A Fourier-Based Data Minimization Algorithm for Fast and Secure Transfer of Big Genomic Datasets

Mohammed Aledhari*, Marianne Di Pierro†, and Fahad Saeed*
*Department of Computer Science
Western Michigan University, Kalamazoo, MI, 49008-5466,
Correspondence should be addressed to Mohammed Aledhari at mohammed.aledhari@wmich.edu
† Graduate College
Western Michigan University, Kalamazoo, MI, 49008-5466

***Abstract*—DNA sequencing plays an important role in the bioinformatics research community. DNA sequencing is important to all organisms, especially to humans and from multiple perspectives. These include understanding the correlation of specific mutations that plays a significant role in increasing or decreasing the risks of developing a disease or condition, or finding the implications and connections between the genotype and the phenotype. Advancements in the high-throughput sequencing techniques, tools, and equipment, have helped to generate big genomic datasets due to the tremendous decrease in the DNA sequence costs. However, the advancements have posed great challenges to genomic data storage, analysis, and transfer. Accessing, manipulating, and sharing the generated big genomic datasets present major challenges in terms of time and size, as well as privacy. Data size plays an important role in addressing these challenges. Accordingly, data minimization techniques have recently attracted much interest in the bioinformatics research community. Therefore, it is critical to develop new ways to minimize the data size. This paper presents a new real-time data minimization mechanism of big genomic datasets to shorten the transfer time in a more secure manner, despite the potential occurrence of a data breach. Our method involves the application of the random sampling of Fourier transform theory to the real-time generated big genomic datasets of both formats: FASTA and FASTQ and assigns the lowest possible codeword to the most frequent characters of the datasets. Our results indicate that the proposed data minimization algorithm is up to 79% of FASTA datasets' size reduction, with 98-fold faster and more secure than the standard data-encoding method. Also, the results show up to 45% of FASTQ datasets' size reduction with 57-fold faster than the standard data-encoding approach. Based on our results, we conclude that the proposed data minimization algorithm provides the best performance among current data-encoding approaches for big real-time generated genomic datasets.**

I. Introduction

DNA sequencing is needed in the most critical areas such as criminal investigations, genotyping and determination of disease-relevant genes or agents causing diseases, mutation analysis, screening of single nucleotide polymorphisms (SNPs), detection of chromosome abnormalities [1], and to identify disease- and/or drug-associated genetic variants to advance precision medicine [2] [3]. Also, the use of high-throughput DNA sequencing instruments, such as next-generation sequencing (NGS) technologies that include whole-genome sequencing (WGS) and whole-exome sequencing (WES), significantly decreases the sequencing costs and enables the genomic datasets to join the big data club. Those instruments became big data generators, not only for big biology centers, but also for small biology laboratories and researchers.

Also, many genome related projects, such as the 1000-Genomes project [4] and the international cancer sequencing consortium [5], suggest using the cloud as an infrastructure to solve the store and analysis challenges [6], [7], [8]. However, the transfer and share of the genomic datasets between biological laboratories and from/to the cloud represents an ongoing bottleneck because of the amount of data and the limitations of the network bandwidth [9]. Therefore, transfer challenges can be solved by either increasing the bandwidth or minimizing the data size during the transfer phase. These elements represent our contributions to the research and are the focus of this paper.

A. Contribution

This paper discusses the design and implementation of a novel data minimization algorithm for the real-time generated big genomic datasets that relies on a combination of the random sampling of Fourier transform [10] and variable-length binary-encoding [11]. This work aims to minimize and secure big genomic datasets during the data transfer phase over the networks, either via wire or wireless. We assert that creating a new minimization mechanism for big genomic datasets can significantly shorten the transfer time and secure the data at the same time by changing the data-encoding of the datasets multiple times during the transfer phase. Consequently, the transfer time will decrease tremendously, in addition to protecting the data against a potential breach by changing the codewords of the genomic symbols frequently. Therefore, we can ensure that the data will transfer faster and in a more secure way. This paper represents an extension of our previous research papers in [12], [13], and [14].

B. Motivation

It is not a minor challenge to transfer big data like genomic datasets faster than the current tool allow. This is because all data transfer protocols, such as Hyper Text Transfer Protocol (HTTP) [15] and File Transfer Protocol (FTP) [16], use the standard content-encoding schemes, such as the American

Standard Code for Information Interchange (ASCII) [17]. Yet, the decrease in the costs of DNA sequencing due to the advancement of the DNA sequence techniques and equipment, encourages biologists to extend their research and produce increasingly greater numbers of genomic datasets that need to be manipulated and transferred between various biology laboratories. Such uses for DNA sequencing data include critical areas such as criminal investigations, genotyping and determination of disease-relevant genes or agents causing diseases, mutation analysis, screening of single nucleotide poly-morphisms (SNPs), and detection of chromosome abnormalities [18]. As a result, there is a high demand to transfer and share data faster than the current transfer time. However, shortening the transfer time of big datasets is complicated because all Internet browsers use the standard content-encoding schemes in terms of symbol lengths, such as 8-bit, 9-bit, 16-bit, 32-bit codewords. Data minimization and privacy are also very important to both users and service providers, especially for remote healthcare services. Many solutions, such as the cloud, have been developed to address the challenges of remote diagnostic devices. Yet, data minimization and privacy challenges have not been addressed at the same level, mainly due to compatibility issues. As a result, scientists are motivated to navigate and search for new methods to transfer big genomic data more efficiently and in a fully secured environment. Current data transfer protocols are not suitable for the increased growth of big data and cloud-based services such as remote healthcare diagnostics due to the use of transfer protocols that belong to different vendors. Observing these facts, we take advantage of the nature of the genomic symbols to implement a more efficient content-encoding algorithm to minimize the data amount during the transfer phase.

