ on our synthetic data set ($W = 750$).

Fig. 2. The function $F()$ on our synthetic data set with nucleotide 5347 deleted. $F()$ changes slope at nucleotide 4969 ($W = 750$).

Let us consider the output of our algorithm on a synthetic sequence. Let us consider the following sequence $X = GAGGAGGAGGAG \dots$ of size a multiple of 3, i.e., X consists of many repeats of a single codon GAG. Let us

consider the phases $\angle \tilde{G}[W/3]$. Consider a window size of $W = 351$. In this synthetic example, for all windows $C_3^G(0) = 117, C_3^G(1) = 0, C_3^G(2) = 117$. So, each window produces the same value for the phase given by

$$\begin{aligned} \angle \tilde{G}[W/3] &= \tan^{-1} \left(\frac{\sqrt{3}(C_3^G(2) - C_3^G(1))}{2C_3^G(0) - C_3^G(1) - C_3^G(2)} \right) \\ &= \tan^{-1} \left(\frac{\sqrt{3}(117 - 0)}{2 \times 117 - 0 - 117} \right) \\ &= \tan^{-1}(\sqrt{3}) = \pi/3. \end{aligned}$$

Therefore each window gets the same phase label, and the cumulative function $F()$ is a straight line. This is shown in Figure 1 (the sequence length used here is 10,800).

Now consider the effect of the 5347th nucleotide being deleted. Any window starting at nucleotide 5348 or later used by the algorithm consists of 117 repetitions of the codon AGG. For these windows, $C_3^G(0) = 0, C_3^G(1) = 117$, and $C_3^G(2) = 117$. Therefore, the phase for $\tilde{G}[W/3]$ is given by the equation:

$$\angle \tilde{G}[W/3] = \tan^{-1} \left(\frac{\sqrt{3}(C_3^G(2) - C_3^G(1))}{2C_3^G(0) - C_3^G(1) - C_3^G(2)} \right) = \pi.$$

There is a change of labels due to the deletion, and so the cumulative function $F()$ changes slope at nucleotide 4969. This is shown in Figure 2.

Prediction of frame shift sites: If the slope of $F()$ changes at location c , the algorithm predicts a frame shift at $c + W/2$ where W is the window size. The absolute difference between our prediction and the actual frame shift site is henceforth called *frame shift site prediction error*. Thus, for the synthetic data described above, our algorithm predicts that the frame shift site is $4969 + 750/2 = 4969 + 375 = 5344$, which implies that the frame shift site prediction error is 3 nucleotides.

Detecting sequencing errors: The algorithm described above detects frame shifts and requires no knowledge of the genome other than the DNA sequence and the coding region boundaries. The algorithm can be used to detect deletion errors in DNA sequences if we have available the genome from an organism in the same family and with similar frame shifts. We compare the $F()$ curves of the newly sequenced genome and the organism from the same family; if the former has an extra location where slope changes take place, we know that the extra frame shift is due to deletions.

Figure 3 shows the plot of $F()$ for a coding region from HCV229E (nucleotides 12520-20568, part of the first gene) for window size $W = 750$. Figure 4 shows the plot of $F()$ for the same coding region with nucleotides 4000 and 5000 deleted. The phase shifts at nucleotides 3742 and 4588 indicate the deletion of the nucleotides within this region.

A. Limitations of our algorithm

Our algorithm has a major limitation – if the frame shifts by a multiple of three our algorithm is unable to detect this. For example, if we insert or delete three nucleotides into

Fig. 3. The function $F()$ on a coding region from HCV229E ($W = 750$).

Fig. 4. The function $F()$ on a coding region from HCV229E with nucleotides 4000 and 5000 deleted ($W = 750$). Slope changes of $F()$ are observed at 3742 and 4598.

a genome, our algorithm fails to detect these insertions or deletions. A second limitation is that it fails to detect insertions and deletions at the ends of coding regions.

B. Running time analysis

The algorithm as described requires that we compute the position count functions in each window – this takes time $O(W)$ where W is the window size. The phase of the DFT coefficient at frequency $W/3$ can be computed from the position count functions in each window very fast (in constant time). However, we can improve on this – as described in [5], the position count functions for each window can be updated quickly by considering the 6 nucleotides that are part of the previous window or the new window but not both. Therefore the algorithm for computing the phase of $\tilde{X}[W/3]$ in each

window runs in time $O(N)$ and is independent of the window size W . Computing a histogram and labeling each window with labels $(-1, 0, +1)$ also takes $O(N)$ time. Finally, computing $F()$ is done in $O(N)$ time, so that the entire algorithm runs in time linear in the size of the sequence.

