Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 6, Issue 11, November -2019

Algorithm for DNA as archival storage solution

[1]Parihar Devendra, [2]Managori Husain, [3]Parihar Pratibha*

[1,2,3]*Department of Biotechnology, Genetics and Bioinformatics, N. V. Patel college of Pure and Applied Sciences, Sardar Patel University, Vallabh Vidyanagar-388120, Gujarat, India.*

ABSTRACT

Background:

Big Data is an ocean of information generated by humans or machines in high volume, velocity and/or variety. Such voluminous data cannot be processed with present database or software and due to limited storage capacity failed to capture, store, manage and analyse the data. Data generated by Large Hadron Collider (LHC) is approximately 25 petabytes per year, thus need for a better data storage medium. Deoxyribonucleic acid (DNA) is extremely dense with an approximate storage limit of 1 exabyte/mm^3 (10^9 GB/mm^3), and has a half-life of 500 years far more long-lasting than the magnetic tape and hard-drives. Theoretically, one can encode 2 bits per nucleotide in DNA that can store 455 exabytes per gram maximum data in single-stranded DNA.

Results

The present work proposed an algorithm for encoding and decoding, based on compression and decompression method. The process involved Huffman algorithm rely on the frequency of each symbol (code) appeared in the text. It reads any form of data in the text file and convert the text file into binary string and finally compressed it to nucleotide code (A,T,G and C). It comprises of DNA-based data archival in a simplest and better compressed form.

Conclusions

In future this storage device can prove as a boon to reduce e-waste and in large scale secured storage system.

Keywords: *Encoding, Decoding, algorithm, compression, big data.*

INTRODUCTION

With the exponential advancement of data generating technologies, the biggest challenge faced by the scientist world-wide is storage of massive information. According to the recent statistical analysis from Domo in year 2017, currently the world is engendering a gigantic 2.5 quintillion bytes of data per day. Moreover, in the life science, the impact of big data analytics and explosion of data generated by various molecular biology projects and omics research upshots in doubling of data in a very short span of time. However, lack of further advancement in the present technology, such as magnetic and optical media, compelled development of new technology. Further, storage of such enormous information crafted a need of superior, universal, reliable, authentic and non-obsolete enduring storage device[1]. Nature's storage medium is incomparable with any artificial storage device with huge and long term storage capacity thus makes DNA as saviour. According to study conducted by the Global E-waste Monitor 2017, 44.7 million metric tonnes of e-waste was generated globally. Their statistical analysis showed globally Asia is the major generator of e-waste which is around 18.2 Mt, following 12.3 Mt by Europe then Americas (11.3 Mt), Africa (2.2 Mt), and least 0.7 Mt by Oceania [2]. This e-waste get accumulated and contaminate the soil, water and food with detrimental ingredients such as cadmium, mercury, chromium, lead, brominated flame retardants and polychlorinated biphenyls and causes substantial neurological, digestive, bone and respiratory issues[3]. It is reported that 80% of the children in Guiyu are facing problems associated with respiratory ailments and are at high risk of lead poisoning[4]. Thus, usage of deoxyribonucleic acid (DNA) as storing device can prove as a boon to reduce the burden of e-waste on the earth as well as its hazardous effect.

METHODOLOGY

The conceptual idea behind the novel approach of digital DNA storage is conversion of long binary strings of ones and zeros into the naturally occurring four types of nucleotides codes including adenine (A), cytosine (C), guanine (G) and thymine (T) [5]. A single nucleotide code signifies 2 bits of digital information[6]. This digital data is fragmented and synthesis as artificial synthetic DNA molecule in the laboratory which can be dehydrated and preserved for long-term storage. Further data retrieval involves sequencing the DNA molecule using specific primers and decoding the information back to the original digital data. One gram of single stranded DNA (ssDNA) can store 455 EB of data[7]. As compared to hard disk and solid-state drive (SSD), DNA is highly condensed, stable for long time with a half-life of more than 500 years even in harsh environmental conditions[8]. However, the cost of artificial DNA synthesis and sequencing is the main limiting factor affecting the DNA storage system. The algorithm for DNA storage system is advancing at a very high rate. In 1999, the system could encode and recover maximum up to 23 character message, which was updated to 739kb message in 2013[9].

Appropriate coding is a foremost part of DNA storage system to get minimum error and maximum efficiency. The method is an modified Huffman algorithm that convert text data into DNA nucleotides (A,T, G and C) by compressing and storing multiple copies along with providing security. The program read the text file and convert the text file into string. Frequency of each character is calculated by the freqTab class. This class uses hash map to sort characters as key and their values as frequency. Based on the sorted map a binary heap/tree is generated. In this algorithm the most frequent character get the least key value and vice-versa.
Information in text.txt used to create character map which is used to develop 4-array huffman tree. Further 4-array huffman tree transverse to get ATGC code and finally data stored in new text.txt file. Further, in order to retrieve back the above compressed file, a decompressor is required. It will convert the ATGC into original file in new text file, it uses get code and convert it into ATGC.