C. Paper Goals and Organization

The purpose of this paper is to design and implement a novel data minimization algorithm to decrease the required time to transfer big genomic datasets over networks. In addition, we increased security in the event of a data breach. Moreover, it will introduce a generic concept that can be used by other limited symbols alphabet of the cloud-based applications to secure data that exchange remotely. The contributions of this paper are outlined as follows:

- Summarizes the data minimization and data encoding methods with quick fundamentals, sans the need to search through the details presented in the standards' specifications.
- Provides an overview of the challenges of the big genomic datasets in terms of transfer time with extra protection if data breach occurs.
- Presents the need for better data minimization techniques to provide faster data transfer, especially cloud-based services.

The remainder of this paper is organized as follows: Section II presents preliminaries in terms of data minimization methods. Section III discusses the model description and formulation. Section IV presents the experiments and results of the proposed data minimization algorithm. Section V presents our conclusions.

II. Preliminaries

Data minimization can be divided into five main mechanisms as follows:

A. Naive Bit Encoding

This approach works by assigning fixed-length codeword/binary representation to each alphabet symbol in a way that represents more than a single symbol in a single byte, such as 2-bit length to genomic symbols [19] and [20], as shown in Figure 1-(a) on page 3.

B. Dictionary-based/Substitutional Encoding

This approach stores different patterns of the input symbols in a dictionary or a database, along with their codewords, and then replaces the new input parts with predefined portions [21] and [22], such as in 1977-78 Ziv and Lempel (LZ-77) [23] and [24], as shown in Figure 1-(b) on page 3. LZ-77 algorithm works by replacing the repeated occurrences of symbols with their references that indicate length and location of that string, which occurred before, and which can be presented in the tuple (offset, length, symbol).

C. Statistical/Entropy Encoding

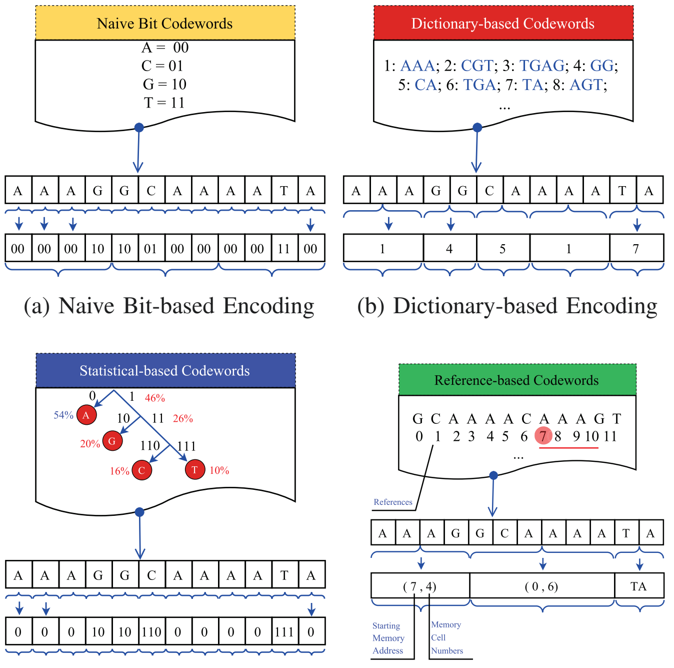
This approach works by statistics, prediction, and a probabilistic model from the input [25] and [26], such as Huffman's coding [23] and [27], as shown in Figure 1-(c) on page 3. Huffman's coding, introduced in 1952, is a statistical method that assigns a fixed-length codeword/binary representation to alphabet symbols, such as 2-bit, 3-bit, 8-bit, etc. The codewords will have different lengths, and the lowest frequency symbols will be assigned with the longest codewords and vice versa.

D. Referential/Reference-based Encoding

This approach is similar to a dictionary-based technique, except that it uses the pointer to the internal and external references, as shown in Figure 1-(d) on page 3.

E. Hybrid Encoding

This approach works by combining two or more encoding methods. For example, The Burrows-Wheeler transform (BWT) [28] [23] and [29], is one of the hybrid encoding methods, especially popular in bioinformatics, used for data minimization. The BWT method works by permuting the input sequence in a way that symbols are grouped by their neighborhood. Our proposed data minimization algorithm can be classified as a hybrid encoding method by incorporating elements of the standard (8-bit) and the statistical encoding methods (variation of 1 - 3 bits.



