

A FOURIER PRODUCT METHOD FOR DETECTING APPROXIMATE TANDEM REPEATS IN DNA

Vincent A. Emanuele II*, Thao T. Tran**, and G. Tong Zhou

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0250, USA

ABSTRACT

Finding repetitive sequences in DNA is of particular interest in biology due to their role in genetic diseases, human gene mapping, evolution, and many other important and interesting applications. We propose a method for finding hidden periodicities in DNA based on Fourier analysis. It is shown that this approach is effective for finding new repeat patterns that were previously undetected by some of the most popular algorithms published in the literature.

1. INTRODUCTION

Repetitiveness and redundancy is inherent to the human genome. In fact, preliminary analysis of the first draft of the human genome concluded that repetitive structures are present in over one-third of the genome [1]. Because of this, it is of particular interest to study and classify repetitive patterns present in the genome and determine their role. Traditionally, there are two types of genomic repeats: transposable interspersed repetitive elements, which are composed of long and short interspersed elements (LINEs, SINEs), and tandem repeats. A tandem repeat is a contiguous stretch of DNA composed of one particular pattern that is approximately repeating itself one after the other. If the tandem repeat is composed of a pattern that is *perfectly* repeating, it is referred to as a perfect tandem repeat (PTR). Otherwise, the tandem repeat is referred to as an approximate tandem repeat (ATR). There has been a great deal of interest in the study of tandem repeats and its associated disorders [2–7]. Furthermore, thousands of short 6 base pairs (bp) tandem repeats present at the ends of chromosomes (telomeres) progressively shortens by 50-100 bp with each replication [8]. This suggests the importance of tandem repeats in the cell death cycle, which is crucial in understanding complex disorders such as cancer.

There has been considerable work done to develop algorithms to find all tandem repeats occurring in a genome. Finding ATRs has proven to be a challenging problem in part due to the fact that there is no general consensus on the definition of an ATR.

One possible approach to finding ATRs is to first find all PTRs occurring in a string and then use these as seeds to find degenerate repeats with these patterns. Mreps is a generalization of this approach [9].

Another approach to finding ATRs is to first find the location of all repetitions (not necessarily in tandem) of small words of length k in a sequence, and then develop a statistical test to see if

these repetitions are due to chance or if in fact they are small subsequences in a larger ATR. Tandem Repeats Finder [10] is based on this principle. Algorithms of this class are limited in the repeat period size they can detect.

In this paper, we propose an ATR finding algorithm based on a different approach to the problem. The philosophy used by our approach is to employ Fourier analysis to detect weak ATR signals hidden in DNA sequences. Using this method, we assume very little about the nature of the repeat we are looking for *a priori*.

2. FORMULATION OF TANDEM REPEAT PROBLEM

We are now ready to proceed to a discussion of the problem of finding ATRs in DNA.

2.1. Numerical Representation of DNA

In order to apply Fourier techniques to DNA, we first map the 4 nucleotides to 4 unique elements in a vector space. We choose the indicator representation of DNA [11]. This method is defined by a mapping $\phi : \{A, C, G, T\} \mapsto \mathbb{R}^4$,

$$\phi(x) = \begin{cases} \mathbf{e}_A = (1, 0, 0, 0)^T & \text{if } x = A \\ \mathbf{e}_C = (0, 1, 0, 0)^T & \text{if } x = C \\ \mathbf{e}_G = (0, 0, 1, 0)^T & \text{if } x = G \\ \mathbf{e}_T = (0, 0, 0, 1)^T & \text{if } x = T \end{cases} \quad (1)$$

2.2. Definition of an ATR

Let $s[n] = (\dots, s_{-k}, \dots, s_{-1}, s_0, s_1, \dots, s_k, \dots)$ be a DNA sequence, and let $\mathbf{x}[n] = \phi(s[n])$, $\forall n$ be the equivalent indicator representation. We now define the *indicator sequence* with respect to nucleic acid α as

$$x_\alpha[n] = \mathbf{e}_\alpha^T \mathbf{x}[n] \quad \forall \alpha \in D, \forall n \in \mathbb{Z}. \quad (2)$$

In this notation, we can rewrite $\mathbf{x}[n] = (x_A[n], x_C[n], x_G[n], x_T[n])^T$.

