Design of Effective Lossless Data Compression Technique for Multiple Genomic DNA Sequences

Mahmud Alosta[1], Alireza Souri[2]

[1] Software Engineering and IT Department, Ecole de technologie superieure, Montreal (Qc), Canada

[2] Department of Computer Engineering, Haliç University, Beyoğlu, İstanbul, Turkey

Emails: mahmud.alosta.1@ens.etsmtl.ca, alirezasouri@halic.edu.tr

Abstract

In recent years, a massive amount of genomic DNA sequences are being created which leads to the development of new storing and archiving methods. There is a major challenge to process, store or transmit the huge volume of DNA sequences data. To lessen the number of bits needed to store and transmit data, data compression (DC) techniques are proposed. Recently, DC becomes more popular, and large number of techniques is proposed with applications in several domains. In this paper, a lossless compression technique named Arithmetic coding is employed to compress DNA sequences. In order to validate the performance of the proposed model, the artificial genome dataset is used and the results are investigated interms of different evaluation parameters. Experiments were performed on artificial datasets and the compression performance of Arithmetic coding is compared to Huffman coding, LZW coding, and LZMA techniques. From simulation results, it is clear that the Arithmetic coding achieves significantly better compression with a compression ratio of 0.261 at the bit rate of 2.16 bpc.

Keywords: Arithmetic coding; Dataset; Data compression; DNA sequences; Lossless Compression

1. Introduction

As the DNA sequences are highly useful in various fields like biology, medicine, and genetics, large amount of DNA sequences has been generated rapidly [1-3]. It is also identified that there will be enormous growth in the volume of DNA sequences in the future. So, it is essential to handle the amount of DNA sequences effectively while storing or transmitting it. DNA is the fundamental unit of living things as the data to create a living organism is totally stored in it [4, 5]. DNA is a long sequence which is usually composed of four varieties of bases namely Cytosine (C), Adenine (A), Thymine (T), and Guanine (G). The combinations of C, A, T, G in the long DNA sequence is shown in Fig. 1. It is commonly represented by {A, C, G, T} and each one is denoted as nucleotides. Adenine is commonly connected by Cytosine and Thymine is connected using Guanine.

In replication process of DNA, when the 2 components are detached to act as templates to produce copies, an enzyme is known as DNA polymerases read the strand in 3' and 5' directions deploying the respective nucleotide [6-8].

The massive amount of DNA sequence imposes a new challenge for storage space and bandwidth resources. Generally, the publicly available gene datasets are archived as normal text files with increased burden of storage space and transmission [9, 10]. Without high speed internet, it is difficult or sometimes impossible to share genetic information across some parts of the world. Several researchers have been made to handle the storage of massive genomic datasets. One manner to deal with this large amount of genomic information is to compress. DC technique reduces the quantity of information transmitted/stored

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

[11]. It is appropriate for compressing images, text, and video/audio. The information could be alphanumerical character in text documents, number that represents the sample in image/audio series/waveforms of number generated with few procedures, and so on. Also, DC is called a manner of demonstrating information in its dense formation. It is employed from satellite imaging, medical imaging to WSNs.

Fig. 1. DNA sequence in text format

Due to the unique nature of DNA, it is not possible to apply conventional compression algorithm to compress DNA sequences [12, 13]. Few researches have only been performed on the compressions of genome datasets. Conventional compression techniques operate by identifying the repeated patterns for DNA encoding. Some of the existing techniques are BioCompress, Cfact, DNACompress, and DNAPack. Though various techniques are available, significantly higher compression ratio is not attained. The DNA sequences can be efficiently compressed only by exploiting the special properties of DNA [14]. The absence of efficient compression technique especially for DNA sequences motivated us to perform this work.

In this paper, a lossless compression technique named Arithmetic coding is employed to compress DNA sequences. To validate the performances of the presented method, the artificial genome dataset is used and the results are investigated interms of different evaluation parameters. Experiments were performed on artificial datasets and the compression performance of Arithmetic coding is compared Huffman coding, LZW coding, and LZMA techniques.

2. Review of Existing Compression Standards

Rashid [15] proposed a new combination of cryptography and steganography methods. The proposed method consists of two phases: the encrypted and hide the message and message extraction phases. The encrypt and hide message phase consists of cryptography phase and steganography phase that includes six steps, firstly Caesar cipher applied to encrypt the message, secondly convert ciphertext to DNA sequences, thirdly convert DNA character to their equivalent ASCII, fourthly convert ASCII to binary, fifthly shift binary based on a specific key. Finally, the sixth step hides the ciphertext in the cover image. In [16], the feature from protein sequence is found out through prolonging the concept of DPC, EDF, and KSB to PSSM. The fundamental data has been determined via a compression method called DCT and the method has been trained by SVM. The predictive accuracy has been additionally increased with GA approach.