RESULT

The algorithm

Compression of text file is through following steps

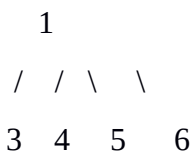
- The program first reads the text file converting it into a string
- Frequency of each character is calculated by the freq tab class. This class uses hash map to sort characters as key and their values as frequency.
- As per sorted Map a Binary Heap/Tree is generated.

For example a textfile with a string "abcdebbcccddddeeeee"

Frequency of each character a=1 b=3 c=4 d=5 e=6

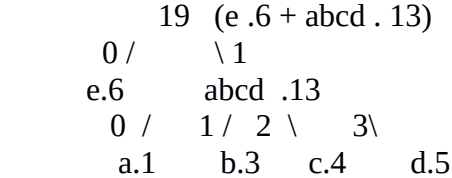
In this 1 (a) is the root and 5 (d) and 6(e) are leaf node and 3(b) and 4(c) are called inner node. This a normal binary tree called as 2 array tree .

The modification of occurs with a four branch tree



A minimum heap in created. In a minimum binary heap, the key at root must be least among all keys present in binary heap means root must always be smaller than its following nodes (parent < child).

A function in the minheap class known as minheapify pops or extracts four least frequent node from the tree. This function is used recursively for extracting four nodes from the four array tree.



As it is four array tree we assign 0 - A , 1 - T , 2 - G , 3 - C

Transversal of the tree will get value of each character as all character are at the leaf node

If value of e = 0 then ...

a = 1 0 which is equivalent to A T

b=1 1 means T T

c = 1 2 means T G

d = 1 3 means A C and so on..

Here the most frequent character get the least key value and vice-versa

Running the code for the string "abcdebbcccddddeeeee"

Will generate compressed text as " TATTTGTCATTTTTGTGTGTCTCTCTCAAAAA "

The uncompression part could be understood by

```
getCode(char c)  {  switch(c)  {
        case 'A':
          return 0;
        case 'T':
          return 1;
        case 'G':
          return 2;
        case 'C':
          return 3;                    }
```

This function takes the text as ATGC and converts it into 0123. Further checks if it's a leaf node or not.

Above algorithm was tested with a file given in the table

File name (original data before processing)	File size in kb	Total character in file	File size after binary conversion	file size after conversion in A,T,G,C code	Length of nucleotides
https://www.americanrhetoric.com /speeches/mlkihaveadream.htm	10kb	10241	70kb	27.8kb	28520

Table 1 shows the size of the file before and after encoding

Discussion

This paper describes an algorithm which compresses high voluminous data into nucleotides codes (A,T,G and C). The major limitation of DNA based storage systems is the cost of sequencing and the data retrieval rate. Oligonucleotide synthesis and sequencing processes is a time consuming process and necessitate skill acquaintance which makes the method inaccessible to general public use. Although DNA is scalable, dense, and highly steady (if stored properly) but the above-mentioned drawbacks cannot be ignored. The major obstacle for DNA based storage systems is oligonucleotide synthesis and sequencing cost.

Conclusion: DNA is a highly stable molecule with a potential as an ultimate archival storage solution. Binary information in the form of 0s and 1s are converted into As, Cs, Gs and Ts, ultimately which synthesized to artificially

oligonucleotides. It is estimated that 20 gms of DNA can hold all available data in the world. It reduces e-waste and is a secured way of data transferring , e.g. military implications.

Declarations

Acknowledgements
Gujarat State Biotechnology Mission (GSBTM), Department of Science and Technology, Government of Gujarat
Funding
This study was not funded by any source.

Availability of data and materials
All data and material are available.

Authors' contributions
All the participant researchers contributed to do this work. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
All the participant researchers are consent for publication.

Competing interests
The authors declare that they have no competing interests.

References :

1) De Silva PY, Ganegoda GU. New Trends of Digital Data Storage in DNA. *BioMed Research International*. 2016; 8072463. doi:10.1155/2016/8072463.
2) Baldé, C.P., Forti V., Gray, V., Kuehr, R., Stegmann,P. : The Global E-waste Monitor – 2017, United Nations University (UNU), International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), Bonn/Geneva/Vienna.
3) Mansour S.A. (2011) Chemical Pollutants Threatening Food Safety and Security: An Overview. In: Hefnawy M. (eds) Advances in Food Protection. NATO Science for Peace and Security Series A: Chemistry and Biology. Springer, Dordrecht.
4) Park JK, Hoerning L, Watry S, Burgett T, Matthias S (2017) Effects of Electronic Waste on Developing Countries. Adv Recycling Waste Manag 2: 128. doi:10.4172/2475-7675.1000128.
5) Shrivastava S., Badlani R. Data storage in DNA. *International Journal of Electrical Energy*. 2014;2(2):119–124.
6) De Silva, P. Y., & Ganegoda, G. U. (2016). New Trends of Digital Data Storage in DNA. *BioMed Research International*, *2016*, 8072463.
7) Castillo M. From hard drives to flash drives to DNA drives. *American Journal of Neuroradiology*. 2014;35(1):1–2. doi: 10.3174/ajnr.a3482.
8) Castillo M. From hard drives to flash drives to DNA drives. *American Journal of Neuroradiology*. 2014;35(1):1–2. doi: 10.3174/ajnr.a3482.
9) Goldman, N. et al. 2013. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*. 494, (2013), 77–80.

