A Novel Approach to Compress and Secure Human Genome Sequence

Garima Mathur[1], Anjana Pandey[2], Sachin Goyal[3]
Department of Computer Science and Engineering, UIT, RGPV, Bhopal[1]
Department of Information Technology, UIT, RGPV, Bhopal[2 3]
Corresponding author: Garima Mathur, Email: garima41mathur@gmail.com

DNA sequences can be considered as a pool of genetic information mostly used for reproduction, classification, and detection of disease. FASTA is the commonly used DNA sequence in a textual format whose size is too large that makes it difficult to store and manage; also securing this data is a big issue. Compression techniques that can reduce the size of these DNA data files are the most appropriate solution, reducing the size also reduces the need for resources for transmission. Therefore, this work proposes a novel ASCII-based compression algorithm, in which DNA characters are first converted into ASCII integers and then delta computed, afterwards the LZW compression technique is applied to the computed result. For ensuring the security of data, the blockchain-based framework is used after the compression module to make data immutable. In this paper, for methods like LZW and Huffman code, compression ratio comparisons were also determined for homosapiens, and from the results, it is clear that the proposed algorithm shows a good compression ratio for some randomly selected data sets. Another aim of this paper is to show the benefits of using a blockchain-based framework in securing healthcare data.

Keywords: Genome sequence, Compression, FASTA, Blockchain, Delta computation, Genbank.

1 Introduction

DNA plays an important role in healthcare because it carries the genomic information related to organisms, which can be used in various researches to identify or treat the disease. Genome sequencing cost has been fundamentally decreased after a greater advance in techniques used for DNA sequence reading. GenBank, which has the largest collection of DNA sequences, is also growing exponentially. Management of this vast amount of data is another major issue involving both secure transfers as well as storage of DNA data [1][2].

Compressing these files is the most appropriate solution which not only reduces size but reduces the need for resources for transmission and that means it reduces transmission time as well as space[3][4][5]. Security of this data is another big issue, blockchain is considered as one of the best solutions as it has already proved its immutability property in the field of crypto-currencies. It can be imagined as a link list of records also known as blocks where each block contains a hash value of its previous block that makes it difficult or impossible to manipulate. Therefore, this work proposes an ASCII-based delta computation of FASTA, a compression algorithm for the management and storage of NCBI data. This resultant compressed data will now get secured using blockchain technology.

2 Background

2.1. DNA data composition

As we know biological information about an organism/human is stored in its DNA, so the study of DNA sequence is an important factor for understanding organisms. The basic structure of DNA. is shown in figure 1[6].

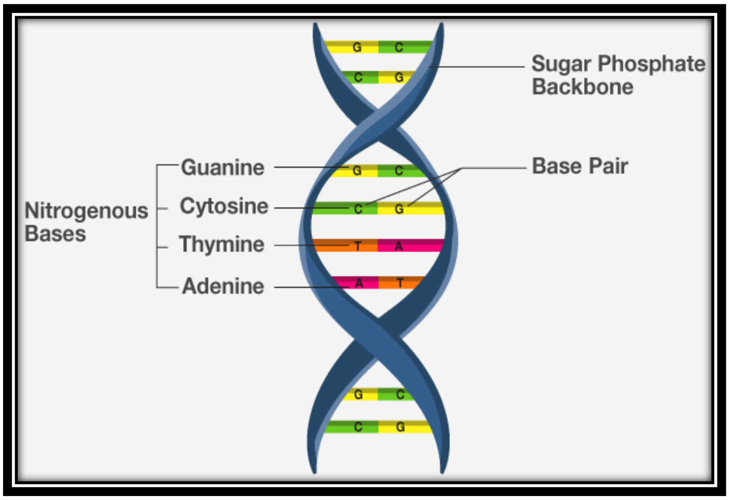


Figure 1. The basic structure of DNA.

