INTERNATIONAL JOURNAL OF BIOLOGY AND BIOMEDICAL ENGINEERING
DOI: 10.46300/91011.2021.15.14 Volume 15, 2021

Security Provisioning and Compression of Diverse Genomic Data based on Advanced Encryption Standard (AES) Algorithm

Raveendra Gudodagi
School of ECE, REVA University, Rukmini Knowledge Park
Yelahanka Bengaluru, 560064
India

R. Venkata Siva Reddy (SMIEEE)
School of ECE, REVA University, Rukmini Knowledge Park
Yelahanka Bengaluru, 560064
India

Received: November 9, 2020. Revised: April 20, 2021. Accepted: May 6, 2021. Published: May 14, 2021.

Abstract—Compression of genomic data has gained enormous momentum in recent years because of advances in technology, exponentially growing health concerns, and government funding for research. Such advances have driven us to personalize public health and medical care. These pose a considerable challenge for ubiquitous computing in data storage. One of the main issues faced by genomic laboratories is the 'cost of storage' due to the large data file of the human genome (ranging from 30 GB to 200 GB). Data preservation is a set of actions meant to protect data from unauthorized access or changes. There are several methods used to protect data, and encryption is one of them. Protecting genomic data is a critical concern in genomics as it includes personal data. We suggest a secure encryption and decryption technique for diverse genomic data (FASTA / FASTQ format) in this article. Since we know the sequenced data is massive in bulk, the raw sequenced file is broken into sections and compressed. The Advanced Encryption Standard (AES) algorithm is used for encryption, and the Galois / Counter Mode (GCM) algorithm, is used to decode the encrypted data. This approach reduces the amount of storage space used for the data disc while preserving the data. This condition necessitates the use of a modern data compression strategy. That not only reduces storage but also improves process efficiency by using a k-th order Markov chain. In this regard, no efforts have been made to address this problem separately, from both the hardware and software realms. In this analysis, we support the need for a tailor-made hardware and software ecosystem that will take full advantage of the current stand-alone solutions. The paper discusses sequenced DNA, which may take the form of raw data obtained from sequencing. Inappropriate use of genomic data presents unique risks because it can be used to classify any individual; thus, the study focuses on the security provisioning and compression of diverse genomic data using the Advanced Encryption Standard (AES) Algorithm.

Keywords—Advanced Encryption Standard, Galois / Counter Mode, compression algorithms, dynamic Markov compression, fast and secure encryption, FASTA / FASTQ format.

I. Introduction

Whole-genome sequencing was traditionally used as a research tool but is now it is being used in clinics [1], [2]. Full genome sequence data can be a valuable tool for personalized medicine to guide clinical intervention in the future[3]. The SNP-level gene sequencing tool is used to classify functional variants from association studies and to boost the information available to evolutionary biologists, laying the groundwork for predicting disease susceptibility and drug response [4][5]. High-throughput sequencing techniques resulted in a drastic decrease in genome sequencing costs and an extremely fast collection of genomic

data [6]–[8]. "These technologies need ambitious efforts in genome sequencing, for example, the 1000 Genomes Project and the 1001 Genomes Project (Arabidopsis thaliana)"[9]. The storage and delivery of large quantities of genomic data have become a major concern and promote the creation of high-performance compression tools specifically designed for genomic data[10]. A recent increase in interest in developing new algorithms and techniques for the storage and handling of genomic data emphasizes the growing need for effective techniques for genomic data compression[11].

Many commercial instruments for extracting DNA from a variety of biological materials are available. The specificity of detection of the polymerase chain reaction (PCR) was found to be similar for different DNA kits[12]. Therefore, it is necessary to choose the right technique for the data. Considering that the genome sequence produces massive data, experts have estimated that by 2025 genomic sequencing will yield data of 40 exabytes per annum[13]. Challenges in the processing of genomic data are large archives, confidential data, holding personal information, and preserving the data. And we should expect genomic data to be stored in a stable storage tier that is always available. Compression of data plays a vital role in the management of genome data since it reduces storage space without losing any information[14]. Smaller files improve device throughput and are useful for bandwidth management in file transfer[15]. While many big data companies have been behind in solving this issue of data crisis for the past few years, the problem remains unresolved[16].

II. Literature Review:

The data on which security services should be provided is made as input to a cryptographic algorithm. The output is the "protected data". "Most crypto algorithms often require an input parameter to be used as a key. The key affects the output of cryptographic algorithm because only the person with the same key can recover the original input or to generate the same output from the same input"[14][1].

"One may differentiate between the following key types based on the type of key used: 1. The symmetric key cryptography–refers to the class of cryptographic algorithms in which the encryption key and decryption key can be easily determined from one another or are (in most cases) identical. 2. Public (or asymmetric) key cryptography–a class of cryptographic algorithms in which encryption and decryption are performed with separate keys. In this method, private key, public key pair: whatever is encrypted with the private key can be decrypted with the public key, and vice versa. The private and public keys are interconnected in the sense that the public key can be easily extracted from the private key, while the private key is almost impossible to extract from the public one. Each participant must keep his private key private, while the public key must be made public. Cypher is concerned primarily with symmetric key encryption. A cypher is an algorithm which encrypts plaintext to ciphertext (encryption) and decrypt ciphertext to the original plaintext (decryption), assuming it has the same secret key with the ends of encryption and decryption"[50].

"There are two cypher classes: blocking and streaming cypher. A block cypher transforms one block of plaintext (P) into one block of ciphertext (C) of the same size by applying the same transformation to any block of input data and using the same key. Encryption and decryption can be specified as CI=E(Pi) and Pi=D(Ci), where E and D are the encoded and decrypted functions, respectively, and I is the block index. Two of the most widely used block cyphers are the Advanced Encryption Standard (AES 2001) and the Data Encryption Standard (DES 1999). The Advanced Encryption Standard receives the most attention in this article (AES). Stream cyphers generate the ciphertext stream by bit-XORing the plaintext stream with a keystream of the same length. Additive stream cyphers are an example of this type of cypher. Ci = Pi Ki, where Ci is the ith bit of the ciphertext, Pi is the ith bit of the plaintext, and Ki is the ith bit of the keystream"[51].