Karmakar et al. [17] proposed an efficient and new sparse depiction that relied on spatial compression of video signal incorporated by a hyperchaotic DNA coded relied on encryption model which provides high performances. The improvement of compression efficacy is attained by presenting sparse coded on the video frame, and high safety is attained with five dimensional hyperchaotic DNA coded on the sparse coding frame. The new method is used for huge sets of video signals to test and compare its efficiency by lately presented video coded as well as encryption system. Alsaffar et al. [18] integrate among numerous phases of encryption technique: image steganography, DNA, GZIP, and AES. They presented increasing with factors and final step of DNA encryptions, as well the output of these procedures were compressed by a GZIP method, in which messages are transformed to a novel version besides its size decreased to (75%) when the messages are encrypted by means of AES encryptions for increasing the

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

safety levels. Further, LSB images Steganography method is used for hiding encrypted messages in higher quality images.

Afify et al. [19] discuss the applications of DNA coded features, services models, and safety problems. It proposes a method to secure information while transferring/storing, that can be low cost and secured by bio computation technique. The tools use DNA, ML and steganography, BD methods, and binary coding rule for making the method secure in which extra layers of bio-security, i.e., are highly efficient compared to traditional cryptographic techniques. An algorithm for building the extended BWT (eBWT) is presented in [20] comprises a string collection from its grammar-compressed representation. Our technique exploits the string repetitions captured by the grammar to boost the computation of the eBWT. Thus, the more repetitive the collection is, the lower are the resources we use per input symbol. We rely on a new grammar recently proposed at DCC'21 whose nonterminals serve as building blocks for inducing the eBWT. A relevant application for this idea is the construction of self-indexes for analyzing sequencing reads -- massive and repetitive string collections of raw genomic data.

Yang et al. [21] presented a 2 image compression encryptions system according to DNA method and fractional hyperchaotic scheme. Initially, 2 images are treated using DCT model. Next, the spectrum of the 2 images was organized in Z-scan, thus the 2 images are mixed and compressed to novel images. Lastly, the resultant images are encrypted with DNA coding. In [22], new hash functions were introduced which eliminate hash collision for DNA sequences. It provides accurate hash and produces hash value appropriate time. They projected 2 accurate strings matching methods according to the presented method. Initially, they replaced a conventional Hash-q algorithm. Next, enhanced the initial method with the shift size.

3. Lossless Compression algorithm on DNA Sequencing

The DNA sequences can be efficiently compressed only by exploiting the special properties of DNA. Arithmetic coding is employed to compress the DNA sequences. It generates variable length code and has greater than Huffman coding from several features. It can be extremely helpful under conditions in which the source contains small alphabet with skewed probability. If the string has been encoding utilize Arithmetic coding, frequent happening symbols were coding with smaller bits than infrequently happening symbols. It changes the input data as to floating point numbers from the range of [0, 1]. This technique was executed by splitting [0-1] as to segment and the length of all segments are dependent upon probabilities of all symbols. Afterward, the output data was recognized from the respective segment dependent upon symbols. It could not easier for implementing if related to another technique. The advantage of Arithmetic coding over Huffman coding was ability for segregating the model and coding feature of compression technique. The algorithm of the Arithmetic encoder and decoder is given in Algorithm 1 and Algorithm 2.

Algorithm 1: Arithmetic Encoding Procedure

Step 1: Call encoder symbol frequently to all symbols from the message

Step 2: Confirm that a notable “terminator” symbol has been encoded later then communicate some values from the range [LL, HH].

Step 3: encoder-symbol (symbol, cum_freq)

range = HH - LL

HH= LL + range * cum_freq [symbol-1]

LL = LL + range * cum_freq [symbol1]

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

Algorithm 2: Arithmetic Decoding Procedure

Step 1: "Value" is the number which is obtained

Step 2: Continue calling decoder-symbol still the terminator symbol has been returned.