Definition 1 - Approximate Tandem Repeat in DNA

Given a DNA sequence $s[n] : \mathbb{Z} \mapsto D$ and its associated indicator representation $\mathbf{x}[n]$, we say $s[n]$ is an *approximate tandem repeat* if there exists an $\alpha \in D$ such that $x_\alpha[n]$ is periodic.

This formulation of the approximate tandem repeat problem, essentially finding a perfect periodicity in a subspace $\text{span}\{\mathbf{e}_\alpha\}$, seems to suggest a discrete Fourier transform solution. Note however, that Definition 1 is for infinite duration DNA sequences. While useful for theoretical analysis and insight, for implementation purposes we must introduce a notion of local periodicity.

*These authors have contributed equally.

** Supported by the U.S. National Science Foundation under the Graduate Fellowship Program.

Definition 2 - Local Approximate Tandem Repeat in DNA

Given a DNA sequence $s[n] : \mathbb{Z} \mapsto D$ and its associated indicator representation $\mathbf{x}[n]$, we say $s[n]$ has a *local approximate tandem repeat* with period p if there exists an $\alpha \in D$, $k \in \mathbb{Z}$, $W \in \mathbb{N}$ such that $x_\alpha[n+p] = x_\alpha[n]$, for $n = k, k+1, \dots, k+W-1-p$.

Thus, a local ATR is essentially an ATR with respect to some window of length W defined by its endpoints $\{k, k+W-1\}$. We next state the approximate tandem repeat problem precisely.

Problem Statement 1 (ATRs in DNA)

Given a DNA sequence $s[n]$, find all *local approximate tandem repeats* in $s[n]$.

The introduction of the notion of a local ATR immediately suggests the use of short-time Fourier transform methods, where a window of some predetermined length scans across the sequence looking for local periodicities. In general, a good ATR finding algorithm should have the characteristics listed in Table 1.

As an example, in the pseudo-DNA sequence:

CGC[ATGATGCTGATG]G, the local ATR with consensus pattern ATG is shown in brackets $[\cdot]$. Note that the corresponding $x_\alpha[n]$ as shown in Table 2 are aperiodic.

In the next section, we discuss and evaluate two possible Fourier-based solutions.

Table 1. Characteristics of a Good ATR Finding Algorithm

- 1) Assumes no *a priori* knowledge of periods of ATRs occurring in the sequence.
- 2) Assumes no knowledge about the pattern(s) being repeated.
- 3) Finds *all* local ATRs in a given DNA sequence.
- 4) Is computationally efficient.

Table 2. Pseudo-DNA example

$s[n]$	CGCATGATGCTGATGG
$x_A[n]$	0001001000001000
$x_C[n]$	1010000001000000
$x_G[n]$	0100010010010011
$x_T[n]$	0000100100100100

3. ANALYSIS OF TWO POSSIBLE FOURIER APPROACHES

In this section, two Fourier approaches to finding approximate tandem repeats in DNA (Problem Statement 1) are considered from a theoretical standpoint. The first method considered is based on the sum spectrum of the indicator representation of the sequence. An alternative to using the sum spectrum is proposed in this paper and is called the Fourier product spectrum (FPS). The FPS was first considered in the framework of the ATR problem in [5].