(a) Naive Bit-based Encoding (b) Dictionary-based Encoding

(c) Statistical-based Encoding (d) Reference-based Encoding

Figure 1: Data-encoding methods

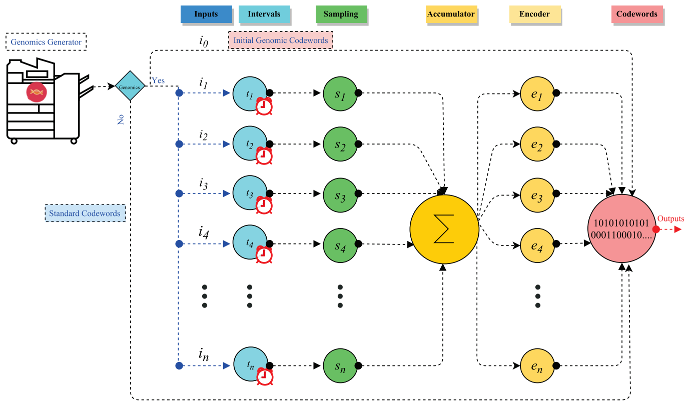
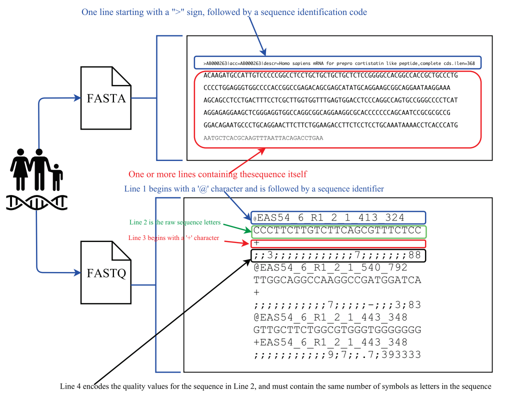


Figure 2: The Proposed Encoding System Framework

III. Model Description and Formulation

A. Model description

This subsection presents our implementation of data minimization algorithm for the generation of big real-time genomic DNA datasets using DNA-sequencing equipment such as NGS. The generated datasets can have many formats, but the common ones are FASTA [30] and FASTQ [31] formats, as illustrated in Figure 3 on page 3. This work has been done to minimize the size of big genomic datasets and then shorten the transfer time. Also, this work adds an extra security layer via changing the symbol codewords several times for each dataset during the transfer phase. This model ensures the assignment of the lowest possible codewords for more symbol occurrences via dividing the datasets into parts based on time and by utilizing a random sampling of Fourier transform theory, as



One line starting with a ">" sign, followed by a sequence identification code
FASTA
One or more lines containing thesequence itself
Line 1 begins with a '@' character and is followed by a sequence identifier
Line 2 is the raw sequence letters
Line 3 begins with a '+' character
FASTQ
Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence

```
>AB000263|acc=AB000263|descr=Homo sapiens mRNA for prepro cortistatin like peptide,complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTG
CCCCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAA
AGCAGCCTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCAT
AGGAGAGGAAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCG
GGACAGAATGCCCTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATG
AATGCTCACGCAAGTTTAATTACAGACCTGAA
```

```
@EAS54_6_R1_2_1_413_324
CCCTTCTTGTCTTCAGCGTTTCTCC
+
;;3;;;;;;;;;;;;7;;;;;;;88
@EAS54_6_R1_2_1_540_792
TTGGCAGGCCAAGGCCGATGGATCA
+
;;;;;;;;;;;7;;;;;-;;;3;83
@EAS54_6_R1_2_1_443_348
GTTGCTTCTGGCGTGGGTGGGGGGG
+EAS54_6_R1_2_1_443_348
;;;;;;;;;;;9;7;;.7;393333
```

Figure 3: The FASTA and FASTQ format components

illustrated in Figure 2 on page 3. Two encoding schemes are used in this model: standard and modified. The standard one uses the dataset headers and non-DNA symbols in the dataset bodies. The modified encoding scheme starts with an initial codeword, and then updates frequently via running a time-based sampling using a Fourier transform theory. The initial codewords of the proposed encoding codewords, such as those shown in Figure 1 (c) on page 3, periodically updates codewords via random sample of Fourier theory, as illustrated in Figure 2 on page 3. The data-encoding switches between the standard and proposed ones by sending 25-byte of 1's from the sender to the receiver, indicating that the alternative codewords will be used from this point on, and continuously, until receiving another data-encoding scheme. Therefore, this model encodes the contents using two main codeword tables: fixed and variable (changes many times), as shown in Figure 2 on page 3. The proposed data-encoding scheme of the genomic dataset benefits from being in the alphabet that can be encoded in four unique decipherable codewords, i.e. [0, 11, 100, 101] or simply [0, 3, 4, 5]. The result of the sampling phase adds to the encoding accumulator in an array form, and then calculates symbol repetitions and creates the new encoding tree that is used in the data-encoding phase. We utilize a binary tree as a structure to represent our genomic data-encoding because it is faster to search, avoids duplicate values, and facilitates decoding at the receiver side. Also, the use of a binary tree, as a structure, gives the programmer special flexibility because it offers one of the two paths to follow and cuts the search time in half, thereby increasing process throughput.