Step 3: decoder-symbol (cum_freq)

```
Define symbol such that
    cum_freq[symbol]<=(value-LL)/(HH-LL)<cum_freq[symbol-1]
    range = HH - LL
    HH = LL + range * cum_freq [symbol-1]
    LL = LL + range * cum_freq [symbol]
    return symbol
```

The arithmetic coding [23] has been lossless data compression technique which allocates short code words to symbols with higher event probability and leaf the extended code word for the symbol with lesser happening probability. An important aim of arithmetic coding is which all symbols of message were demonstrated in half-open subinterval of primary half-open interval [0, 1), and next all subsequent symbols under the message reduce the interval size by an equivalent sub-interval based on symbol existence probabilities [19]. The encoder as well as decoder functions are as follows: (1) Encoder: A primary interval was [0, 1). If the primary symbol "b" is encoding, it restricts the interval in [0, 1) to [0.3, 0.6), where [0.3, 0.6) has the interval allocated to "b". Then the second symbol "c" is encoded, it narrows the [0.3, 0.6) to [0.48, 0.6) based on the interval allocated to "c". Eventually, the message "bca" has been encoded as interval [0.48, 0.516). (2) Decoder: The decoded technique employs a similar possibility model. With the interval [0.48, 0.516) of existence encoder, a primary symbol "b" has been decoded as the [0.48, 0.516) is sub-interval of interval [0.3, 0.6) that is the interval allocated to "b". The sub-interval of "b" is more separated in similar approach for deriving the following symbols still the interval of existence decoder was equivalent to interval of existence encoder such as [0.48, 0.516) in this sample.

4. Performance Evaluation

For ensuring the efficacy of the Arithmetic coding on DNA sequence compression, a comparative analysis is made with Huffman coding, LZW, and LZMA. For experimentation, publicly available artificial genome dataset is used [24]. The artificial dataset contains 6 sequences which are implanted with exact subsequences of length 100. The characters in the dataset are randomly generated with the four bases A, T, C, and G. Inter-sequences similarities are included as identical subsequences across the six sequences. Particularly, the ith sequence was generated with i groups of subsequences, each of which is composed of 100 non-overlapping subsequences interleaved with one base symbol. The subsequences in the k^{th} group of the ith sequence were randomly matched with those in the $(k+1)^{th}$ group of the $(i+1)^{th}$ sequence.

Table 1 offers a comprehensive result analysis of the proposed with existing compression techniques. Fig. 2 investigates the CR analysis of the proposed model and the value of CR should be minimum for better performance. The figure depicted that the proposed model has resulted to lower CR over the other techniques. For instance, on the Seq1.fasta dataset, the arithmetic coding offers a reduced CR of 0.283 whereas the LZMA, Huffman, and LZW techniques have obtained a higher CR of 0.331, 0.285, and 0.478 respectively. Besides, on the Seq2.fasta dataset, the arithmetic coding offers a minimum CR of 0.272 whereas the LZMA, Huffman, and LZW approaches have gained an increased CR of 0.314, 0.284, and 0.434 correspondingly. Additionally, on the Seq3.fasta dataset, the arithmetic coding offers a lower CR of 0.269 whereas the LZMA, Huffman, and LZW techniques have obtained a superior CR of 0.308, 0.284,

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

and 0.411 respectively. In line with, the Seq4.fasta dataset, the arithmetic coding offers a reduced CR of 0.213 whereas the LZMA, Huffman, and LZW techniques have reached an enhanced CR of 0.305, 0.284, and 0.397 correspondingly.

Table 1 CR and CF analysis of different compression models

Dataset	Compression ratio				Compression factor			
	LZMA	Arithmetic	Huffman	LZW	LZMA	Arithmetic	Huffman	LZW
Seq1.fasta	0.331	0.283	0.285	0.478	3.018	3.536	3.509	2.090
Seq2.fasta	0.314	0.272	0.284	0.434	3.175	3.664	3.517	2.304
Seq3.fasta	0.308	0.269	0.284	0.411	3.238	3.714	3.516	2.428
Seq4.fasta	0.305	0.213	0.284	0.397	3.271	3.742	3.517	2.517
Seq5.fasta	0.301	0.265	0.409	0.386	3.326	3.760	2.443	2.588
Seq6.fasta	0.302	0.265	0.284	0.377	3.302	3.772	3.518	2.647