3.1. Sum Spectrum

Given a DNA sequence $s[n]$ and its indicator representation $\mathbf{x}[n]$, the discrete-time Fourier transform (DTFT) of $\mathbf{x}[n]$, denoted as

$\mathbf{X}(f) = \mathcal{F}\{\mathbf{x}[n]\}$, is defined as

$$\mathbf{X}(f) = \sum_{n=-\infty}^{+\infty} \mathbf{x}[n]e^{-j2\pi fn}. \quad (3)$$

The *sum spectrum* is then defined as

$$P_{\mathbf{x}}(f) = \|\mathbf{X}(f)\|^2. \quad (4)$$

The *indicator spectra* of $\mathbf{x}[n]$ are defined as

$$X_\alpha(f) = \sum_{n=-\infty}^{+\infty} x_\alpha[n]e^{-j2\pi fn} \quad \alpha \in D. \quad (5)$$

The sum spectrum and the indicator spectra are related to each other by the simple formula

$$P_{\mathbf{x}}(f) = \sum_{\alpha \in D} |X_\alpha(f)|^2. \quad (6)$$

Tiwari *et al.* proposed a Fourier based method for detecting coding regions in DNA based on an analysis of the sum spectrum [12]. It can be shown that the equivalent time domain representation of the sum spectrum is,

$$\mathcal{F}^{-1}\{P_{\mathbf{x}}(f)\} = \sum_{\alpha \in D} r_\alpha[n] \triangleq p_x[n], \quad (7)$$

where $r_\alpha[n] = x_\alpha[n] * x_\alpha[-n]^*$ and $*$ denotes convolution. If $x_\alpha[n]$ is periodic with period p , then $r_\alpha[n]$ is also periodic with period p . Clearly, if all of the $x_\alpha[n]$ are periodic with the same period p , then $p_x[n]$ will also be periodic with the same period p and $P_{\mathbf{x}}(f)$ should reflect this with a peak evident at $f = 1/p$ (and possibly peaks at harmonics of $1/p$). However, suppose $x_A[n]$ is periodic with period p , but $x_C[n]$, $x_G[n]$, and $x_T[n]$ are aperiodic, then $p_x[n]$ will not be periodic with period p . While in this case component $|X_A(f)|^2$ will have a peak at the appropriate frequency, observing Eq. 6, it is clear that the other three indicator spectra will essentially act as additive noise. Because of this, $P_{\mathbf{x}}(f)$ may not give a clear peak at $f = 1/p$. Now, recalling our definition of ATR, the scenario just described is an example of when the sum spectrum will have difficulty detecting an ATR. Thus, the sum spectrum method will have difficulty finding certain classes of ATRs.

3.2. Fourier Product Spectrum

We now introduce the Fourier product spectrum (FPS) [5], which can avoid the limitations of the sum spectrum. The FPS, denoted by $S_x(f)$, is defined as

$$S_x(f) = \prod_{\alpha \in D} |X_\alpha(f)|. \quad (8)$$

The time domain counter-part of $\prod_{\alpha \in D} X_\alpha(f)$ is

$$s_x[n] = x_A[n] * x_C[n] * x_G[n] * x_T[n]. \quad (9)$$

One can immediately observe that if any of the $x_\alpha[n]$ are periodic with period p , then $s_x[n]$ will also be periodic with period p . Thus, $S_x(f)$ should have peaks reflecting any periodicities in the indicator sequences $x_\alpha[n]$. Thus, this method seems to be a superior solution for detecting ATRs, except for the case when $X_\alpha(1/p) = 0$, for some α . This is an implementation issue that will be addressed in subsequent sections. See Figure 2 in [5] for an example comparing the sum spectrum and the product spectrum.

4. FPS IMPLEMENTATION: THE ITERATIVE MFPS ALGORITHM

In order to implement the FPS method to successfully find all local ATRs in a DNA sequence, we must address several important practical issues as listed below.

4.1. Long periods

To avoid interference from the DC component of the FPS of an indicator sequence, we modify the FPS by subtracting the mean of each indicator sequence $x_\alpha[n]$ from $x_\alpha[n]$ (see Eqs. 10, 11).

$$m_\alpha = \frac{1}{N} \sum_{n=0}^{N-1} x_\alpha[n] \quad \text{for } x_\alpha[n] \text{ with length } N \quad (10)$$

$$\hat{x}_\alpha[n] = x_\alpha[n] - m_\alpha, \quad \alpha \in D \quad (11)$$

We refer to $\hat{x}_\alpha[n]$ (Eq. 11) as the *modified indicator sequence*.