The Advanced Encryption Standard (AES), also known as Rijndael, is a specification for the encryption of electronic data established by the National Institute of Standards and Technology (NIST) in the United States in 2001. Rijndael is a cypher family of various key and block sizes. NIST chose three Rijndael family members for AES, each with a block size of 128 bits but three different key lengths: 128, 192, and 256 bits. Table 1 gives the usage of Advanced Encryption Standard (AES) algorithm. It explains the advantages and capabilities of AES in different data and scinarios.

Table1: How security is provided to data using AES

01	General (Security provisioning with AES algorithm)	
	1	AES-128 offers a sufficiently large number of possible keys, making an exhaustive search
	2	Using AES in second layer gives extra security to data.
02	Security provisioning on distributed networks (IoT Cloud)	
	3	It provides strong security from the attackers.
	4	Larger size of data increases the data delay time for encrypting data.
03	Security provisioning (Genomic data)	
	5	Allows 128 parties across 5 continents to perform an AES computation in under 3 minutes and is the first to examine garbled circuits at such a large scale
	6	AES gives more efficiency, secrecy, integrity and avoids reply attacks.

"Recent work has seen DNA as an important medium for long-term and ultra-compact storage of information, as well as a stego-medium for secret messages[3]. Artificial components of DNA can be added to the genome of living organisms with encoded information, such as common bacteria"[4]. Use the "genetic code degeneration and, in particular, the silent mutations, produces coding that does not change the properties of the inserted gene or the characteristics of the host genome (very critical conditions when dealing with the living organisms). Memorizing the key information and generating the hidden message in the form of a physical

polypeptide provide additional security for data transfer while the coding protocol is being implemented"[5], [6].

FASTQ is a text-based format created to store both a biological sequence and the subsequent feature rankings. For brevity both the series letter and the ranking value are encoded with one ASCII character. It was originally created at the "Welcome Trust Sanger Institute" to package a FASTA-formatted sequence and its quality data but has recently become the de facto standard for storing the performance of high-throughput sequencing instruments such as the "Illumina Genome Analyzer"[7]. The SAM Format is a text format used in a series of ASCII columns delimited by tab to store the sequence data. "Currently most SAM format data is output from aligners that read FASTQ files and assign sequences to a position relative to a known reference genome. In addition, SAM can also be used to store unaligned sequence data directly generated from sequencing machines. VCF is a text file format (most likely to be stored in a compressed way)"[22],[23]. It contains 'meta-info lines, the header line, and then the data lines each containing information about the location of the genome. The format also has the capability of storing sample genotype information for each position"[8]. Table 2 gives description of common file formats corresponding to different data types. Currently all these are managed either natively or by Pysam (BAM files), Bio-python (FASTA), or two bit reader (2bit). Additionally , it is possible to index BED, GTF2, GFF3 and PSL files with a tabix. Supports reading of tabix-compressed files also (via pysam). In this paper we have implemented using Pysam.

Table 2. Data types with corresponding formats

Data type	***Unindexed formats***	***Indexed formats***
Sequence	FASTA	2bit
Annotations	BED, GTF2, GFF3, PSL	BigBed
Quantitative data	bedGraph, wiggle	BigWig
Read alignments	bowtie, SAM, PSL	BAM

It is most often generated as a human-readable variant of its sister BAM file, containing the same data in a compact, indexed, binary form[9]. "The first biological polynucleotide sequence was described twelve years after the Watson and Crick double helix DNA structure was published in 1953"[10]. Even though the anticodon, the three nucleotides that combine to the "mRNA sequence, was not yet discovered in the sequence, it was the 77-nt yeast alanine tRNA with a recommended junction structure"[11]. Around that time , scientists could sequence just a few base pairs a year, not nearly enough to sequence an entire gene. "Previous works showed that, in ancient sequence evidence postmortem damage was artificially caused, spurious demographic patterns were reconstituted. This has underscored the necessity for unique data quality in nucleotide sequence analysis of skyline-plot"[12]. The coverage of multiple sequencing attained employing high-throughput sequencing techniques will slash the probability of errors and their existence[13].

Firstly, a reference genome suits the reads in the FASTQ file[14]. In short, for each read, the alignment phase infers the corresponding position in the series of references from which the reading was created (or that there is no such region). Besides the mapping position, the alignment often produces, if any, the missing information along with some additional fields. This alignment information is stored in the Sequence Alignment Map (SAM) /Binary Alignment Map (BAM) format along with the original reads and accuracy scores ("BAM is the binarized, compressed version of the SAM file"). These files are incredibly large, usually hundreds of gigabytes, and are extensively used for most downstream applications. "Cryfa, an industry-oriented platform for safe encryption of genomic data in Fasta / Fastq / VCF / SAM / BAM formats, as well as lightweight Fasta / Fastq formats"[15]. The security of these data is greatly improved by a simple, shuffling process. We further preserve genomic data security by failing to explore complexity in those files. "Cryfa therefore cannot be used for identification of animals. The tool is around one order of magnitude faster than the best state-of-the-art compression plus encryption tools, including those for general and limited use. Cryfa is not only high-speed and has a high security standard, but also has very limited memory consumption (just a few megabytes)". "In addition to their variant calls (compact and summarized form of the raw data), geneticists prefer to store aligned, raw genomic data of patients, mainly due to the immaturity of bioinformatic algorithms and sequencing platform. Thus, we propose a system to protect the privacy of aligned, raw genomic data. A program for the storage , retrieval and transmission of compatible, raw genomic data (i.e., SAM files) to preserve privacy. We are confident that the proposed scheme will accelerate genomic research, because participants in clinical trials will be more willing to consent to the sequencing of their genomes if their genomic privacy is maintained"[16][17].