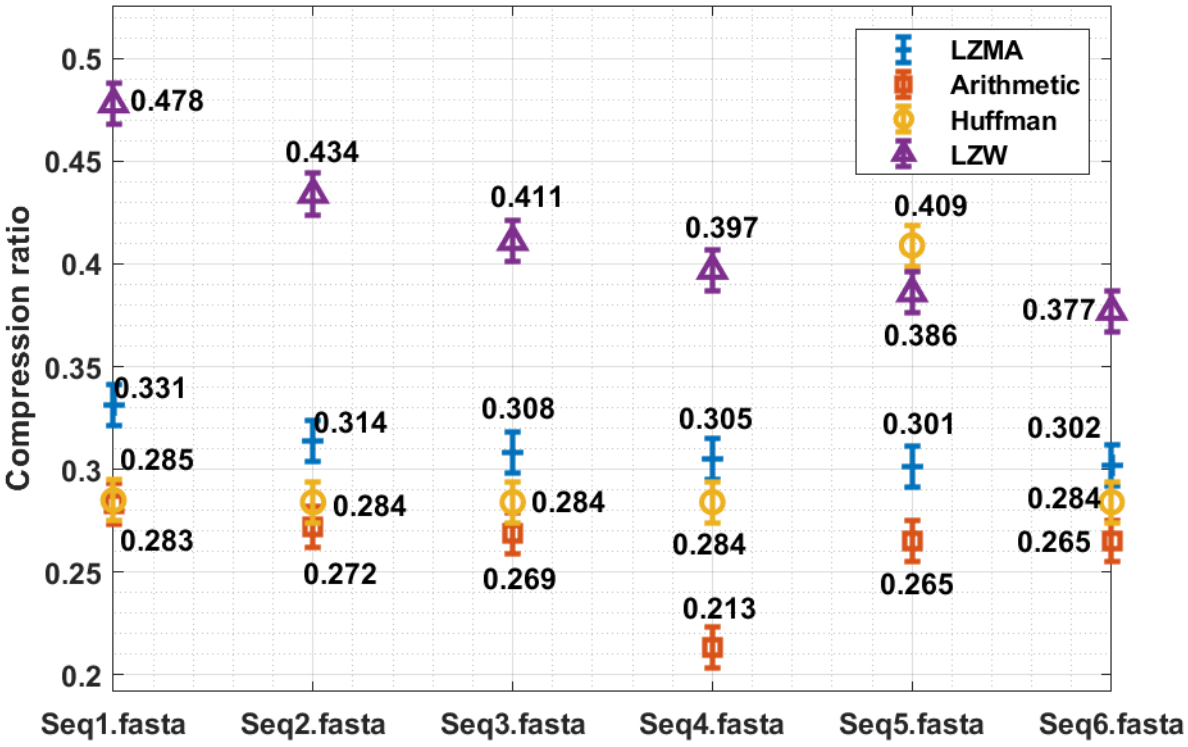

Fig. 2. CR analysis of different compression models on DNA sequences

Along with that, on the Seq5.fasta dataset, the arithmetic coding gives a decreased CR of 0.265 whereas the LZMA, Huffman, and LZW techniques have achieved a maximal CR of 0.301, 0.409, and 0.4386 respectively. Finally, on the Seq6.fasta dataset, the arithmetic coding offers the least CR of 0.265 whereas the LZMA, Huffman, and LZW methodologies have obtained a higher CR of 0.302, 0.284, and 0.377 correspondingly.