4.2. Nulling effects

Another issue arises when a nucleotide is absent from a given (windowed) DNA sequence. In this case, one of the indicator sequences will be zero for all n . Thus, the FPS as defined by Eq. 8 will be equal to zero. Note that this situation is quite unlikely, especially for large windows. Regardless, to avoid this we simply check for this case and modify the procedure as follows. Thus, given an indicator sequence (or a modified indicator sequence), we can define

$$c(x_\alpha) = \begin{cases} 1 & \text{if } x_\alpha[n] = 0 \quad \forall n \\ 0 & \text{otherwise} \end{cases}. \quad (12)$$

We now define the *modified indicator spectrum* as

$$\hat{X}_\alpha(f) = \left(\sum_{n=-\infty}^{+\infty} \hat{x}_\alpha[n] e^{-j2\pi f n} \right) + c(x_\alpha). \quad (13)$$

The *modified Fourier Product Spectrum (mFPS)* can then be defined as,

$$\hat{S}_x(f) = \prod_{\alpha \in D} |\hat{X}_\alpha(f)| \quad (14)$$

4.3. Peak Detection/Period Identification

We now address the problem of how to analyze the mFPS of a window to determine whether a local ATR exists in the window, and if applicable, its period. In particular, we are interested in knowing whether $\hat{S}_x(f)$ indicates the presence of a local ATR or reflects randomly occurring DNA in the window. Define

$$S_{max} = \max_f \hat{S}_x(f), \quad (15)$$

$$f_{max} = \arg \max_f \hat{S}_x(f). \quad (16)$$

We need a test that evaluates how far S_{max} lies above the rest of the values in the spectrum. If S_{max} is significantly higher relative to the rest of the spectrum, then we declare that S_{max} is a peak and there exists a potential local ATR of period $p = 1/f_{max}$. For this, we propose to test

$$l(\hat{S}_x(f)) = \frac{\max_f \hat{S}_x(f)}{\text{median}_f \hat{S}_x(f)} \geq \gamma \quad (17)$$

Such a test was used in [13]. The test $l(\hat{S}_x(f))$ should be large when an ATR is present, and small when the DNA in the window is generated at random. Our approach to choosing the right threshold γ is similar in principle to the Neyman-Pearson testing strategy. We choose a constant false-alarm rate of $P_{fa} = 0.05$, and calculate the γ necessary to operate at this significance level. Theoretical analysis of the distribution of $l(\hat{S}_x(f))$ in Eq. 17 is non-trivial. Using monte-carlo simulations for the random DNA case, we generate the empirical distribution of Eq. 17, and choose γ based on this observed distribution. We have observed that the threshold depends on the window size M used according to an empirically determined relationship $\gamma = 5.7232 \ln(M) + 8.6516$. In a similar fashion, for the $P_{fa} = 0.01$ significance level the threshold is given by $\gamma = 4.9679 \ln(M) + 26.654$.

4.4. Annotation

Once we have detected a local ATR and identified its fundamental period, we need to identify what subsequence in our window corresponds to the local ATR. Our annotation approach is to start with small perfect tandem repeats as seeds which we then grow into a more general ATR structure. An overall description of the annotation process depicted by Fig. 1 is given in Table 3.

Table 3. Annotation Procedure

1. **Find PTRs:** For the detected period p , we first find all p -periodic PTRs in the window and store them in a list.
2. **Determine Consensus:** Next, for each PTR, we identify the *consensus pattern*, which is the pattern that is being repeated. For example, for the repeat CTGCTGCTG, the consensus pattern is CTG.
3. **Search for Anchors:** Now, we look for the occurrence of the consensus pattern in other parts of the sequence within the window. An *anchor* is the occurrence of the consensus pattern elsewhere in the sequence. We use anchors with the same consensus pattern, along with the detected period p , to partition the sequence into a potential local ATR and perform Needleman-Wunsch pairwise alignments [14, 15] between two halves of the potential ATR. The Needleman-Wunsch alignment gives us a measure of the homogeneity of the potential ATR. In general, the Needleman-Wunsch pairwise alignment algorithm measures global similarity between two sequences. If the alignment score is greater than some threshold ψ , then the region of interest is annotated as a local ATR.
4. **Grow to left or right:** At this point, we attempt to extend the left/right ends of the local ATR further. To do this, we perform the Needleman-Wunsch alignment between the consensus and the p nucleotides immediately to the left/right end of the current annotation of the local ATR. If the alignment score is greater than ψ , we extend the endpoints of the ATR to include this part. We iterate this until we can no longer extend the endpoints. A final annotation is recorded, and then we proceed forward from there to construct local ATRs for all PTRs of period p found in the window.