The FASTA file format is a text-based format used to denote DNA sequences in bioinformatics [7][8], consisting of base pairs using the single-letter code as shown in table 1-

Table 1. DNA base pairs

A	Adenosine
C	Cytoline
G	Guanine
T	Thymidine
N	Can be any of these (A,C,G,T)

The layout of organisms depends upon the order in which they are stored. The format allows sequences of preceding sequence names and comments. FASTA file format sequence starts with a one-line description of identifier along with DNA data sequence.

```
>AR147821.1 Sequence 1 from patent US 6225051
GGCATCTGAGACCAGTGAGAA
```

The details about the identifier are separated from the sequence of data by the '>' sign. The identifier of the sequence is the word after the">" symbol and the remaining part represents description which is an optional part that separates the identifier from a white space or tab. The data sequence will start from the next line after the text line and the second line begins with another ">" sign indicating the ending of the sequence and also the beginning of another sequence.

2.2. Blockchain technology

The term blockchain refers to the linked list of records that are secured using the traditional secure hash algorithm with a fixed key of 256; the hash value of each block is stored in the next block along with the data of that block [9, 10]. Every time a new block is created is appended at the end of the generated linked list and every new block will point towards its previous block which is secured using a secure hash [11][12]. These blocks are distributed and stored in a peer-to-peer manner in the blockchain. For every block, there are two 2 keys namely private and public keys, one used for encryption and the other for decryption. The correct private key can only decrypt the message by which the message has been encrypted. This will help in achieving immutability and consistent behavior [14]. This is known as asymmetric cryptography.

2.3. Data compression techniques

Various data compression methods depend on the criteria, whether it is eliminating data's part or after decomposition data cannot be recovered to its original form [15].
They are classified into two-

- **Lossy Data Compression**

The lossy data compression algorithm is used where there is no need to keep data the same as the original because this algorithm eliminates some data's part after decomposition[16]. Some data compression algorithms are MPEG, MP3, and JPEG. However, this type of algorithm has the disadvantage that because of repeated compression and decompression the data quality degrades.

- **Lossless data compression.**

Various algorithms that compress data losslessly have been proposed and are now used widely. When we want the original data and decompressed data to be consistent or identical then this data compression algorithm is used that allows recovery of original data from compressed one[16]. One of the most popular lossless data compression algorithms is a ZIP file format. Some lossless data compression techniques are Run Length Encoding, Huffman Coding, Dictionary Based Encoding, and Arithmetic Encoding.

3 Current Lossless Data Compression Techniques

3.1. Huffman Coding

Huffman coding is a top-down method in which data compression is done through ASCII value to obtain optimal results [18]. Here each character is replaced by a binary number according to its occurrence in a text file. Frequently appearing characters have shorter length codes whereas rarely appearing characters have lengthy codes [19]. It is obtained by constructing a binary tree where each leaf node represents the character to be encoded. Each node has a probability of occurrence of character with it. Each edge will be labeled either with 0 or 1. Step by step procedure of the algorithm is as follows-

- Identify each character's occurrence and count them
- Find its probability using character count
- Sort according to probability with most probable first
- Create leaf node of each character and add them in a queue
- Two least frequency characters will be grouped to form a new combined frequency that will lead to a binary tree-like structure
- Step 5 is repeated until only one parent is left for all nodes known as a root node
- Each parent's left child edge will be labeled as digit 0 and right as digit 1. Tracing the tree will give Huffman code where greater frequency characters have shorter code.

3.2. Lempel-Ziv-Welch (LZW) Algorithm

Lempel-Ziv-Welch (LZW) algorithm [20,21] is a dictionary-based compression method in which string patterns that are repeating constantly are saved in a table like a dictionary and indexed value is used to represent them. During the compression process in place of using large strings, their short index values are used and there is no need to send it along with a coded message during the decompression process. This algorithm is also known as adaptive compression because during decompression same dictionary is created dynamically instead of sending the previously generated dictionary.

The compression technique used for FASTA file format DNA sequence is designed especially for English alphabet –A, C, G, T. Lempel-Ziv plus Huffman coding is used for such type of compression where Huffman coding is used to converts FASTA format DNA into respective code.