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

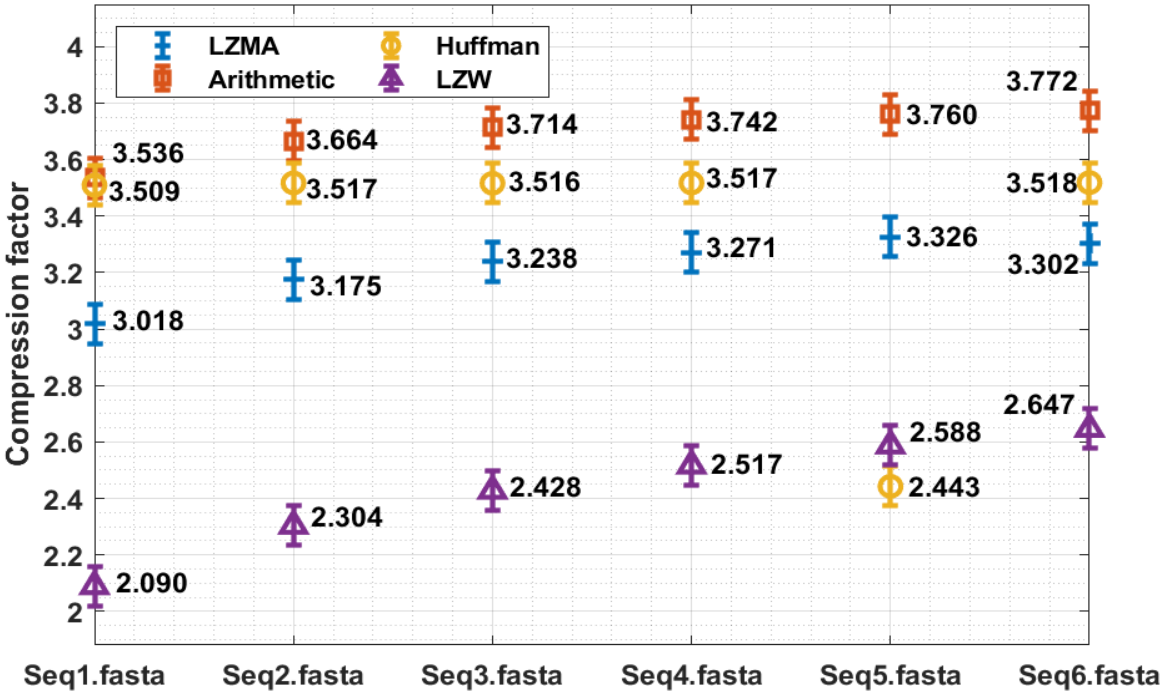

Fig. 3. CF analysis of different compression models on DNA sequences

Next, a comprehensive CF analysis of the proposed model with the state of art compression techniques is provided in Fig. 3. The figure demonstrated that the proposed model has achieved a higher CF and yields effective compression. For instance, on the Seq1.fasta dataset, the arithmetic coding offers an increased CF of 3.536 whereas the LZMA, Huffman, and LZW techniques have obtained a higher CR of 3.018, 3.509, and 2.090 respectively. Moreover, on the Seq2.fasta dataset, the arithmetic coding provides an enhanced CF of 3.664 whereas the LZMA, Huffman, and LZW manners have reached a maximal CR of 3.175, 3.517, and 2.304 correspondingly. Furthermore, on the Seq3.fasta dataset, the arithmetic coding offers an enhanced CF of 3.714 whereas the LZMA, Huffman, and LZW approaches have attained a superior CR of 3.238, 3.516, and 2.428 respectively. Eventually, on the Seq4.fasta dataset, the arithmetic coding offers an increased CF of 3.742 whereas the LZMA, Huffman, and LZW methods have attained a superior CR of 3.271, 3.517, and 2.517 correspondingly. Meanwhile, on the Seq5.fasta dataset, the arithmetic coding offers a maximum CF of 3.760 whereas the LZMA, Huffman, and LZW techniques have obtained a higher CR of 3.326, 2.443, and 2.588 correspondingly. At last, on the Seq6.fasta dataset, the arithmetic coding offers an increased CF of 3.772 whereas the LZMA, Huffman, and LZW methodologies have reached an increased CR of 3.302, 3.518, and 2.647 correspondingly.

Table 2 BPC and CT analysis of various compression models

Dataset	Bit per character				Compression time (s)			
	LZMA	Arithmetic	Huffman	LZW	LZMA	Arithmetic	Huffman	LZW
Seq1.fasta	2.65	2.26	2.27	3.82	11.67	10.00	10.50	13.89
Seq2.fasta	2.51	2.18	2.27	3.47	18.99	15.00	16.00	19.88
Seq3.fasta	2.47	2.15	2.27	3.29	27.90	23.09	24.89	29.90
Seq4.fasta	2.44	2.13	2.27	3.17	31.33	28.80	29.90	33.43
Seq5.fasta	2.40	2.12	3.27	3.09	44.89	41.60	42.78	46.23
Seq6.fasta	2.42	2.12	2.27	3.02	75.65	70.44	72.87	76.96

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

Table 2 provides a comprehensive result analysis of the proposed with existing algorithms interms of Bit per character (BPC) and compression time (CT).

Fig. 4 examines the BPC analysis of the proposed approach and the value of BPC should be minimal for better performance. The figure demonstrated that the proposed model has resulted to lower BPC over the other techniques. For instance, on the Seq1.fasta dataset, the arithmetic coding offers a reduced BPC of 2.26 whereas the LZMA, Huffman, and LZW techniques have obtained a higher BPC of 2.65, 2.27, and 3.82 respectively. Similarly, on the Seq2.fasta dataset, the arithmetic coding offers the least BPC of 2.18 whereas the LZMA, Huffman, and LZW methods have achieved a superior BPC of 2.51, 2.27, and 3.47 respectively. Also, on the Seq3.fasta dataset, the arithmetic coding offers a lower BPC of 2.15 whereas the LZMA, Huffman, and LZW techniques have gained a maximum BPC of 2.47, 2.27, and 3.29 respectively. Likewise, on the Seq4.fasta dataset, the arithmetic coding offers a reduced BPC of 2.13 whereas the LZMA, Huffman, and LZW methodologies have obtained a higher BPC of 2.44, 2.27, and 3.17 respectively. Followed by, on the Seq5.fasta dataset, the arithmetic coding offers a decreased BPC of 2.12 whereas the LZMA, Huffman, and LZW techniques have obtained a higher BPC of 2.40, 3.27, and 3.09 correspondingly. Eventually, on the Seq6.fasta dataset, the arithmetic coding offers a reduced BPC of 2.12 whereas the LZMA, Huffman, and LZW methodologies have obtained a higher BPC of 2.42, 2.12, and 2.27 respectively.