"Genome-wide association studies (GWASs) aim to recognize a trait-related genetic variants and have been a powerful approach to understanding complex diseases. A critical challenge for GWASs has been the reliance on individual-level data which typically have strict privacy requirements, creating an urgent need for methods that preserve participants' individual-level privacy. They developed an initial Double-Chinese Remainder Theorem (CRT), or RNS, version. Our version is based on the same security assumptions as the original scheme, namely the problem of Ring Learning With Errors (RLWE), but relies on native 64-bit integer arithmetic as opposed to multi-precision integer arithmetic for better efficiency and parallelization"[18][19].

III. IMPLEMENTATION OF PROPOSED ALGORITHM:

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write "15 Gb/cm^2 (100 Gb/in^2)." An exception is when English units are used as identifiers in trade, such as "3½ in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

Figure 1 explains the steps involved in compressing the sequenced raw data, which is available in FASTA/FASTQ format and encrypting it with AES (Advanced Encryption Standard) key technique[23][34]. The raw data is made to split to have three segments namely, Headers, Bases and Quals. The headers stand for sequence names and comments that precede the sequences. Transformation packaging for an exemplifying collection of DNA bases for triplets[23][11].

Quals will have the remaining part of data. These three segments are compresses individually to have packs namely, PackH, PackB, and PackQ, for Headers, Bases and Quals respectively. After compressing these segments or packs are subjected to encryption [35]. In this stage the data is encrypted using key (normally termed as Cipher text) and fed to the next stage where we apply AES key encryption to safeguard the data in an efficient way. The algorithm for this will be explained in Implementation section. Finally, we will have the compressed and encrypted data as output of this stage.

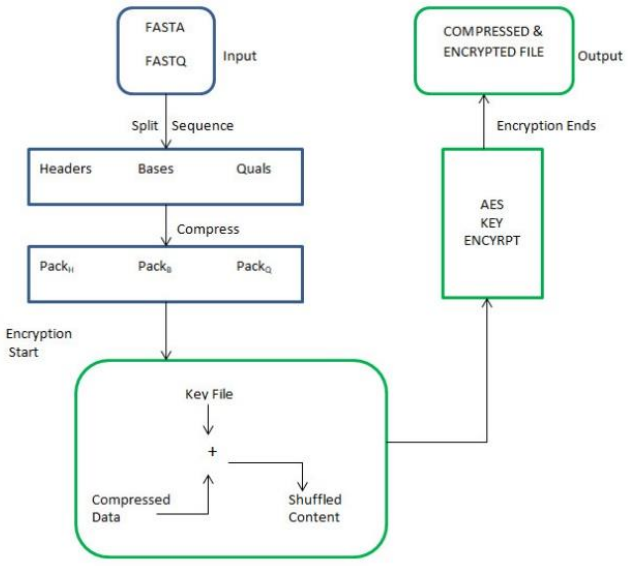

Fig. 1. Encrypting the sequenced raw data.

```
Algorithm 1: Encryption Process
Start Encryption
Input FASTA/FASTQ File
Input the password file
Start Compressing and Shuffling
  Compacting in < 1 second
  Shuffling in < 1 second
Obtain the encrypted file Output
Stop Encryption
```

The algorithm 1, gives the steps involved in encryption process. The input file is either FASTA or FASTQ file, which is encrypted using the key, which is done using inputting the Password.txt file. The file is compressed in the allotted time stamp of less than 1 sec and the file is compacted. The shuffling process is performed in the next allocated time stamp of less than 1 sec, and the final encrypted file is obtained as output result. The block diagrams are illustrated in figure 2 and figure 3, for encryption and decryption process[36], [37]. Here at first input fasta file . fa is multiplied with string 1 with Cartesian product vector resulting into the bytes input, and this process is repeated as chain of multiplications up to string 4 until a compressed data is obtained[38][39][40][41]. In mathematics, specifically set theory, the Cartesian product of two sets A and B, denoted A × B, is the set of all ordered pairs (a, b) where a is in A and b is in B. In terms of set-builder notation, that is.

$$A\ X\ B = \{(a, b) | a \in \text{A and b} \in \text{B}\}$$

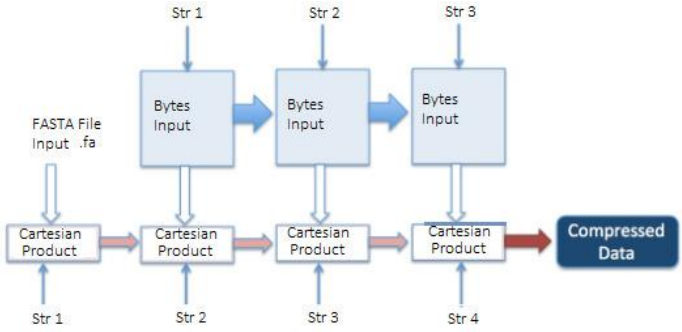

Fig. 2. Compression of input data using cartesian produc.

A table can be created by taking the Cartesian product of a set of rows and a set of columns[39], [42]. The life sciences are becoming "big data companies," and that is setting the standard for addressing that storage problem in the scientific community[43]. Scientists have over the past decade needed the storage space provided by genomic data, yet in the future brain data that is equivalent to world digital information would be difficult to manage[44], [45]. Therefore, there is a need for a modern, efficient approach that will resolve all the challenges of genomic data such as storage space, fast processing, and system throughput[15], [45].