3.3. Algorithm for bit reduction

The primary thought with this method is to modify the popular seven digits encoding into a 5-bit application-specific encoding framework and afterward pack it in a byte array [17]. This algorithm lessens the size of a long string; it is used when the content of the string doesn't influence the compression ratio. Step by step procedure of the bit reduction algorithm is as follows-

- Pick some rapidly appearing characters that have to be encoded and convert them to their respective ASCII value
- Convert this ASCII value to its respective binary code (number).
- Place these numbers into an 8-bit array or we can say an array of bytes
- Eliminate unwanted bits or extra bits from the front.
- Again arrange and maintain this array
- This text be will be then encoded and compressed.
- At the client's end decompression will be done following the reverse steps.

3.4. Vertical DNA Sequences Compression Algorithm Based on Hexadecimal Representation

The vertical DNA sequence compression algorithm represents the nucleotides in binary format and packs them into two bits. These bits are then converted to a hexadecimal representation to reduce the size of the sequence, where each character codes two nucleotides. At last, this algorithm is utilized to recognize areas of comparability between a few DNA groupings [22]. For an illustration of an algorithm, we will use the following sequence as an example "AGGT TACG GACT TAAG AGTT CACG AAGT ATGT TAGG CGTT "

- **Encoding Phase**

In this phase, the DNA sequence is converted into binary digits. The nucleotide A is coded as 00, C as 01, G as 10, and T as 11. The result of encoding our example will be as follows:

00101011 11000110 10000111 11000010 00101111
01000110 00001011 00111011 11001010 01101111

In the next step, the binary numbers are converted into hexadecimal numbers.

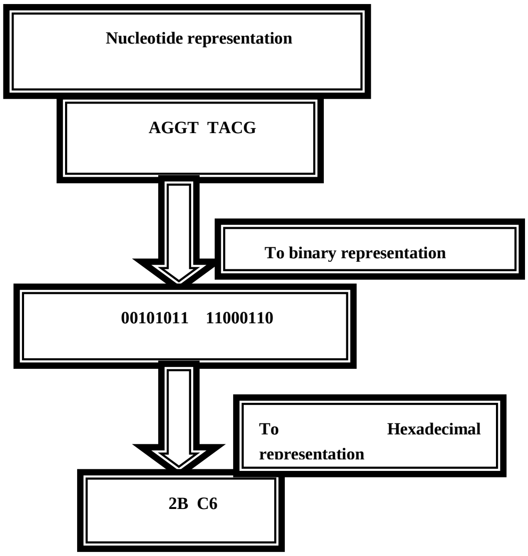


Figure 2(a). Conversion to Hexadecimal representation

Converted hexadecimal representation is as follows:

00101011 11000110 10000111 11000010 00101111
01000110 00001011 00111011 11001010 01101111
2B C6 87 C2 2F 46 0B 3B CA 6F

- **Decoding Phase**

The decoding phase is the inverse of the encoding phase.

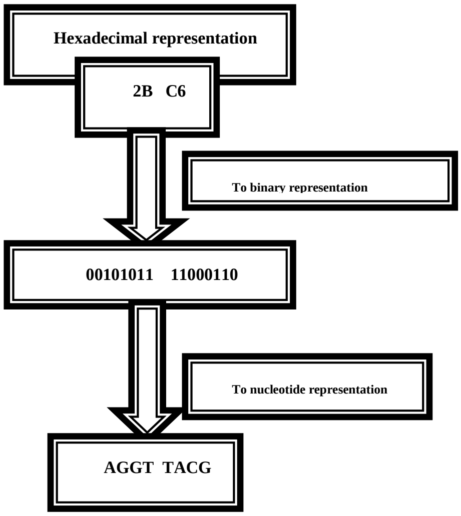


Figure 2(b). Conversion to nucleotide representation

4 Shortcomings of Previous Algorithm

For existing algorithms of text compression, it is a very tough task to compress complicated genome sequences as they were designed for simple textual data only. DNA sequences have only four symbols (base) A, C, G, T, and N (undefined base) in its FASTA format[23], so the regularity in this type of data are very rare and small and only 2 bits can be used for each base. Let us see by taking an example.