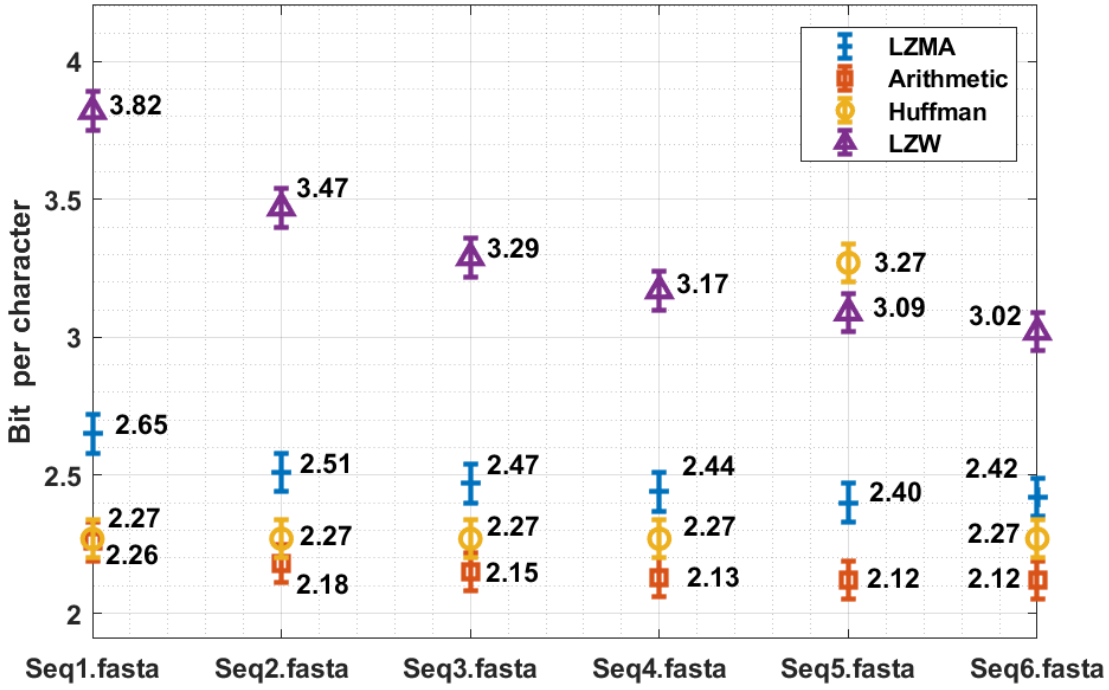

Fig. 4. BPC analysis of different compression models on DNA sequences

Fig. 5 explores the CT analysis of the proposed model and the value of CT should be minimum for better performance. The figure outperformed that the proposed technique has resulted in the least CT over the other techniques. For instance, on the Seq1.fasta dataset, the arithmetic coding offers a reduced CT of 10.00 whereas the LZMA, Huffman, and LZW approaches have gained a superior CT of 11.67, 10.50, and 13.89 correspondingly. At the same time, on the Seq2.fasta dataset, the arithmetic coding gives a decreased CT of 15.00 whereas the LZMA, Huffman, and LZW techniques have attained a higher CT of 18.99, 16.00, and 19.88 correspondingly. In addition, on the Seq3.fasta dataset, the arithmetic coding provides a minimal CT of 23.09 whereas the LZMA, Huffman, and LZW techniques have gained a higher CT of 27.90, 24.89, and 29.90 correspondingly. Then, on the Seq4.fasta dataset, the arithmetic coding offers a minimal CT of 28.80 whereas the LZMA, Huffman, and LZW methods have achieved a superior CT of 31.33, 29.90, and 33.43 correspondingly. Afterward, on the Seq5.fasta dataset, the arithmetic coding offers a reduced CT of 41.60 whereas the LZMA, Huffman, and LZW algorithms have obtained a higher CT of 44.89, 42.78, and 46.23 respectively. Lastly, on the Seq6.fasta dataset, the arithmetic

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

coding offers a lower CT of 70.44 whereas the LZMA, Huffman, and LZW approaches have gained an increased CT of 75.65, 72.87, and 76.96 correspondingly.