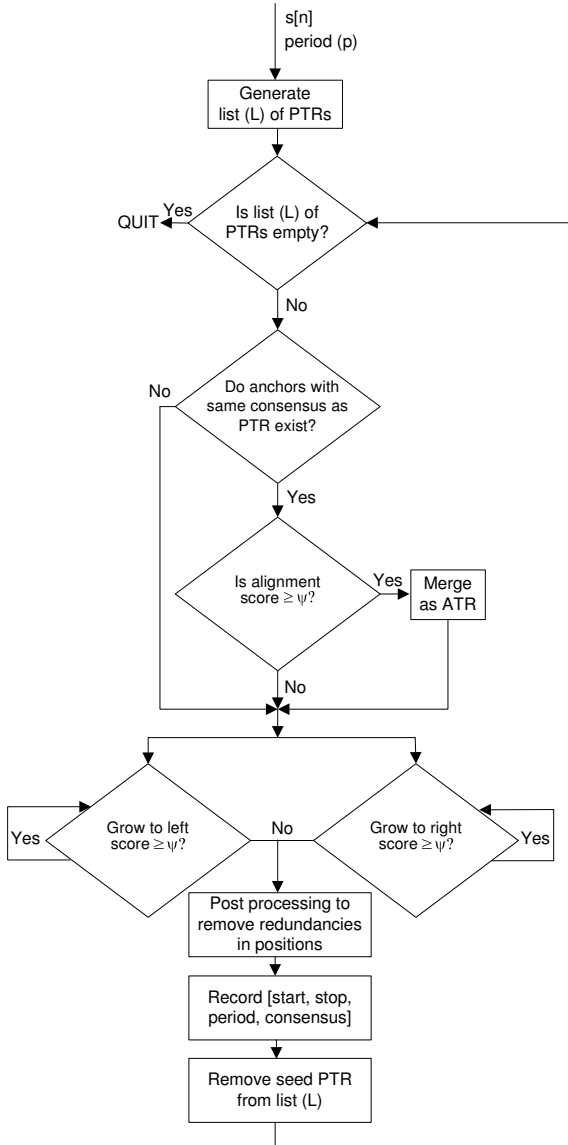


Fig. 1. Algorithm for determining local ATR regions

4.5. Local Periodicities

To find local periodicities, we perform mFPS analysis in a window to find the local frequency content. Given a window of length M , we calculate the mFPS of the first M sequence elements, and then detect all local ATRs corresponding to this window. Next, we advance the window by $M/2$, calculate the mFPS corresponding to sequence elements $M/2$ to $3M/2 - 1$, and find all local ATRs corresponding to this window. This process is continued until the end of the DNA sequence. In order to have the capability to detect a wide range of local ATR periods we scan the sequence using windows of varying sizes.

We must also be aware that there may be more than one local ATR in the window and that the local ATRs in the window may have different fundamental periods. To combat this problem, we propose an approach called the *iterative mFPS algorithm*. An outline of this approach is shown in Fig. 2. For the given window, the mFPS is calculated and we perform our test $l(\hat{S}_x(f))$. If the test fails, we move to the next window. If the test passes (local ATR present), then let p_0 be the initial period detected from the mFPS of the entire window. The algorithm first finds all local ATRs of period p_0 using the annotation algorithm shown in Fig. 1. Next, we rerun the mFPS for parts of the sequence that have not been marked as a local ATR. We then perform the test on $s[n]$ again, choosing the threshold γ based on the new window. Then we annotate these sub-windows based on detected local ATRs and continue to iterate exhaustively until the entire sequence within the original window is examined.