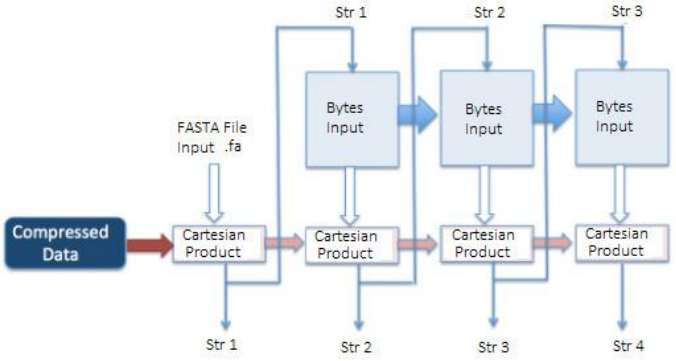

Fig. 3. Decompression of input data using cartesian product.

Figure 4 depicts the decryption of the encrypted data which is reverse process of encryption. The compressed and encrypted data is fed to AES GCM (Galois/Counter Mode) and in the subsequent stage the key is removed from the shuffled data [15]. Later it is unpacked to get the original data back at the output stage. In algorithm 2. a compressed format is given as input in the decryption process previously obtained encrypted file. The file is un-compressed in the allotted time stamp of less than 1 sec, and the de-compacting process is completed. The un-shuffling procedure is performed in the next allocated time stamp of less than 1 sec and the final decrypted file is obtained as the resulting output.

Genomic data produced by high-throughput sequencing (HTS) are generally stored as raw sequencing readings in the FASTQ format or as readings mapped to the SAM reference genome[46],[47]. Both formats have significant footprints on memory. The growth in HTS data worldwide has contributed to the development of advanced compression methods aimed at reducing the HTS data size substantially. Due to the enhancement of interest in genome sequencing, there are many developments in sequencing technologies, both in terms of performance and affordability. The algorithm 3 gives the steps which are followed in compressing the DNA data using dynamic Markov model[39], [48]. In the initial step the raw data, which is in FASTA form is fetched. A string is created for the sequenced data and compared with the original string.

```
Algorithm 2: For Decryption
Start Decryption
Input Compressed File
Input Encrypted Sequence
Start De-Compressing and Un-Shuffling
  Un-shuffling done in < 1 second
Decryption done in < 1 second
End Decryption
```

This comparison is done in order to find the unique characters present in the string. Predictive arithmetic coding is used in Dynamic Markov Compression, which is similar to prediction by partial matching except that the input is predicted one bit at a time (rather than one byte at a time). DMC, like PPM, has a decent compression ratio and moderate speed, but it takes more memory and is not commonly used. The compression algorithm is based on a priori data assumptions. The new approach here is to use an algorithmic approach to find a Markov chain model that describes the data. If the first part of the data can be used to construct it, it can be used to predict the next set of characters.

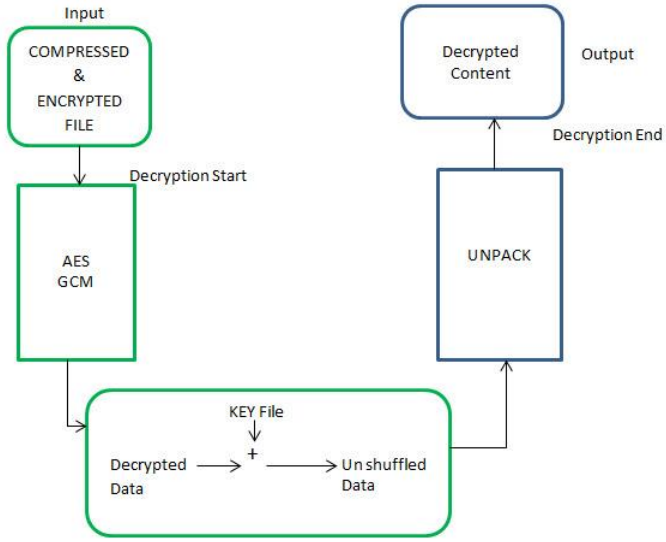

Fig. 4. Decrypting the data.

```
Algorithm 3:  Algorithm for DNA compression
using Dynamic Markov Compression

Start:
Input: FASTA file in a List;
Create single string of sequence data;
Get size of original string;
Obtain unique characters in the string;
While(Count occurrences of all the bases in
the sequence) do
{
compressing… bytes in %d, bytes out %d, ratio
%f;
comp() and de-comp() implement arithmetic
coding;
}
end while
If (Count > 1)
string has 10735 characters;
string takes up 10.53 MB in disk;
Count Number of Characters in string;
Obtain Histogram for base frequency;
else
Setup Transition matrix for k-th order markov
chain;
Creating labels of the heatmap using cartesian
product;
Calculate dynamic Markov compression;
end if

end
```

Such advances have made it possible for us to see whole-

genome sequencing as an invaluable tool for both precision medical care and public health. As a consequence, genomic data sets that are increasingly wide and widespread are being produced. This presents a major challenge to the preservation and dissemination of such information. It is now more difficult to retain genomic data for a decade than to collect the information in the first place.

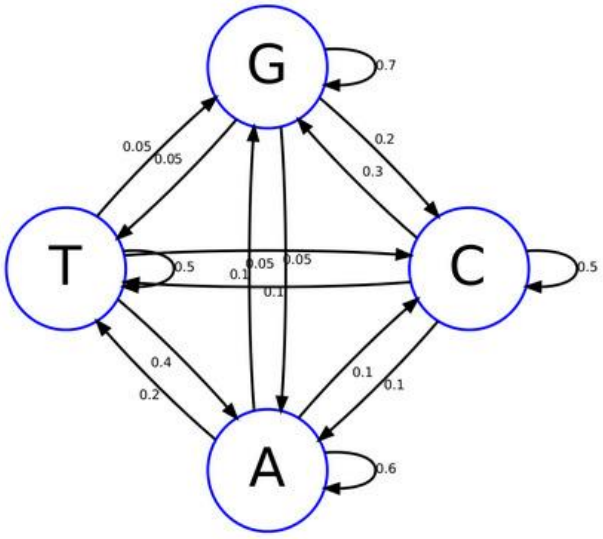

Fig. 5. Signal flow graphs of with probabilities nucleotide prediction/existence

This condition calls for effective genomic knowledge representations. Signal flow graphs shown in figure 5 shows probabilities of nucleotide prediction/existence. This will indicate the existence of nucleotide in the prediction. For example, if we consider A as reference then the predicted nucleotide with its probability of existence is shown in signal flow graph as A→A is 0.6, A→C is 0.1, A→G is 0.05, A→T is 0.2. Similarly, it is also shown for two letter nucleotide in the figure 5.