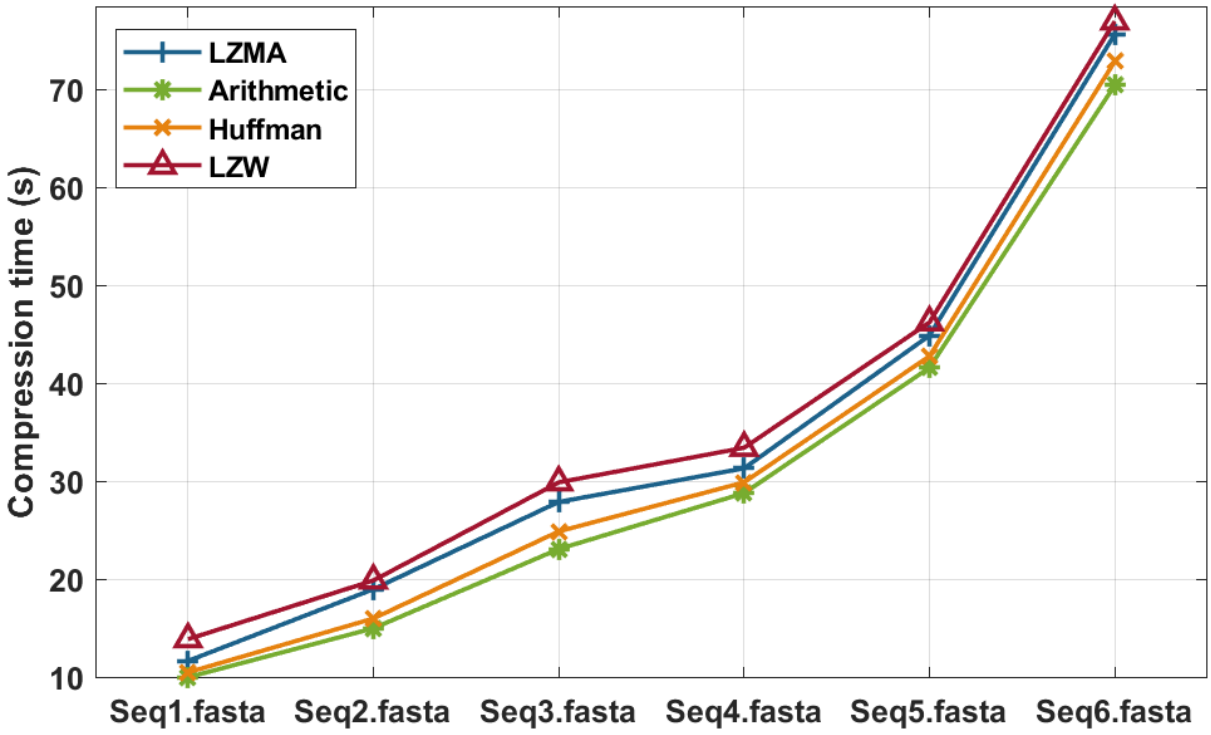

Fig. 5. CT analysis of different compression models on DNA sequences

5. Conclusion

To reduce the number of bits required to store and transmit DNA sequences, DC is proposed. DC is the process of reducing the amount of data without negotiating the data quality to a certain extent. In this paper, a lossless compression technique named Arithmetic code is employed to compress DNA sequences. It is a variable length coding technique which is highly useful in situations where the sources contain small alphabets with skewed probabilities. Experiments were performed on artificial datasets and the compression performance of Arithmetic coding is compared Huffman coding, LZW coding, and LZMA technique. From simulation results, it is clear that the Arithmetic coding attains meaningfully improved compression with a compression ratio of 0.261 at the bit rate of 2.16 bpc.

References

[1] Pratas, D., Hosseini, M. and Pinho, A.J., 2019, June. GeCo2: An optimized tool for lossless compression and analysis of DNA sequences. In International Conference on Practical Applications of Computational Biology & Bioinformatics (pp. 137-145). Springer, Cham.

[2] Hossein, S.M., De, D., Mohapatra, P.K.D., Mondal, S.P., Ahmadian, A., Ghaemi, F. and Senu, N., 2020. DNA Sequences Compression by GP² R and Selective Encryption Using Modified RSA Technique. IEEE Access, 8, pp.76880-76895.

[3] Saada, B. and Zhang, J., 2018. DNA sequence compression technique based on nucleotides occurrence. In Proceedings of the international multiconference of engineers and computer scientists (Vol. 1, pp. 14-16).