5. RESULTS AND DISCUSSION

In this section we present the results of the iterative mFPS algorithm on both simulated DNA cases and real DNA sequences. We compare our algorithm to mreps v2.5 [9] and Tandem Repeats Finder (TRF) v3.21 [10], since these two programs satisfy some of the characteristics of a good ATR finding algorithm listed in Table 1. Note that each algorithm has a different definition for an ATR, so how each handles local ATRs might be slightly different.

5.1. Simulated DNA

5.1.1. Heavily mutated sequence with comparison to mreps and TRF

To illustrate the robustness of the iterative mFPS compared to mreps and TRF, we took a sequence with the PTR pattern CTG and mutated it with a probability of mutation to a different nucleotide of $q = 0.1667$ such that the probability of non-mutation is $(1 - 3 \cdot q) = 0.5$. Such an experiment is motivated by the Jukes-Cantor model of evolution for DNA sequences [14]. We ran each trial with a randomly mutated sequence on all three programs.

With 50 trials, 37 sequences tested positive for repeats with period $p \geq 3$, albeit not necessarily the original PTR pattern of CTG. Repeat regions found by the programs are classified as the same if they have identical start, stop, and period size. The mFPS found 61 repeats compared to 62 repeats in mreps and 5 repeats in TRF. The iterative mFPS was able to find more ATRs than TRF with comparable performance to mreps in this set of cases.

It is important to note that the definition of sensitivity may vary depending on the program used, for example, mreps uses a resolution parameter equivalent to the analogy of a magnifying glass: the

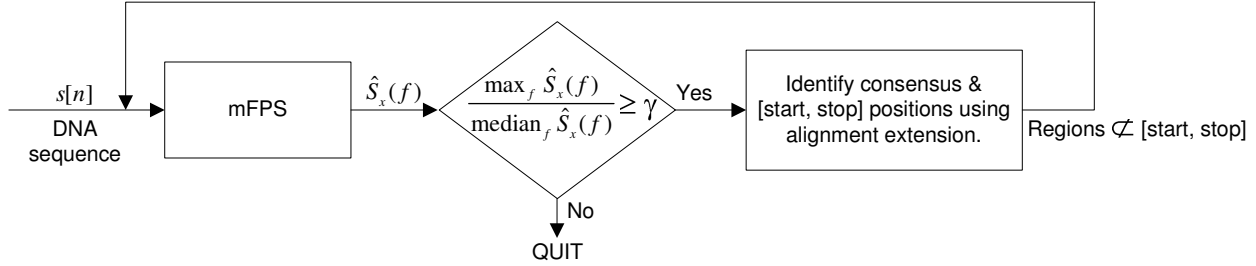


Fig. 2. Iterative mFPS algorithm overview

Fig. 3. Locus Y-27H39 (accession X77751)

```

1 ctactgagtt tctgttatag tgttttttaa
31 tatatatata gtattatata tatagtgta
61 tatatatata gtgtttttaga tagatagata
91 ggtagataga tagatagata gatagataga
121 tagatagata gatagataga tatagtgaca
151 ctctccttaa cccagatgga ctctctgtcc
181 tcactacatg ccat

```

smaller the value, the more strict the repeat pattern, the larger the value, the more fuzzy the repeat pattern can become. This makes it difficult to compare the two ATR programs in general. With the parameters chosen, the repeats found by mFPS tended to be longer stretches of ATR than that found by mreps. Table 4 shows a partial list of some of the longer repeats found by mFPS compared to mreps. Note that mreps may be able to find even longer repeat regions if the resolution of mreps is increased allowing mreps to detect more fuzzy repeats (i.e. longer ATRs with more mutations).

Table 4. Iterative mFPS and mreps repeats from section 5.1.1.