IV. Results and Discussions:

Algorithms 1 and 2 have the computational complexity of O(1) (horizontal computational complexity) as it deals with the constant terms such as compressing and shuffling. Algorithm 3 has the computational complexity of O(n) (Linear Computational Complexity) after several loopings and iterations. We tested the cost-effectiveness of operating on multi-core computing resources on two sample genomic datasets, each with a distinct number of threads. Running with eight distinct threads is 2.4 times faster than instantaneously than running with one thread, and it takes 1.4 times more CPU time as an average of user and device times. There is also an insignificant distinction between memory usages while running with one thread and eight threads that are 10MB. Our research method uses at most 31MB of RAM.

The figure 6 gives the frequency of bases, depending on this we can have the heat map for the nucleotides. To get the heat map of nucleotides we use the confusion matrix in the algorithm.

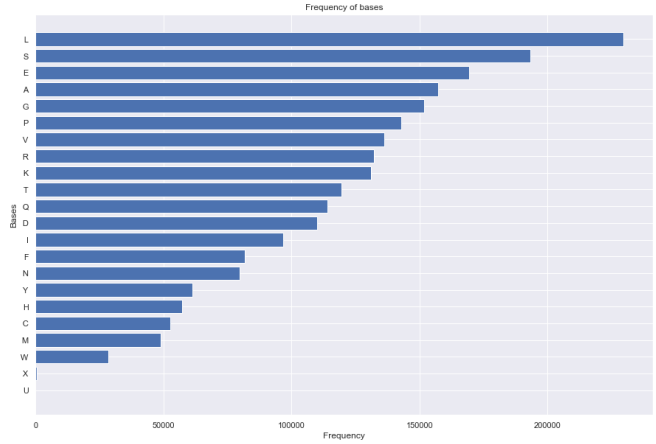

Fig. 6. Frequency of Bases

Different bases are being used and their frequency is plotted along X-axis. As we can see fron the frequency plot W is having lowest frequency and L is having highest. Similarly figure 7 depicts the frequency of nucleobases in the sequences.

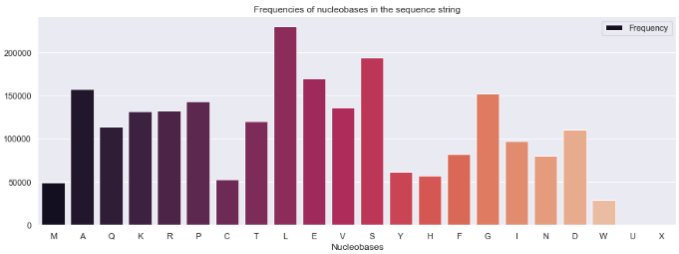

Fig. 7. Frequencies of Nucleobases in the sequences

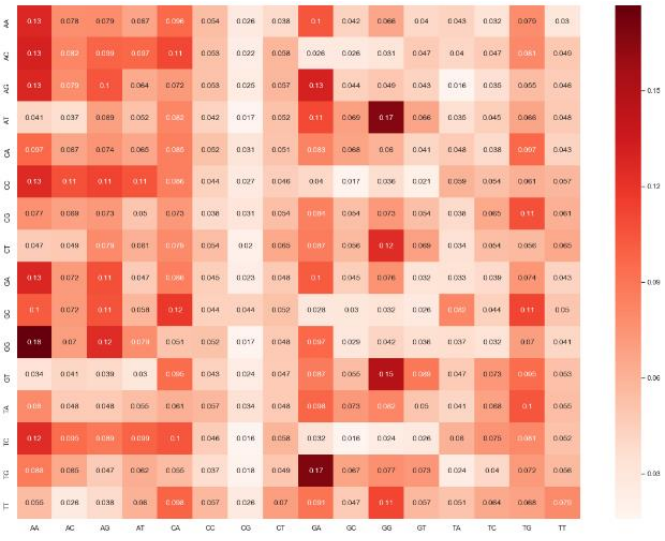

Fig. 8. Heatmap of Nucleobases in the sequences

Figure 8 gives the heat map of nucleotides with different color levels depending on their occurrences. Different 2 set

possible combinations of nucleobases are considered along X and Y axes. The figure 9 gives the confusion matrix used in evaluating the proposed model. In this confusion matrix the basic 4 nucleotides have been used.

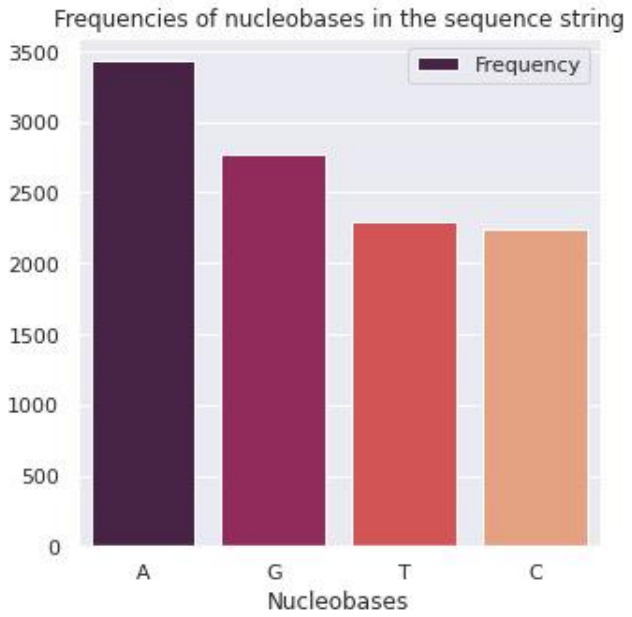

Fig. 9. Graphical Frequencies of Nucleobases in sequence Strings

Figure 10 shows the nucleotide FASTA sequence, nucleotides C-G along X-axis and A-T along Y-axis. Nucleobases and DNA data are shown in graphs represented in figure 11.