Genbank number- LY844744.1
Data Set - KR 1020200033836-A/4: Therapeutics of blood cancer.
FASTA format – NNGGTGGTGGTTGTGGTGGTGG

Traditional textual data compression algorithms such as LZW (Lempel-Ziv-Welch) and Huffman algorithm are not suitable for compressing DNA sequences because using the above data set, Huffman's code performs very badly as the probability of occurrence of these symbols is almost the same. The average compression ratio received by the LZW algorithm is also very low for the above data set as there are 44 bits in the original text and after compression, there are 42 bits, and 6 zero bits are added to it. On the other hand, the Vertical DNA sequences compression algorithm has a good compression ratio but has poor compression and decompression speed.

5 Proposed Model

To overcome the shortcomings of previously discussed algorithms and to get a good compression ratio, this work proposes a new delta computation-based algorithm. The proposed model is depicted in figure 3.

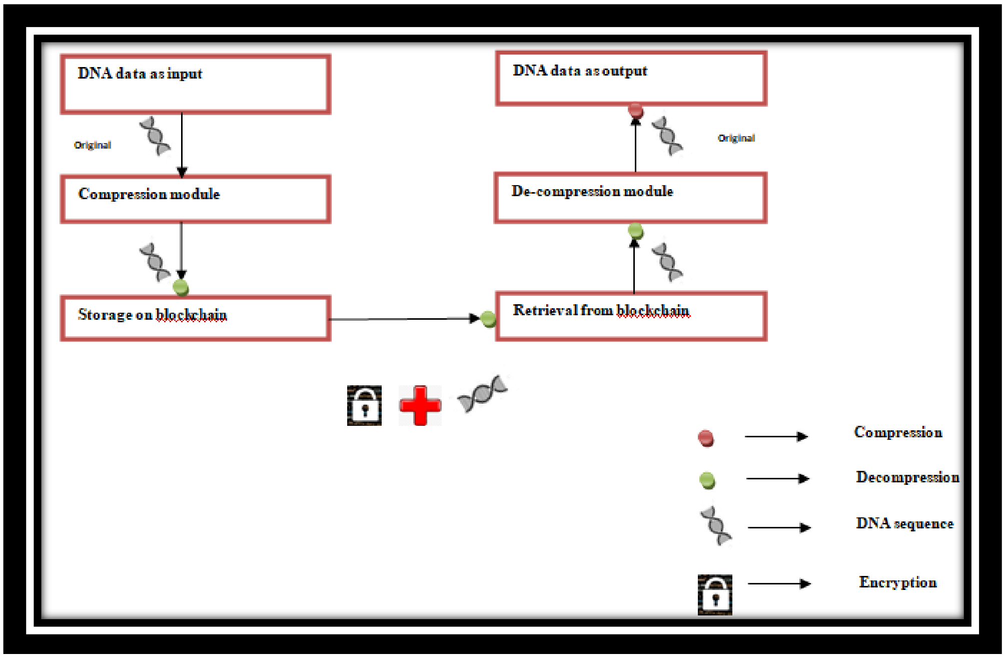


Figure 3. Proposed Model

A detailed description of the compression and decompression module is shown in Figures 4(a) and 4(b). For compression phase/module DNA sequence in FASTA format is given as an input. As we have already discussed DNA FASTA format files are in textual form. DNA characters are first converted into ASCII integers and then delta computation is done on them. Delta computation can be defined as the previous and next values difference, which means only the difference will remain at last. LZW compression technique will be applied to the computed result. The output of this stage is compressed DNA.

Each step of the compression module in reverse form is done at the decompression module. DNA LZW decompressor is used to decode compressed DNA data given as input. After this delta value is reverted, all data will come in its original form. Finally, we will get the decompressed data as an output.