B. Problem formulation

We are using the random sampling of Fourier transform theory to update the content-encoding tree for big genomic symbols. Accordingly, we need to find the best representation of Fourier i_{best} of X complex exponential terms to approximate i for a given a moment $i \in \mathbb{C}^N$. The fast Fourier transform

(FFT) can be applied to find the i within X largest terms. Fourier representation i* can be found by sampling a subset $\tau \subseteq [0, N-1]$ of i according to [32] as follows:

$$||i - i||_2^2 \leq (1 + \epsilon)||i - i_{best}||_2^2 \quad (1)$$

the ϵ refers to the error bound. Applying independent Bernoulli trials on the set $[0, N$ - 1] with a fixed probability to set the τ We can find the amount of the sampling rate by employing i in the [33] as follows:

$$i[n] = \frac{1}{\sqrt{N}} \sum_{x=0}^{X-1} \alpha_{x^{e^{j2\pi\omega_x n/N}}}, w_x \subseteq [0, N-1]. \quad (2)$$

That can be written in an array form such as i = $\mathbf{F}\alpha$, for the discrete Fourier transform (DFT) array $\mathbf{F}$, the elements of the DFT can be found using the formula $\mathbf{F}_{w,t} = \frac{1}{\sqrt{N}} e^{j2\pi\omega t/N}, \omega, t = 0, ..., N-1$, and α only X non-zero values in the regular repetitions ω_x. The goal is to recover α from the random samples of i. The sampling pool generator τ is the Fourier random sampling theory. The generated elements M of the sampling pool τ. Therefore, we can obtain each sample cycle through the M/N or discarded with probability $1 - M/N$. The sparse vector α of time slots can be conducting with high probability if $M = O(Xlog^4 N)$ [34] that utilizes geometric probability distributions. This paper applies the Fourier random sampling to generate updated encoding tree of genomic datasets to minimize the data size during transfer phase. Also, we do a continue randomize of sampling intervals using the following theories [35] [36] [37] [38] [39] . Then the random interval sampling $s(t)$ is defined as

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - t_n). \quad (3)$$

The random operation $i(t)$ can be sampled using $s(t)$ can be written as $y(t) = i(t)s(t)$. If t_n is independent from $i(t)$, then

$$\Phi_y(f) = \Phi_i(f) * \Phi_s(f), \quad (4)$$

where $\Phi_y(f), \Phi_i(f)$, and $\Phi_s(f)$ are the power spectral densities (PSD) of $y(t), i(t)$ and $s(t)$, respectively. When $t_n = nT$,

$$\Phi_s(f) = \frac{1}{T^2} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T}). \quad (5)$$

Therefore, the periodic sampling of the encoding tree of the genomics alphabet $\Phi_s(f)$ can use [36] to obtain the following theorem.

Theorem 3.1

Let's assume the β is the average of the sampling cycles, we can find $1/E[\tau_x]$, where τ_x refers to the independently and identically distributed (i.i.d) intervals between samples. If the characteristic function of τ_x is $\psi_{\tau_x}(f)$, then

$$\Phi_s(f) = \beta \Re \left\{ \frac{1 + \psi_{\tau_x}(2\pi f)}{1 - \psi_{\tau_x}(2\pi f)} \right\}. \quad (6)$$

Since $\psi_{\tau_x}(0) = 1$, $Phi_s(f)$ will have value $f = 0$.

The sampling time t_x can be found using the collective random sampling (CRS) in the following equation

$$t_x = t_{x-1} + \tau_x, \quad (7)$$

applying the exponential distribution $\sim$ Exp(λ) to the i.i.d interval τ_x, we can get the following equation

$$\Phi_s(f) = \lambda^2 \delta(f) + \lambda. \quad (8)$$

When τ_x is uniformly distributed $\tau_x \sim$ Uniform[a, b],

$$\Phi_s(f) = \begin{cases} P(\rho \frac{\sin((b-a)\pi f)}{(b-a)\pi f}), (b-a)\pi f & f \neq 0 \\ (\frac{2}{a+b})^2 \delta(f) & f = 0 \end{cases}, \quad (9)$$

where $P(r, \theta)$ is a Poisson kernel defined as:

$$P(r, \theta) = \frac{1 - r^2}{1 - 2r\cos\theta + r^2}, \quad (10)$$

Moreover, Implementing periodically sampling is not a minor challenge because the sampling intervals are usually divided into fixed time slots determined mostly by the CPU clock speed. Therefore, we can use rounding up Δ to specify the sampling intervals τ_x^q such that:

$$\tau_x^q = n\Delta, if (n-1)\Delta < \tau_x \leq n\Delta, \qquad n \in \Omega, \quad (11)$$

where Ω refers to the adequate integers that can be used. For instance, we can get the Ω = 1, 2,... when the the exponential distribution is used to the τ_x^q. Whereas, the $\Omega = \{1, 2, ..., \lfloor \frac{T}{\Delta} \rfloor\}$ in case of using the uniform distribution in $[0, T]$ to the τ_x^q. The time slots of the sampling of τ_x can be found via their characteristic function $\psi_{\tau_x}(f)$. Accordingly, we ensure that the $\Phi_s(f)$ becomes periodic synchronizing to the periodicity of the $\frac{1}{\Delta}$. Consequently, we can ensure getting the best possible time slots of sampling rates only if $i(t)$ place within $[-\frac{1}{2\Delta}, \frac{1}{2\Delta}]$ limits. In other words, we can obtain the best possible encoding tree with minimum sampling intervals that includes the highest frequency of genomics symbols.