V. PERFORMANCE EVALUATION

Our algorithm was implemented in MATLAB and tested on several genomes from the NCBI database. In this section, we describe our results in detecting insertions/deletions in coding regions and also in detecting frame shifts arising from other reasons in some genomes.

We simulated sequencing errors by introducing insertions and deletions in a coding region in the genome, and running our algorithm on the modified coding region. We repeated the experiment by introducing indels in different locations. These locations are chosen sufficiently far apart so that each window had at most one site in which a frame shift had taken place.

In the following paragraphs, we only present results for detecting deletions. The results for nucleotide insertions are identical and are omitted due to space constraints.

The effect of the window size: We found that the window size W is an important parameter and quantifies a tradeoff in detecting frame shifts. Larger window sizes help in filtering out local heterogeneity in coding regions and provide more accurate detection of frame shifts. In particular, small window sizes result in false positives (i.e. they result in prediction of non-existent deletions). However, small windows also provide more accurate estimates of the locations of frame shifts. In our experiments, we found that with window sizes of 750 or more, we were able to eliminate virtually all false positives.

A. Detection of deletions or insertions in coding regions

We used coding regions from several viral genomes to test the performance of our algorithm in detecting deletions of nucleotides. The set of viruses that we used include: Human Coronavirus HCV229E (accession no. NC_002645), HIV1 (accession no. AY_878062), HIV2 (accession no. NC_001722), Herpes virus HSV1 (accession no. NC_001806), HSV2 (accession no. NC_001798), Measles (accession no. NC_001498), Rabies (accession no. NC_001542), SARS (accession no. NC_004718), Polio (accession no. NC_002058), Smallpox Variola (accession no. NC_001611), Sudanese Ebola virus (accession no. NC_006432) and Yellow Fever (accession no. NC_002031). The non-viral genomes we used were Human chromosome 21 (accession no. NC_000021), Human chromosome 22 (accession no. NC_000022) and the bacteria E. Coli (accession no. NC_002695). All the experiments in this section use only the Fourier coefficient $\tilde{A}[W/3]$.

We found that our algorithm detects deletions for all cases we tested and estimates the location of the missing nucleotide to within about a hundred base pairs. The accuracy of prediction of the frame shift sites is shown in Figures 5 and 6. Note that the x-axis is not the frame shift site prediction error but the distance of the actual frame shift site from the location of the slope change of $F()$, normalized as a fraction of the

Fig. 5. Accuracy of viral frame shift site prediction ($W = 750$).

Fig. 7. Accuracy of frame shift site prediction for human DNA ($W = 1002$).

window size in each case. The figures show that the actual frame shift sites are clustered around a distance of $W/2$ of the slope change of the $F()$ curve.

Figure 5 shows the accuracy of prediction of the frame shift sites for window size $W = 750$. In this case, the average frame shift site prediction error is about 5 nucleotides. Figure 6 shows the same results for window size $W = 1002$. In this case the average frame shift site prediction error is about 42 nucleotides.

Fig. 6. Accuracy of viral frame shift site prediction ($W = 1002$).

We also tested our algorithm on coding regions from non-viral organisms. The results are as follows. Figure 7 shows the histogram of phases for human DNA (chromosomes 21 and 22). The average frame shift site prediction error in this data set was 42.18 nucleotides. Figure 8 shows the histogram of phases for E. Coli. The average frame shift site prediction error in this data set was about 3.5 nucleotides.

Fig. 8. Accuracy of frame shift site prediction for E. Coli ($W = 1002$).

B. Analysis of viral DNA

We also applied our technique for analyzing frame shifts in viral DNA sequences that are caused by reasons other than sequencing errors, e.g., mutations or overlaps in genes.

Figures 9 and 10 shows the $F()$ curves for the Human Coronavirus (HCV 229E) for two window sizes. The genome has a frame shift at nucleotide 12,520. Figure 9 shows the $F()$ curve for window size 750. In this graph, the slope change is at nucleotide 12,064 so our predicted frame shift location is $12,064 + 750/2 = 12,439$. So the frame shift site prediction error is 81 nucleotides. Note that the graph allows us to visually distinguish small local fluctuations in the phase (e.g.