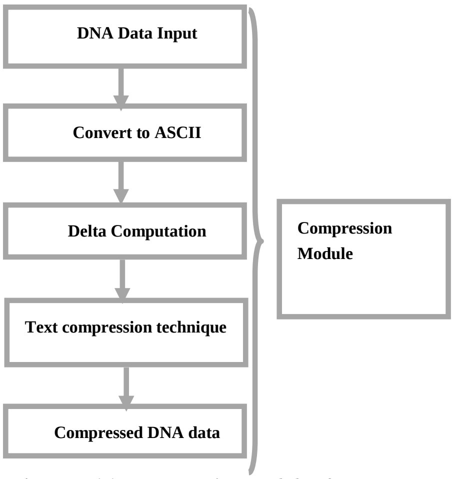


Figure 4(a). Compression module of DNA data set.

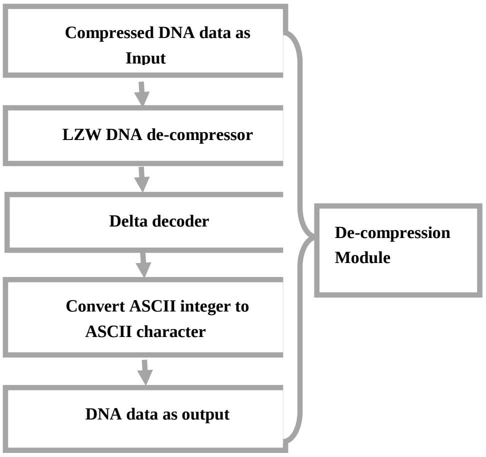


Figure 4(b). De-Compression module of DNA data set.

Once the size of the DNA sequence is reduced it can be stored in the blockchain to increase the security of data, as a primary goal of blockchain is to make data immutable. The property of the hash function will not allow any type of change in an nth block as it directly gets reflected in the n-1th block. This process is similar for all consecutive blocks and will make the chain completely different. This unchangeable feature of blockchain is termed immutability. This process can be better understood with the following example of SHA256 [24].

Table 2. Hashing Process

Input	Hash value
Hello	185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969
Bye	128901223aac8df3b89cd75d7ec644f9924ed9dcd01e0c65ae99334a3cf9273a

The example shown above proves that the size of information doesn't matter every information has its equivalent 256-bit value. However, it becomes more complicated as the size of information increases. Remembering the hash value is simpler than remembering the original data. A change in original data can make a big difference in output hash value, this property of cryptographic hash is known as the avalanche effect.

Table 3. Avalanche's effect

Input	Hash value
Hello	185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969
hello.	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

The above table shows that a small change can make a big difference and it can easily detect any attack or changes in original data. Adding a blockchain-based framework at the end of the proposed compression algorithm will increase the security of stored DNA sequences and make data immutable.

6 Result and Analysis

For analysis of the result, several parameters are used, depending on the nature of the application. For any compression algorithm, the main criteria while measuring performance is space efficiency. Since time efficiency is also one of the important factors of compression behavior which entirely depends on the redundancy of symbols of the source file that makes performance measurement of compression algorithm more difficult. Two parameters used for measuring the performance of any compression algorithm are Compression Ratio and Saving Percentage [25].

6.1. Compression ratio

The term compression ratio can be calculated as a ratio of the compressed file to the uncompressed file.
Compression ratio (CR) =(UC/C)
Where C= Compressed file
UC= Un-compressed file

6.2. Saving Percentage

It calculates the percentage of source file shrinkage.
Saving Percentage =((UC - C) / UC) * 100%
Where, C= Compressed file
UC= Un-compressed file

The below table shows the compression ratio of LZW, Huffman, and our proposed algorithm on three different data sets from NCBI Genbank.

Table 4. Compression ratio of various text algorithm

Data Set	Input text size(in bits)	LZW CR	Huffman CR	Proposed algorithm CR
LY844744.1	44	75.4%	55%	20.2%
S56177.1	609	77%	59.1%	32.3%
EU275349.1	972	65.6%	57%	29%

From table 4 it is observed that the average compression ratio of LZW, Huffman, and Delta computation-based compression algorithm is CRL= ±72.6, CRH= ±57.03, and CRD= ±27.16. Figure 5 shows the graph of three popular text compression algorithms on some random data set collected from NCBI Genbank.