Consensus	mFPS repeat (mreps result in [·])
ctg	[tgctgctg]atgatccagccgc
atg	cgatgctgc[tgatgat]
ctg	[tgctgct]cctcc
ctg	[ctgctgc]agctgctgatgctcttg
ctg	ctgcggtt[ctgctgct]c

5.2. Real DNA sequences

5.2.1. Short tandem repeat polymorphism

Consider the human polymorphism at locus Y-27H39 (accession X77751), shown in Fig. 3. Imagine having to detect by hand, tandem repeats in this sequence without knowing the period size or consensus pattern! The mFPS program was able to expand the ATR repeat region to one larger than that currently known in the annotated Genbank sequence. Specifically, it was able to identify the ATR pattern GATA from position [79-140], compared to the Genbank annotation of [93-140].

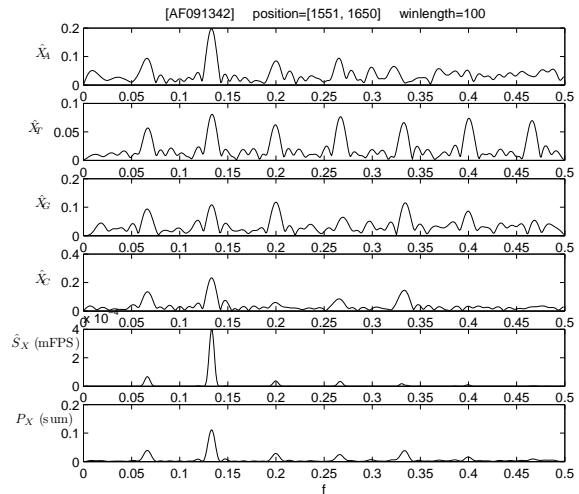


Fig. 4. From top to bottom: The indicator spectra $|\hat{X}_A(f)|$, $|\hat{X}_T(f)|$, $|\hat{X}_G(f)|$, $|\hat{X}_C(f)|$, the mFPS $\hat{S}_X(f)$, and the power spectrum $P_X(f)$ for the DNA sequence in section 5.2.2. Note that the ratio between the first and second peaks in the mFPS $\hat{S}_X(f)$ and $P_X(f)$ are 5.97 and 2.85, respectively.

5.2.2. Detection of period 15 repeats in cow sequence

Running the mFPS program as shown in Fig. 4 detected a period 15 repeat of consensus (AAGTCCCCAGAGGCA) in the *Bos taurus* (cow) neurofilament-M subunit (AF091342). Note that the actual period detected was $\frac{15}{2}$ but multiples of the period detected was also tested to discover the period $p = 15$.

5.2.3. Comparison with mreps and TRF for a polymorphic trinucleotide repeat disorder

In order to illustrate the strength of the mFPS program, we ran the spinocerebellar ataxia type 3 (SCA3) sequence (NM_004993) against mreps and TRF. SCA3 or Machado-Joseph Disease is an autosomal dominant trinucleotide repeat expansion disorder with CAG repeats in the coding region. Clinical symptoms of individuals affected by SCA3 are characterized by bulging eyes, small contracts of the facial muscles, and general rigidity.

Figure 5 summarizes the results from all three algorithms. A

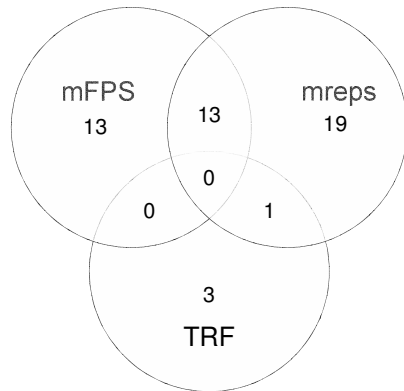


Fig. 5. Venn diagram to compare results of ATR algorithms for the Machado-Joseph disease sequence in Section 5.2.3.

repeat is classified as being the same if the programs produced identical start, stop, and period size results. All the results were filtered for period $p \geq 3$.