IV. Experiments and Results

In this section, we discuss the size and transfer time of the simulated big genomic datasets by using the proposed and standard data-encoding schemes over HTTP. The examined datasets in both formats, FASTA and FASTQ, were generated via our simulated NGS generator, as can been seen in Tables II on page 5 and III on page 6.

A. Experimental setup

This paper applies the proposed and standard data-encoding schemes during transfer of big genomic datasets using HTTP. Several sizes of genomic datasets in both formats, FASTA and FASTQ, were generated using our NGS simulation and tested up to 1TB to validate our data-encoding. The experiments were performed on machines that have specifications shown in Table I on page 5.

Table I: Experimental setup

Specifications	Details
Processor	2.4 GHz Intel Core i7
Memory	8 GB 1600 MHz DDR3
Graphics	Intel HD Graphics 4000 1024 MB
Operating System	Windows 8.1 Pro
Download	87 Mb/s
Upload	40 Mb/s
Programming Language	C# .Net
Used Protocol	HTTP
Dataset Sizes	1GB,5GB,10GB,20GB,50GB,100GB,200GB,400GB,500GB,1TB

B. Experimental Results

Our experimental results of big genomic datasets are shown in Tables II on page 5, III on page 6, IV on page 6, and V on page 6, and Figures 4 on page 5, 5 on page 5, 6 on page 6, and 7 on page 6.
In order to assess the effectiveness of our proposed data minimization method, several genomic dataset sizes were tested using both data-encoding methods: propose and standard during transfer using HTTP. The results show that the propose method significantly decreases the dataset sizes, and then shortens the transfer time, as shown in Tables and Figures. For example, the original 1GB dataset in FASTA format was minimized to 293393 KB using the proposed method during transfer phase in contrast to 1048576 KB using the standard encoding method with about 72% size reduction. Consequently, the transfer time was 5255 milliseconds (ms) using the proposed method, while 2.54 x $10^5$1 ms resulted by using the standard scheme with about 98-fold of time acceleration. Also, 223136461 KB were transferred of 1TB FASTA dataset using the proposed method, while 1073741824 KB resulted by using the standard method for the same dataset with about 79% size reduction. The transfer time of 1TB FASTA dataset using the proposed data-encoding were 8 x 10^6 ms, while 1.79 x 10^8 ms resulted in the transfer of the same dataset using the standard scheme, with about 96-fold of time acceleration. Also, the 1GB FASTQ dataset was minimized to 356516 KB by using the proposed scheme, while 1048576 KB resulted by using the standard scheme, with about 34% of size reduction. Consequently, 8333 ms were needed to transfer the dataset by using the proposed method, while 15150 ms were required to transfer the same dataset using the standard approach, with 55-fold of time acceleration. Moreover, 1TB FASTQ dataset was minimized to 418759311 KB by using a proposed encoding method, while 1073741824 KB transferred using the standard approach with about 39% of size reduction. The time acceleration of the 1TB FASTQ dataset was 54-fold by sending the dataset in 7.74 x 10^6 ms using the proposed method, while 143 x 10^7 ms were needed to transfer the same dataset. The maximum size reduction of examined FASTA datasets using the proposed data-encoding scheme is 79% and about 98-fold of time acceleration, The maximum rate of size reduction for the examined FASTQ datasets using the proposed encoding method is 45%, while 57-fold is the maximum time acceleration.

Table II: Simulated Next-generation sequencing (NGS) Dataset Sizes, FASTA Format

Datasets	Data-Encoding Size in KB		Number of Genomic Symbols in each Dataset			
	Propose	Standard	A	C	G	T
1GB	293393	1048576	536870912	214748160	214748160	107374592
5GB	1181950	5242880	3758095360	268431360	429496320	912686080
10GB	2731022	10485760	6442444800	1610608640	858992640	1825372160
20GB	4722414	20971520	15461867520	2147471360	1717985280	2147512320
50GB	14369917	52428800	26843545600	13421772800	9663641600	3758131200
100GB	23321911	104857600	73014374400	10737356800	2147430400	21475020800
200GB	53935177	209715200	1.33144E+11	21474713600	25769779200	34359910400
400GB	99176254	419430400	2.96353E+11	47244492800	42949427200	42950246400
500GB	127617346	524288000	3.22122E+11	1.07374E+11	96636416000	10738176000
1TB	223136461	1073741824	7.91648E+11	1.31941E+11	1.64926E+11	10996416512