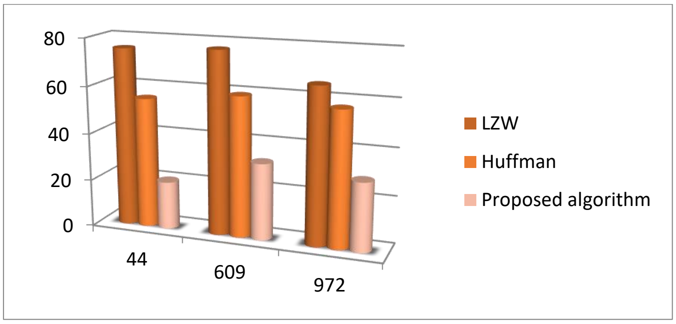


Figure 5. Compression ratio comparison graph for random dataset

Saving percentage calculates the percentage of source file shrinkage. The algorithm with a high compression ratio has a low saving percentage as shown in table 5.

Table 5. Saving percentage of various text algorithm

Data Set	Input text size(in bits)	LZW SP	Huffman SP	Proposed algorithm SP
LY844744.1	44	24.6%	45%	79.8%
S56177.1	609	23%	40.9%	67.7%
EU275349.1	972	34.4%	43%	71%

It is clear from table 5 that the LZW and Huffman coding algorithm for DNA data compression has a low saving percentage were as the proposed algorithm has an average saving percentage of ±72.8. Figure 6 shows the graph of saving a percentage of LZW, Huffman, and the proposed algorithm tested on three random datasets.

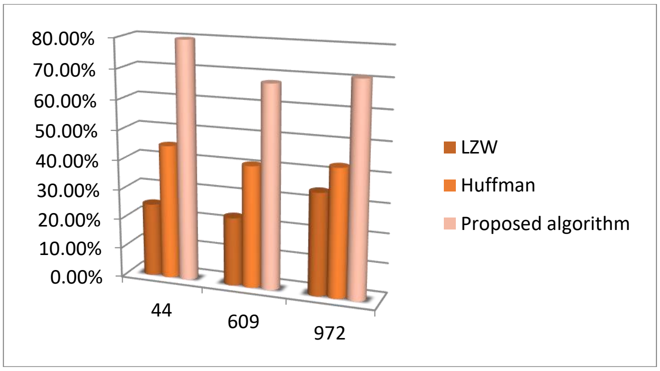
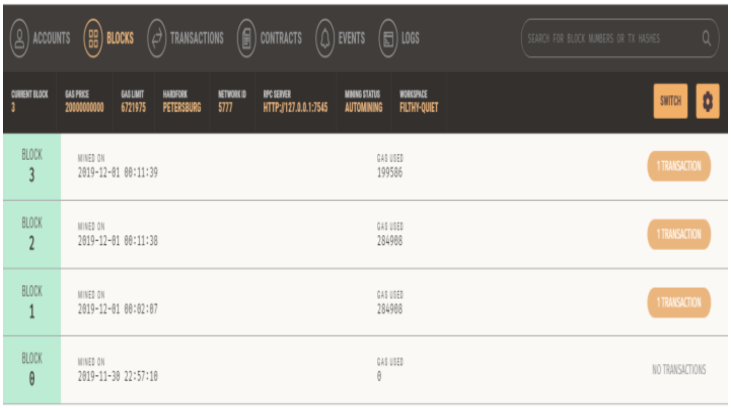


Figure 6. Saving percentage comparison graph for random dataset

Security of data is achieved through blockchain technology where the result of the compression module is secured using a cryptographic hash function such as SHA 256. The compressed DNA data is stored in blocks that are linked together using the hash value. The property of the hash function will not allow any type of change in an nth block as it directly gets reflected in the n-1th block. This process is similar for all consecutive blocks and will make the chain completely different. This unchangeable feature of blockchain is termed immutability. The below figure shows the storage of compressed DNA data in blocks.