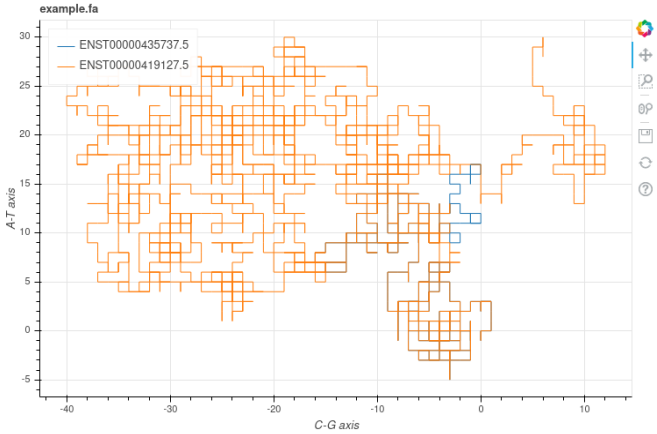

Fig. 10. Graph of example nucleotide FASTA sequence

Here at first the FASTQ file is encrypted with the corresponding password containing in Password.txt file. In the allocated time stamp of less than 1 sec the file is compressed, and the file compacting is done. In the next allocated time stamp of less than 1 sec the shuffling process is carried out and the final encrypted file is obtained as an output result. Further in the decryption process previously obtained encrypted file is provided as input in a compressed format. In the allocated time stamp of less than 1 sec the file is un-compressed, and the file de-compacting is done. In the next allocated time stamp of less than 1 sec the un-shuffling process is carried out and the final decrypted file is obtained as resulting output.

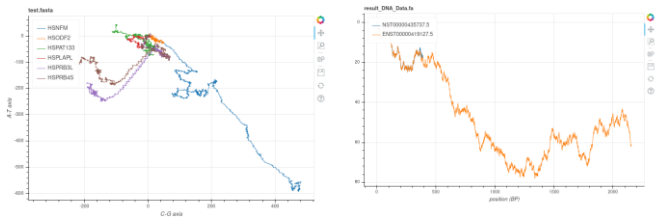

Fig. 11. Graph of nucleobase and DNA data FASTA sequence

V. Conclusion

A secure encryption and decryption for various genomic data in the FASTA/FASTQ format is proposed in this paper. Big chunks of raw sequenced data are broken into components and compressed. To encrypt the data, the Advanced Encryption Standard (AES) key is used, and the encrypted data is decrypted using the Galois/Counter Mode (GCM). The paper refers to sequenced DNA that can be in the form of raw data derived from sequencing. Inappropriate use of genomic data poses specific risks as it can be used to identify any individuals, hence the research addresses the security provisioning and compression of diverse genomic data based on the Advanced Encryption Standard (AES) Algorithm. This study also paves the way for the physical space needed to store the data to be minimized and the data can be preserved. This work focuses on developing methods in the areas of structure prediction, deciphering mechanistic understanding of interactions between various biological molecules and working with multi-scale systems.

Acknowledgment

Authors want to thank Dr. Mohammed Riyaz Ahmed, Mr. B. U. V. Prashanth and Dr. R. C. Biradar for their guidance.

References

[1] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] M. Hernaez, D. Pavlichin, T. Weissman, and I. Ochoa, Genomic Data Compression, vol. 2, no. 1. 2019.

[3] I. Ochoa, M. Hernaez, and T. Weissman, “Aligned genomic data compression via improved modeling,” J. Bioinform. Comput. Biol., vol. 12, no. 6, pp. 1–17, 2014, doi: 10.1142/S0219720014420025.

[4] I. Ochoa-Alvarez, “Genomic Data Compression and Processing: Theory, Models, Algorithms, and Experiments,” no. August, p. 153, 2016.

[5] Y. Liu and D. Wang, "Application of deep learning in genomic selection," pp. 2280–2280, 2017, doi: 10.1109/bibm.2017.8218025.

[6] R. Campos, M. Branco, S. Weiss, and N. Ferrand, "Patterns of hemoglobin polymorphism [α-globin (HBA) and β-globin (HBB)] across the contact zone of two distinct phylogeographic lineages of the European rabbit (Oryctolagus cuniculus)," Phylogeography South. Eur. Refug. Evol. Perspect. Orig. Conserv. Eur. Biodivers., pp. 237–255, 2007, doi: 10.1007/1-4020-4904-8_8.

[7] R. Wang et al., "DeepDNA: A hybrid convolutional and recurrent neural network for compressing human mitochondrial genomes," Proc. - 2018 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2018, pp. 270–274, 2019, doi: 10.1109/BIBM.2018.8621140.

[8] A. A. Hernandez-Lopez, J. Voges, C. Alberti, M. Mattavelli, and J. Ostermann, "Differential Gene Expression with Lossy Compression of Quality Scores in RNA-Seq Data," Data Compression Conf. Proc., vol. Part F1277, no. March 2016, p. 444, 2017, doi: 10.1109/DCC.2017.75.

[9] B. Lee, T. Moon, S. Yoon, and T. Weissman, "DudE-Seq: Fast, flexible, and robust denoising for targeted amplicon sequencing," PLoS One, vol. 12, no. 7, pp. 1–25, 2017, doi: 10.1371/journal.pone.0181463.