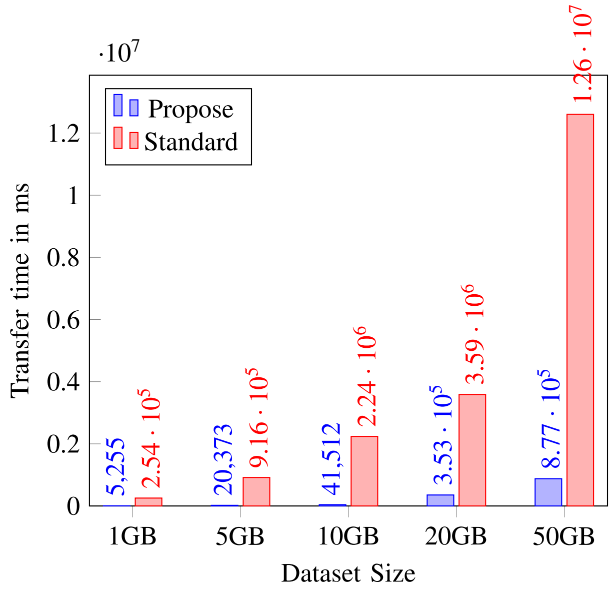


Figure 4: Transfer time of FASTA datasets: 1G, 5GB, 10GB, 20GB, and 50GB

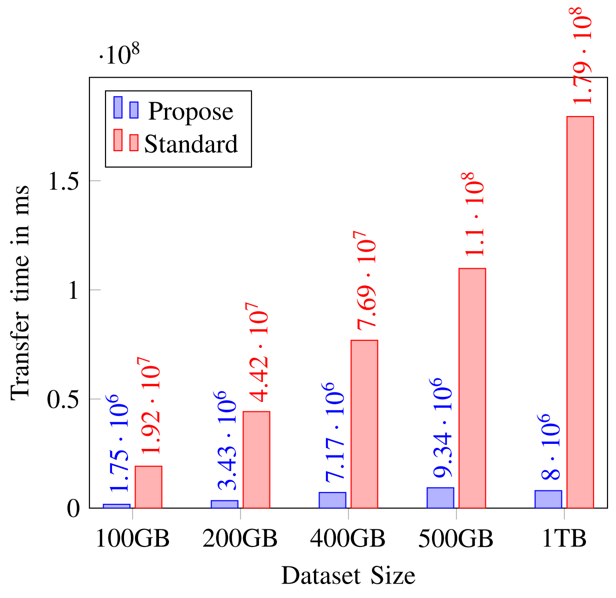


Figure 5: Transfer time of FASTA datasets: 100G, 200GB, 400GB, 500GB, and 1TB

Table III: Simulated Next-generation sequencing (NGS) Dataset Sizes, FASTQ Format

Datasets	**Data-Encoding Size in KB**		**Number of Genomic Symbols in each Dataset**			
	Propose	**Standard**	**A**	**C**	**G**	**T**
1GB	356516	1048576	1100243	1171475	1194506	1070103
5GB	2097152	5242880	4587490	4722908	4853198	4293654
10GB	3879731	10485760	10579317	10871037	11069642	10603326
20GB	8808038	20971520	18383332	19099195	19068835	16777600
50GB	20971520	52428800	55189255	58677592	55350383	55861160
100GB	33554432	104857600	90769611	98405480	100395898	93309194
200GB	79691776	209715200	202744248	208871811	217755796	209397412
400GB	188743680	419430400	387005677	391542999	392718494	375820583
500G	214958080	524288000	490035888	508771266	550501004	490664651
1TB	418759311	1073741824	864354300	910476782	892603677	768828857