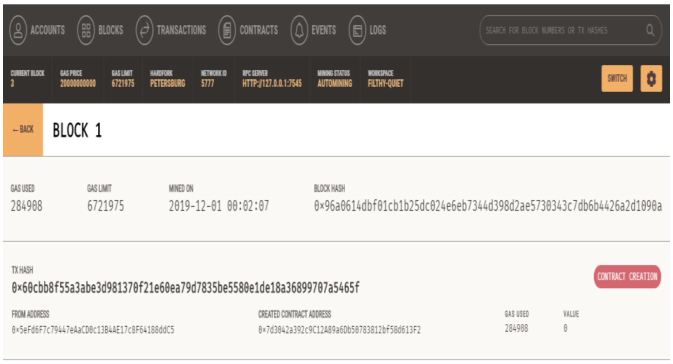


CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
3	20000000000	6721975	PETERSBURG	5777	HTTP://127.0.0.1:7545	AUTOMINING	FILTHY-QUIET

BLOCK	MINED ON	GAS USED	
3	2019-12-01 00:11:39	199586	1 TRANSACTION
2	2019-12-01 00:11:38	284908	1 TRANSACTION
1	2019-12-01 00:02:07	284908	1 TRANSACTION
0	2019-11-30 22:57:10	0	NO TRANSACTIONS

Figure 7. Exchanges of ethereum

Each block contains complete detail about the generated hash value of DNA sequence, the source address of that block, and its creation time i.e mined time, etc. Figure 8 shows complete detail about block 1.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
3	20000000000	6721975	PETERSBURG	5777	HTTP://127.0.0.1:7545	AUTOMINING	FILTHY-QUIET

BLOCK 1

GAS USED	GAS LIMIT	MINED ON	BLOCK HASH
284908	6721975	2019-12-01 00:02:07	0×96a0614dbf01cb1b25dc024e6eb7344d398d2ae5730343c7db6b4426a2d1090a

TX HASH
0×60cbb8f55a3abe3d981370f21e60ea79d7835be5580e1de18a36899707a5465f

FROM ADDRESS	CREATED CONTRACT ADDRESS	GAS USED	VALUE
0×5eFd6F7c79447eAaCD0c13B4AE17c8F64188ddC5	0×7d3042a392c9C12A89a6Db50783812bf58d613F2	284908	0

Figure 8. Ethereum block detailed preview.

The result shows that the proposed algorithm has a good compression ratio compared to traditional compression methods and in addition to that high security is achieved using blockchain technology.

7 Conclusion

In today's pandemic situation prior analysis of disease is considered an important factor in healthcare. Early diagnosis of the disease can be carried out using a DNA sequence as it contains all genetic information related to organisms. However, storage of this immense amount of data is a big issue.

Compressing these files is the most appropriate solution which not only reduces size but reduces the need for resources for transmission and that means it reduces transmission time as well as space. Therefore, this work proposes an ASCII-based delta computation of FASTA, a compression algorithm for the management and storage of NCBI data also for methods like LZW and Huffman code, compression ratio comparisons were determined for homosapiens, and from the results, it is clear that the proposed algorithm shows a good compression ratio for some randomly selected data sets. DNA data security is another big issue for which blockchain is considered as one of the best solutions as it has already proved its immutability property in the field of crypto-currencies. It can be imagined as a link list of records also known as blocks where each block contains a hash value of its previous block that makes it difficult or impossible to manipulate. The resultant compressed data will now get secured using blockchain technology.

References

[1] Sardaraz, M.; Tahir, M.; Ikram, A.A. Advances in high throughput DNA sequence data compression. J. Bioinf. Comput. Biol. 2016, 14, 1630002.

[2] Zhu, Z.; Zhang, Y.; Ji, Z.; He, S.; Yang, X. High-throughput DNA sequence data compression. Briefings Bioinf. 2015, 16, 1–15.

[3] Jones, D.C.; Ruzzo, W.L.; Peng, X.; Katze, M.G. Compression of next-generation sequencing reads aided by highly efficient de novo assembly. Nucleic Acids Res. 2012, 40, e171.

[4] Hach, F.; Numanagic, I.; Sahinalp, S.C. DeeZ: Reference-based compression by local assembly. Nat. Methods 2014, 11, 1082–1084.