[10] S. Deorowicz and S. Grabowski, "Compression of DNA sequence reads in FASTQ format," Bioinformatics, vol. 27, no. 6, pp. 860–862, 2011, doi: 10.1093/bioinformatics/btr014.

[11] O. U. Nalbantoğlu and K. Sayood, "Compression of quality factors in next generation sequencing," Data Compression Conf. Proc., p. 419, 2014, doi: 10.1109/DCC.2014.46.

[12] J. Voges et al., "GABAC: An arithmetic coding solution for genomic data," Bioinformatics, vol. 36, no. 7, pp. 2275–2277, 2020, doi: 10.1093/bioinformatics/btz922.

[13] D. E. Sabath et al., "Characterization of Deletions of the HBA and HBB Loci by Array Comparative Genomic Hybridization," J. Mol. Diagnostics, vol. 18, no. 1, pp. 92–99, 2016, doi: 10.1016/j.jmoldx.2015.07.011.

[14] D. Greenfield, V. Wittorff, and M. Hultner, "The Importance of Data Compression in the Field of Genomics," IEEE Pulse, vol. 10, no. 2, pp. 20–23, 2019, doi: 10.1109/MPULS.2019.2899747.

[15] M. Hernaez, D. Pavlichin, T. Weissman, and I. Ochoa, "Genomic Data Compression," Annu. Rev. Biomed. Data Sci., vol. 2, no. 1, pp. 19–37, 2019, doi: 10.1146/annurev-biodatasci-072018-021229.

[16] M. Aledhari, M. Di Pierro, M. Hefeida, and F. Saeed, "A Deep Learning-Based Data Minimization Algorithm for Fast and Secure Transfer of Big Genomic Datasets," IEEE Trans. Big Data, vol. 7790, no. DECEMBER 2017, pp. 1–1, 2018, doi: 10.1109/tbdata.2018.2805687.

[17] M. S. Rao et al., "Novel Computational Approach to Predict Off-Target Interactions for Small Molecules," Front. Big Data, vol. 2, no. July, pp. 1–17, 2019, doi: 10.3389/fdata.2019.00025.

[18] S. Jiao and R. Goutte, "Code for encryption hiding data into genomic DNA of living organisms," Int. Conf. Signal Process. Proceedings, ICSP, pp. 2166–2169, 2008, doi: 10.1109/ICOSP.2008.4697576.

[19] A. A. Alonso and E. Balsa-canto, "A Normalisation Strategy to Optimally Design," vol. 2, no. Mci, 2017, doi: 10.1007/978-3-319-60816-7.

[20] A. Mu, "済無No Title No Title," J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004.

[21] D. P. C. C. L. E. Y. N. to K. in 20 Weeks, "済無No Title No Title," Dk, vol. 53, no. 9, pp. 1689–1699, 2015, doi: 10.1017/CBO9781107415324.004.

[22] S. Ambardar, R. Gupta, D. Trakroo, R. Lal, and J. Vakhlu, "High Throughput Sequencing: An Overview of Sequencing Chemistry," Indian J. Microbiol., vol. 56, no. 4, pp. 394–404, 2016, doi: 10.1007/s12088-016-0606-4.

[23] C. Kockan et al., "Sketching algorithms for genomic data analysis and querying in a secure enclave," Nat. Methods, vol. 17, no. 3, pp. 295–301, 2020, doi: 10.1038/s41592-020-0761-8.

[24] M. Hosseini, D. Pratas, and A. J. Pinho, "Cryfa: A secure encryption tool for genomic data," Bioinformatics, vol. 35, no. 1, pp. 146–148, 2019, doi: 10.1093/bioinformatics/bty645.

[25] T. Wang et al., "Gene Essentiality Profiling Reveals Gene Networks and Synthetic Lethal Interactions with Oncogenic Ras," Cell, vol. 168, no. 5, pp. 890-903.e15, 2017, doi: 10.1016/j.cell.2017.01.013.

[26] M. Vasinek and J. Platos, "LZ77 like lossy transformation of quality scores," Data Compression Conf. Proc., vol. 2018-March, no. 19, p. 429, 2018, doi: 10.1109/DCC.2018.00082.

[27] T. Ahmed, B. Johnson, C. Oppenheim, and C. Peck, "Highly cited old papers and the reasons why they continue to be cited. Part II. The 1953 Watson and Crick article on the structure of DNA," Scientometrics, vol. 61, no. 2, pp. 147–156, 2004, doi: 10.1023/B:SCIE.0000041645.60907.57.

[28] X. Kong, X. Dong, Y. Zhang, W. Shi, Z. Wang, and Z. Yu, "A novel rearrangement in the mitochondrial genome of tongue sole, Cynoglossus semilaevis: Control region translocation and a tRNA gene inversion," Genome, vol. 52, no. 12, pp. 975–984, 2009, doi: 10.1139/G09-069.

[29] S. Y. W. Ho and B. Shapiro, "Skyline-plot methods for estimating demographic history from nucleotide sequences," Mol. Ecol. Resour., vol. 11, no. 3, pp. 423–434, 2011, doi: 10.1111/j.1755-0998.2011.02988.x.

[30] C. Albert et al., "An introduction to MPEG-G, the new ISO standard for genomic information representation," bioRxiv, no. October, p. 426353, 2018, doi: 10.1101/426353.

[31] S. Chandak, K. Tatwawadi, and T. Weissman, "Compression of genomic sequencing reads via hash-based reordering: Algorithm and analysis," Bioinformatics, vol. 34, no. 4, pp. 558–567, 2018, doi: 10.1093/bioinformatics/btx639.