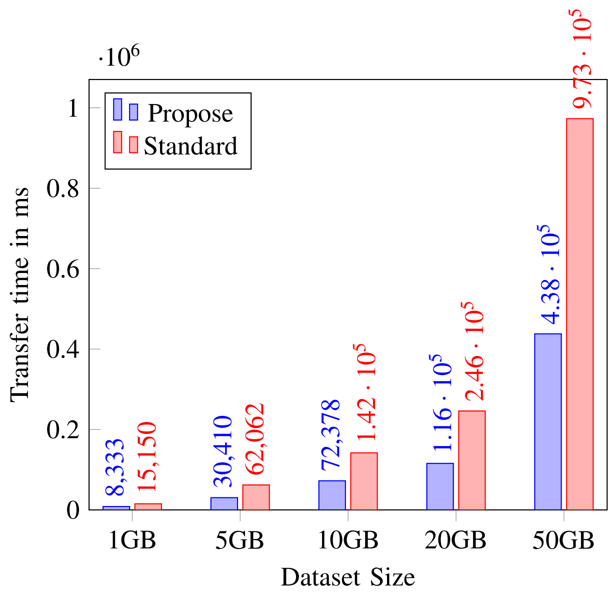


Figure 6: Transfer time of FASTQ datasets: 1G, 5GB, 10GB, 20GB, and 50GB

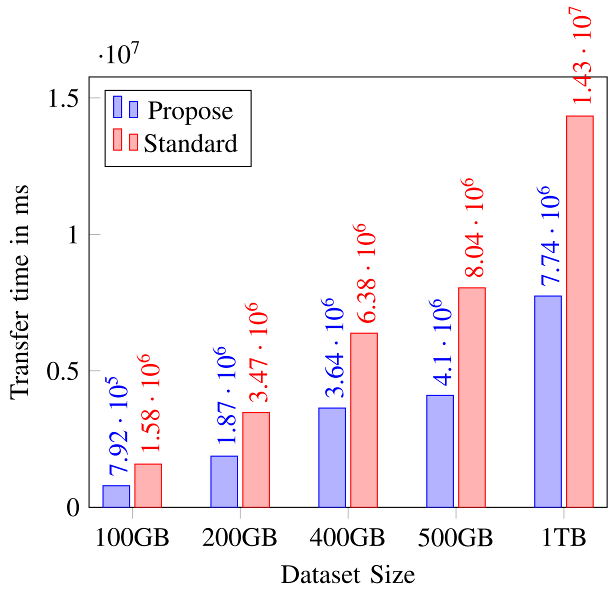


Figure 7: Transfer time of FASTQ datasets: 100G, 200GB, 400GB, 500GB, and 1TB

Table IV: Size Reduction and Time Acceleration of Datasets, FASTA Format

Dataset	Size Reduction	Time Acceleration
	Propose vs Standard	Propose vs Standard
1GB	72%	98x
5GB	77%	98x
10GB	74%	98x
20GB	77%	90x
50GB	73%	93x
100GB	78%	91x
200GB	74%	92x
400GB	76%	91x
500GB	76%	91x
1TB	79%	96x

Table V: Size Reduction and Time Acceleration of Datasets, FASTQ Format

Dataset	Size Reduction	Time Acceleration
	Propose vs Standard	Propose vs Standard
1GB	34%	55x
5GB	40%	49x
10GB	37%	51x
20GB	42%	47x
50GB	40%	45x
100GB	32%	50x
200GB	38%	54x
400GB	45%	57x
500GB	41%	51x
1TB	39%	54x

V. Conclusion

In this paper, we implemented an innovative data minimization algorithm that relies on a combination of the naive bit encoding approach and the random sampling of Fourier transform theory to form a time-based random sampling data-encoding. The proposed algorithm is designed to be integrated with transfer protocols to reduce the size of the real-time big genomic datasets in both formats: FASTA and FASTQ, during the transfer phase, and then to transfer the data securely in a shorter time. The results indicate that the proposed data minimization algorithm is capable or reducing the transfer time up to 98-fold of FASTA datasets and 57-fold of FASTQ datasets, compared to the standard data-encoding on tested datasets. Further, the results also show the dataset can be reduced up to 79% of the original FASTA format dataset sizes and up to 45% of FASTQ format. The proposed data minimization approach is integrated to HTTP to assess how the transfer protocol behaves in terms of transfer time and size. Also, we showed that our data minimization algorithm provides a significant size reduction and transfer time acceleration, as well as adds an extra level of security during the transfer because dataset symbols encode in different codewords during the transfer phase. We demonstrated that the data size can be significantly reduced by our hybrid data-encoding, compared to a standard scheme. Also, we implemented an NGS simulator to generate genomic datasets in both formats: FASTA and FASTQ, to verify the performance of our current data minimization scheme and compared it to the standard. Our experiments indicated that the Fourier transform theory-based random sampling, data-encoding performs much better than the standard scheme by ensuring the assignment of short codewords for the genomic dataset symbols. We conclude that the proposed data minimization algorithm provides the best performance among current data-encoding approaches for big real-time generated genomic datasets.

Acknowledgment

The authors thank Dr. Todd Barkman, a professor in the Department of Biological Sciences at Western Michigan University for his valued feedback and comments. This work was supported in part by grants US National Science Foundation CRII CCF-1464268 and US National Science Foundation CAREER ACI-1651724.

References

[1] C. Mora, D. P. Tittensor, S. Adl, A. G. Simpson, and B. Worm, "How many species are there on earth and in the ocean?" *PLoS biology*, vol. 9, no. 8, p. e1001127, 2011.

[2] F. S. Collins and H. Varmus, "A new initiative on precision medicine," *New England Journal of Medicine*, vol. 372, no. 9, pp. 793–795, 2015.

[3] T. C. Carter and M. M. He, "Challenges of identifying clinically actionable genetic variants for precision medicine," *Journal of healthcare engineering*, vol. 2016, 2016.

[4] . G. P. Consortium *et al.*, "A map of human genome variation from population-scale sequencing," *Nature*, vol. 467, no. 7319, pp. 1061–1073, 2010.

[5] T. J. Hudson, W. Anderson, A. Aretz, A. D. Barker, C. Bell, R. R. Bernabé, M. Bhan, F. Calvo, I. Eerola, D. S. Gerhard *et al.*, "International network of cancer genome projects," *Nature*, vol. 464, no. 7291, pp. 993–998, 2010.