Fig. 9. Visualization of the frame shift site for HCV229E, $W = 750$.

around nucleotide 3000) from the major change at 12,064.

Figure 10 shows the $F()$ curve for window size 1002. In this graph, the slope changes at nucleotide 11,902 so our estimate of the frame shift location is $11,902 + 1002/2 = 12,403$. This implies that the frame shift site prediction error is 117 nucleotides. Note that the larger window size removes the small local fluctuations observed in Figure 9, but it also increases the prediction error.

Fig. 10. Visualization of the frame shift site for HCV229E, $W = 1002$.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a fast computational technique for detecting frame shifts in coding regions in different organ-

isms. We used our technique to detect sequencing errors in coding regions and also in detecting slippage sites in viral genomes. While our algorithm detects all frame shifts we studied, its prediction of the frame shift site was on average a few nucleotides away from the actual site. We are currently working on a second phase for our algorithm that refines the frame shift site predictions of the proposed algorithm.

ACKNOWLEDGMENT

This work is supported by NSERC grants 228415 and 503900 and by an undergraduate summer research award (USRA).

REFERENCES

- [1] V. Chechetkin and A. Turygin, "Size-dependence of three-periodicity and long-range correlations in DNA sequences," *Physics Letters A*, vol. 199(1-2), pp. 75–80, 1995.
- [2] 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*, vol. 13, pp. 263–270, 1997.
- [3] D. Anastassiou, "DSP in genomics," in *Proceedings of the IEEE International Conference ICASSP 2001*, May 2001.
- [4] —, "Genomic signal processing," *IEEE Signal Processing Magazine*, July 2001.
- [5] S. Datta and A. Asif, "A fast DFT based gene prediction algorithm for identification of protein coding regions," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 5, March 2005, pp. 653–656.
- [6] D. Kotlar and Y. Lavner, "Gene prediction by spectral rotation measure: a new method for identifying protein-coding regions," *Genome research*, vol. 13, no. 8, pp. 1930–1937, 2003.
- [7] J. Posfai and R. Roberts, "Finding errors in DNA sequences," *Proc. Natl. Acad. Sci.*, pp. 4698 – 4702, 1992.
- [8] G. Churchill and M. Waterman, "The accuracy of DNA sequences: Estimating sequence quality," *Genomics*, vol. 14, pp. 89–98, 1992.
- [9] J. Claverie, "Detecting frame shifts by amino acid sequence comparison," *J. Mol. Bio.*, pp. 1140–1157, 1993.
- [10] N. Brown, C. Sander, and P. Bork, "Frame: detection of genomic sequencing errors," *Bioinformatics*, vol. 14, pp. 367–371, 1998.
- [11] G. Fichant and Y. Quentin, "A frameshift error detection algorithm for DNA sequencing projects," *Nucleic Acids Research*, pp. 2900 – 2908, 1995.
- [12] Y. Xu, R. Mural, and E. Uberbacher, "Correcting sequencing errors in DNA coding regions using a dynamic programming approach," *Comp. Appl. Sci.*, pp. 117 – 124, 1995.
- [13] C. Medigue, M. Rose, A. Viari, and A. Danchin, "Detecting and analyzing DNA sequencing errors: Toward a higher quality of the bacillus subtilis genome sequence," *Genome Research*, vol. 9, no. 11, pp. 1116–27, November 1999.
- [14] A. Shah, M. Giddings, J. Parvaz, R. Gesteland, J. Atkins, and I. I.P., "Computational identification of putative programmed translational frameshift sites," *Bioinformatics*, vol. 18, no. 8, pp. 1046–53, August 2002.
- [15] T. Schiex, J. Gouzy, A. Moisan, and Y. de Oliveira, "FrameD: a flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences," *Nucl. Acids Res.*, vol. 31, no. 13, pp. 3738–3741, 2003. [Online]. Available: <http://nar.oxfordjournals.org/cgi/content/abstract/31/13/3738>
- [16] S. Moon, Y. Byun, H.-J. Kim, S. Jeong, and K. Han, "Predicting genes expressed via -1 and +1 frameshifts," *Nucleic Acids Res.*, vol. 32, no. 16, pp. 4884–4892, 2004.
- [17] P. P. Vaidyanathan and B.-J. Yoon, "The role of signal-processing concepts in genomics and proteomics," *Journal of the Franklin Institute, Special Issue on Genomics*, 2004.