[32] C. Ting, R. Gooding, R. Field, and J. Caswell, "Reordering genomic sequences for enhanced classification via compression analytics," Proc. - 18th

IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019, pp. 252–258, 2019, doi: 10.1109/ICMLA.2019.00047.
[33] Y. Liu, X. Zheng, and C. Rong, "Machine learning based LncRNA function prediction," Proc. - 2017 Int. Conf. Green Informatics, ICGI 2017, pp. 67–70, 2017, doi: 10.1109/ICGI.2017.16.
[34] Ruslan Skuratovskii, Volodymyr Osadch, Yevgen Osadchyy, The Timer Incremental Compression of Data and Information, WSEAS Transactions on Mathematics, ISSN / E-ISSN: 1109-2769 / 2224-2880, Volume 19, 2020, Art. #41, pp. 398-406.
[35] Z. Huang et al., "A privacy-preserving solution for compressed storage and selective retrieval of genomic data," Genome Res., vol. 26, no. 12, pp. 1687–1696, 2016, doi: 10.1101/gr.206870.116.
[36] M. Blatt, A. Gusev, Y. Polyakov, and S. Goldwasser, "Secure large-scale genome-wide association studies using homomorphic encryption," Proc. Natl. Acad. Sci. U. S. A., vol. 117, no. 21, pp. 1–6, 2020, doi: 10.1073/pnas.1918257117.
[37] J. S. Sousa et al., "Efficient and secure outsourcing of genomic data storage," BMC Med. Genomics, vol. 10, no. Suppl 2, 2017, doi: 10.1186/s12920-017-0275-0.
[38] Z. Huang, E. Ayday, J. Fellay, J. P. Hubaux, and A. Juels, "GenoGuard: Protecting genomic data against brute-force attacks," Proc. - IEEE Symp. Secur. Priv., vol. 2015-July, pp. 447–462, 2015, doi: 10.1109/SP.2015.34.
[39] H. Yao, Y. Ji, K. Li, S. Liu, J. He, and R. Wang, "HRCM: An Efficient Hybrid Referential Compression Method for Genomic Big Data," Biomed Res. Int., vol. 2019, 2019, doi: 10.1155/2019/3108950.
[40] L. Mertzanis, A. Panotonoulou, M. Skoularidou, and I. Kontoyiannis, "Deep Tree Models for 'Big' Biological Data," IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC, vol. 2018-June, pp. 0–4, 2018, doi: 10.1109/SPAWC.2018.8445994.
[41] J. Li, X. Lan, Y. Liu, L. Wang, and N. Zheng, "Compressing Unknown Images with Product Quantizer for Efficient Zero-Shot Classification National Engineering Laboratory for Visual Information," Cvpr, pp. 1–10, 2019.
[42] E. Ernst, "Dynamic inheritance and static analysis can be reconciled," Nord. Work. Program. Environ. …, no. December 1998, 1998, [Online]. Available: http://forskningsbasen.deff.dk/Share.external?sp=Sd6b7117a-ac12-4d0c-b72b-e5ef0afb6a77&sp=Sau.
[43] K. K. Kaipa, K. Lee, T. Ahn, and R. Narayanan, "System for random access DNA sequence compression," 2010 IEEE Int. Conf. Bioinforma. Biomed. Work. BIBMW 2010, pp. 853–854, 2010, doi: 10.1109/BIBMW.2010.5703942.
[44] H. M. Waidyasooriya, D. Ono, M. Hariyama, and M. Kameyama, "Efficient data transfer scheme using word-pair-encoding-based compression for large-scale text-data processing," IEEE Asia-Pacific Conf. Circuits Syst. Proceedings, APCCAS, vol. 2015-Febru, no. February, pp. 639–642, 2015, doi: 10.1109/APCCAS.2014.7032862.
[45] C. L. Biji and A. S. Nair, "Benchmark Dataset for Whole Genome Sequence Compression," IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 14, no. 6, pp. 1228–1236, 2017, doi: 10.1109/TCBB.2016.2568186.
[46] S. Al Yami and C. H. Huang, "LFasTQC: A lossless non-reference-based FASTQ compressor," PLoS One, vol. 14, no. 11, pp. 1–10, 2019, doi: 10.1371/journal.pone.0224806.
[47] A. S. Keerthy and S. M. Priya, "Genomic Sequence Data Compression using Lempel-Ziv-Welch Algorithm with Indexed Multiple Dictionary," Int. J. Eng. Adv. Technol., vol. 9, no. 2, pp. 541–547, 2019, doi: 10.35940/ijeat.b3278.129219.
[48] I. Numanagić et al., "Comparison of high-throughput sequencing data compression tools," Nat. Methods, vol. 13, no. 12, pp. 1005–1008, 2016, doi: 10.1038/nmeth.4037.
[49] D. Pratas and A. J. Pinho, "EXPLORING DEEP MARKOV MODELS IN GENOMIC DATA COMPRESSION USING SEQUENCE PRE-ANALYSIS Diogo Pratas and Armando J . Pinho Signal Processing Lab , DETI / IEETA University of Aveiro , 3810 – 193 Aveiro , Portugal," 2014 22nd Eur. Signal Process. Conf., pp. 2395–2399.
[50] A. Asvadishirehjini, M. Kantarcioglu and B. Malin, "A Framework for Privacy-Preserving Genomic Data Analysis Using Trusted Execution Environments," 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2020, pp. 138-147, doi: 10.1109/TPS-ISA50397.2020.00028.
[51] R. Skuratovskii, Y. Osadchyy and V. Osadchyy, "The Timer Compression of Data and Information," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 455-459, doi: 10.1109/DSMP47368.2020.9204126.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Author Contributions: Please, indicate the role and the contribution of each author:

Example

Raveendra Gudodagi carried out the simulation and the optimization and implemented all the 3 Algorithms in Python.

R. Venkata Siva Reddy organized and executed the experiments of Section 4 and he was responsible for the Simulation and Statistics.