[6] V. A. Fusaro, P. Patil, E. Gafni, D. P. Wall, and P. J. Tonellato, "Biomedical cloud computing with amazon web services," *PLoS computational biology*, vol. 7, no. 8, p. e1002147, 2011.

[7] M. C. Schatz, B. Langmead, and S. L. Salzberg, "Cloud computing and the dna data race," *Nature biotechnology*, vol. 28, no. 7, p. 691, 2010.

[8] L. D. Stein, "The case for cloud computing in genome informatics," *Genome biology*, vol. 11, no. 5, p. 207, 2010.

[9] O. Trelles, P. Prins, M. Snir, and R. C. Jansen, "Big data, but are we ready?" *Nature Reviews Genetics*, vol. 12, no. 3, pp. 224–224, 2011.

[10] K. Kazimierczuk, A. Zawadzka, W. Koźmiński, and I. Zhukov, "Random sampling of evolution time space and fourier transform processing," *Journal of biomolecular NMR*, vol. 36, no. 3, pp. 157–168, 2006.

[11] E. N. Gilbert and E. F. Moore, "Variable-length binary encodings," *Bell Labs Technical Journal*, vol. 38, no. 4, pp. 933–967, 1959.

[12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys &amp; Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[13] M. Aledhari and F. Saeed, "Design and implementation of network transfer protocol for big genomic data," in *Big Data (BigData Congress), 2015 IEEE International Congress on*. IEEE, 2015, pp. 281–288.

[14] M. Aledhari, M. S. Hefeida, and F. Saeed, "A variable-length network encoding protocol for big genomic data," in *International Conference on Wired/Wireless Internet Communication*. Springer, 2016, pp. 212–224.

[15] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–http/1.1," Tech. Rep., 1999.

[16] J. Postel and J. Reynolds, "File transfer protocol," 1985.

[17] S. Gorn, R. W. Bemer, and J. Green, "American standard code for information interchange," *Communications of the ACM*, vol. 6, no. 8, pp. 422–426, 1963.

[18] M. A. Jobling and P. Gill, "Encoded evidence: Dna in forensic analysis," *Nature Reviews Genetics*, vol. 5, no. 10, pp. 739–751, 2004.

[19] S. Grumbach and F. Tahi, "A new challenge for compression algorithms: genetic sequences," *Information Processing &amp; Management*, vol. 30, no. 6, pp. 875–886, 1994.

[20] L. Chen, S. Lu, and J. Ram, "Compressed pattern matching in dna sequences," in *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*. IEEE, 2004, pp. 62–68.

[21] N. J. Larsson and A. Moffat, "Off-line dictionary-based compression," *Proceedings of the IEEE*, vol. 88, no. 11, pp. 1722–1732, 2000.

[22] Y. Shibata, T. Matsumoto, M. Takeda, A. Shinohara, and S. Arikawa, "A boyer—moore type algorithm for compressed pattern matching," in *Annual Symposium on Combinatorial Pattern Matching*. Springer, 2000, pp. 181–194.

[23] D. Salomon and G. Motta, *Handbook of data compression*. Springer Science &amp; Business Media, 2010.

[24] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on information theory*, vol. 23, no. 3, pp. 337–343, 1977.

[25] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

[26] M. D. Cao, T. I. Dix, L. Allison, and C. Mears, "A simple statistical algorithm for biological sequence compression," in *Data Compression Conference, 2007. DCC'07*. IEEE, 2007, pp. 43–52.

[27] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[28] M. Effros, K. Visweswariah, S. R. Kulkarni, and S. Verdú, "Universal lossless source coding with the burrows wheeler transform," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1061–1081, 2002.

[29] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," 1994.

[30] T. Tao, "Single letter codes for nucleotides," *NCBI Learning Center. National Center for Biotechnology Information. Retrieved*, pp. 03–15, 2012.

[31] T. D. Blacker, "Fastq users manual version 1.2," *Sandia National Laboratories, SAND88-1326*, 1988.

[32] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, "Near-optimal sparse fourier representations via sampling," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 152–161.

[33] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[34] M. Rudelson and R. Vershynin, "Sparse reconstruction by convex relaxation: Fourier and gaussian measurements," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 207–212.

[35] H. S. Shapiro and R. A. Silverman, "Alias-free sampling of random noise," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 225–248, 1960.

[36] F. J. Beutler and O. A. Leneman, "The spectral analysis of impulse processes," *Information and Control*, vol. 12, no. 3, pp. 236–258, 1968.

[37] O. A. Leneman and F. J. Beutler, "The theory of stationary point processes," *Acta Mathematica*, vol. 116, no. 1, pp. 159–190, 1966.

[38] F. J. Beutler and O. A. Leneman, "Random sampling of random processes: Stationary point processes," *Information and Control*, vol. 9, no. 4, pp. 325–346, 1966.

[39] O. A. Leneman, "Random sampling of random processes: Impulse processes," *Information and Control*, vol. 9, no. 4, pp. 347–363, 1966.