[5] Mohammed, M.H.; Dutta, A.; Bost, T.; Chadaram, S. DELIMINATE—A fast and efficient method for loss-less compression of genomic sequences. Bioinformatics 2012, 28, 2527–2529.

[6] What is DNA?–Genetics Home Reference–NIH. Available online: https://ghr.nlm.nih.gov/primer/basics/ dna (accessed on 23 August 2018).

[7] Seo-Joon Lee, Gyoun-Yon Cho, Fumiaki Ikeno and Tae-Ro Lee. "BAQALC: Blockchain Applied Lossless EfficientTransmission of DNA Sequencing Data for NextGeneration Medical Informatics". Appl. Sci. 27 August 2018, 8(9), 1471

[8] G. Mathur, A. Pandey and S. Goyal, "Immutable DNA Sequence Data Transmission for Next Generation Bioinformatics Using Blockchain Technology," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170715.

[9] Aste, T.; Tasca, P.; Di Matteo, T. Blockchain Technologies: The Foreseeable Impact on Society and Industry. Computer 2017, 50, 18–28.

[10] Roehrs, A.; da Costa, C.A.; da Rosa Righi, R.; Alex, R.; Costa, C.A.; Righi, R.R. OmniPHR: A distributed architecture model to integrate personal health records. J. Biomed. Inform. 2017, 71, 70–81.

[11] Sleiman, M.D.; Lauf, A.P.; Yampolskiy, R. Bitcoin Message: Data Insertion on a Proof-of-Work Cryptocurrency System. In Proceedings of the 2015 International Conference on Cyberworlds (CW), Visby, Sweden, 7–9 October 2015; pp. 332–336.

[12] Khan, M.A.; Salah, K. IoT security:Review, blockchain solutions, and open challenges.Future Gener.Comput. Syst. 2018, 82, 395–411.

[13] Aumasson, J. Serious Cryptography: A Practical Introduction to Modern Encryption; No Starch Press: San Francisco, CA, USA, 2017.

[14] Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Boston, MA, USA, 11–14 December 2017; pp. 557–564.

[15] Khalid Sayood, " Introduction to Data Compression (Fifth Edition) ", The Morgan Kaufmann Series in Multimedia Information and Systems,2018, Pages 1-10.

[16] D.Mathpal, M. Darji, S. Mehta." A Research Paper on Lossless Data CompressionTechniques". IJIRST – International Journal for Innovative Research in Science & Technology, Volume 4 , Issue 1 , June 2017.

[17] R. Singh, B. B singh, "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", ijarcsse, Volume 3, Issue 3, March 2013

[18] M. Sharma, “Compression using Huffman Coding ” IJCSNS Intern ational Journal of Computer Science and Network Security, VOL.10 No.5, May 2010
[19] Al-Okaily, A.; Almarri, B.; Al Yami, S.; Huang, C.H. Toward a Better Compression for DNA Sequences Using Huffman Encoding. J. Comput. Biol. 2017, 24, 280–288.
[20] Ziv, J.; Lempel, A. Compression of Individual Sequences via Variable-Rate Coding. IEEE Trans. Inf. Theory 1978, 24, 530–536.
[21] Pinho, A.J.; Pratas, D. MFCompress: A compression tool for FASTA and multi-FASTA data. Bioinformatics 2014, 30, 117–118.
[22] Nour S. Bakr , Amr A. Sharawi ,”DNA Lossless Compression Algorithms: Review”, American Journal of Bioinformatics Research, e-ISSN: 2167-6976,2013; 3(3): 72-81.
[23] Ziv, J.; Lempel, A. Universal Algorithm for Sequential Data Compression. IEEE Trans. Inf. Theory 1977, 23, 337–343.
[24] LichengWanga, Xiaoying Shen a, Jing Li b, Jun Shao c, Yixian Yang. "Cryptographic primitives in blockchains". Elsevier Journal of Network and Computer Applications 127 (2019) 43–58.
[25] R. Karthik, V. Ramesh, M. Siva,” Text Data Compression and Decompression Using Modified Deflate Algorithm”, ICETAC 2K17, ISSN: 2395-1303,2017.