[4] Jahaan, A., Ravi, T.N. and Panneer Arokiaraj, S., 2017. A Comparative Study and Survey on Existing DNA Compression Techniques. International Journal of Advanced Research in Computer Science, 8(3).

[5] Mansouri, D. and Yuan, X., 2018, December. One-bit dna compression algorithm. In International Conference on Neural Information Processing (pp. 378-386). Springer, Cham.

[6] Cheng, K.O., Law, N.F. and Siu, W.C., 2017. Clustering-based compression for population DNA sequences. IEEE/ACM transactions on computational biology and bioinformatics, 16(1), pp.208-221.

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021

[7] Pasricha, N. and Hayes, C., 2019, December. Detecting bot behaviour in social media using digital dna compression. In 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science. AICS (Artificial Intelligence and Cognitive Science) 2019.
[8] Al-Okaily, A., Almarri, B., Al Yami, S. and Huang, C.H., 2017. Toward a better compression for DNA sequences using Huffman encoding. Journal of Computational Biology, 24(4), pp.280-288.
[9] Kerbiriou, M. and Chikhi, R., 2019, May. Parallel decompression of gzip-compressed files and random access to DNA sequences. In 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 209-217). IEEE.
[10] Yin, C., 2019. Encoding and decoding DNA sequences by integer chaos game representation. Journal of Computational Biology, 26(2), pp.143-151.
[11] Bakr, N.S. and Sharawi, A.A., 2017, December. Improve the compression of bacterial DNA sequence. In 2017 13th International Computer Engineering Conference (ICENCO) (pp. 286-290). IEEE.
[12] Habib, N., Ahmed, K., Jabin, I. and Rahman, M.M., 2018. Modified HuffBit compress algorithm–an application of R. Journal of integrative bioinformatics, 15(3).
[13] Pratas, D. and Pinho, A.J., 2018, May. A DNA sequence corpus for compression benchmark. In International Conference on Practical Applications of Computational Biology & Bioinformatics (pp. 208-215). Springer, Cham.
[14] Najam, M., Rasool, R.U., Ahmad, H.F., Ashraf, U. and Malik, A.W., 2019. Pattern matching for dna sequencing data using multiple bloom filters. BioMed research international, 2019.
[15] Rashid, O.F., 2021. Text Encryption and Hiding based on DNA Cryptography and Image Steganography. International Journal of Computing and Digital System.
[16] Barukab, O., Ali, F. and Khan, S.A., 2021. DBP-GAPred: An intelligent method for prediction of DNA-binding proteins types by enhanced evolutionary profile features with ensemble learning. Journal of Bioinformatics and Computational Biology, p.2150018.
[17] Karmakar, J., Pathak, A., Nandi, D. and Mandal, M.K., 2021. Sparse representation based compressive video encryption using hyper-chaos and DNA coding. Digital Signal Processing, p.103143.
[18] Alsaffar, Q.S., Mohaisen, H.N. and Almashhdini, F.N., 2021, February. An encryption based on DNA and AES algorithms for hiding a compressed text in colored Image. In IOP Conference Series: Materials Science and Engineering (Vol. 1058, No. 1, p. 012048). IOP Publishing.
[19] Afify, F.M. and Rahouma, K.H., 2021, March. Applying Machine Learning for Securing Data Storage Using Random DNA Sequences and Pseudo-Random Sequence Generators. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 286-298). Springer, Cham.
[20] Díaz-Domínguez, D. and Navarro, G., 2021. Efficient construction of the extended BWT from grammar-compressed DNA sequencing reads. arXiv preprint arXiv:2102.03961.
[21] Yang, Y.G., Guan, B.W., Zhou, Y.H. and Shi, W.M., 2021. Double image compression-encryption algorithm based on fractional order hyper chaotic system and DNA approach. Multimedia Tools and Applications, 80(1), pp.691-710.
[22] Karcioglu, A.A. and Bulut, H., 2021. Improving hash-q exact string matching algorithm with perfect hashing for DNA sequences. Computers in Biology and Medicine, 131, p.104292.
[23] Hao, W., Xiang, L., Li, Y., Yang, P. and Shen, X., 2018. Reversible natural language watermarking using synonym substitution and arithmetic coding. Comput. Mater. Contin, 55, pp.541-559.
[24] http://www.eie.polyu.edu.hk/~nflaw/DNAComp/

DOI: 10.5281/zenodo.5497475
Received: April 08, 2021 Accepted: August 12, 2021