Slight variations in start and stop positions accounted for the large number of different repeats found among the three programs. All programs were able to detect the trinucleotide repeat CAG in the SCA3 sequence as shown in Table 5. The mreps program detected chunks of the repeat but was not able to find a single region encompassing the positions listed in the Genbank annotation. Subsequent trials with higher resolution parameters yielded better results but only after trial and error. Even with the most sensitive parameters, TRF was only able to detect 4 repeats with period $p \geq 3$. This example helps to illustrate how mFPS achieves a fair compromise between the best of both worlds by being sensitive enough to detect other repeat regions that may be of biological importance missed in TRF without the need to experiment with a sensitivity parameter like that in mreps.

Table 5. Results of ATR repeats of (CAG) found in SCA3 (NM_004993)

Source	Start	Stop	Consensus
Genbank	932	973	CAG
mFPS	930	974	CAG
mreps	932	939	CAG
	942	948	AGC
	950	973	CAG
TRF	932	973	CAG

6. CONCLUSION

We have shown some success in applying a classic signal processing method, the Fourier transform, to the analysis of DNA sequences. The application of Fourier techniques to the problem of finding ATRs in DNA is not trivial. When no prior knowledge about the presence/absence of the ATR, the period, or the repeat pattern is assumed, many practical matters must be considered. We propose an iterative mFPS method which is capable of finding repeats that two premier ATR finding algorithms, mreps and

Tandem Repeat Finder, cannot find. Likewise, these methods also found repeats that elude our algorithm. This is likely due to the fact that our proposed method, [9], and [10] are developed with very different formulations of the problem. From our results, we conclude that for a thorough investigation into the repeat content of a DNA sequence, one should use all three programs.

7. REFERENCES

- [1] E S Lander et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, February 2001.
- [2] H Herzel, O Weiss, and E N Trifonov, "10-11 bp periodicities in complete genomes reflect protein structure and DNA folding," *Bioinformatics*, vol. 15, no. 3, pp. 187–193, 1999.
- [3] C Ruitberg, D Reeder, and J Butler, "Strbase: a short tandem repeat DNA database for the human identity testing community," *Nucleic Acids Research*, vol. 29, pp. 320–322, 2001.
- [4] Y Nakamura et al., "Variable number of tandem repeat (vnr) markers for human genome mapping," *Science*, vol. 235, pp. 1616–1622, 1987.
- [5] T T Tran, V A Emanuele II, and G T Zhou, "Techniques for detecting approximate tandem repeats in DNA," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004, vol. 5, pp. 449–452.
- [6] R D Wells, "Molecular basis of genetic instability of triplet repeats," *Journal of Biological Chemistry*, vol. 271, pp. 2875–2878, 1996.
- [7] M Mitas, "Trinucleotide repeats associated with human disease," *Nucleic Acids Research*, vol. 25, pp. 2245–2253, 1997.
- [8] D Hanahan and R A Weinberg, "The hallmarks of cancer," *Cell*, vol. 100, pp. 57–70, 2000.
- [9] R Kolpakov, G Bana, and G Kucherov, "mreps: efficient and flexible detection of tandem repeats in DNA," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3672–3678, 2003.
- [10] G Benson, "Tandem repeats finder: a program to analyze DNA sequences," *Nucleic Acids Research*, vol. 27, no. 2, pp. 573–580, 1999.
- [11] S Tavaré and B W Giddings, "Some statistical aspects of the primary structure of nucleotide sequences," in *Mathematical Methods for DNA Sequences*, Michael S Waterman, Ed., pp. 117–131. CRC Press, Boca Raton, Florida, 1989.
- [12] S Tiwari, S Ramachandran, A Bhattacharya, S Bhattacharya, and R Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *CABIOS*, vol. 13, pp. 262–270, 1997.
- [13] M Z Ikram and G T Zhou, "Estimation of multicomponent polynomial phase signals of mixed orders," *Signal Processing*, vol. 81, no. 11, pp. 2293–2308, October 2001.
- [14] R Durbin, S Eddy, A Krogh, and G Mitchison, *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*, Cambridge University Press, Cambridge, UK, 1998.
- [15] S B Needleman and C D Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.