DNA Sequences Compression Techniques Based on Modified DNABIT Algorithm

Bacem Saada, *Member, IAENG*, Jing Zhang

Abstract— **The large amount of DNA sequences stored on online databases has led scientists to implement compression algorithms for DNA sequences. These algorithms attempt to remove the redundant and unuseful data before storing it in databases. In this paper, we will present a two phases compression algorithm based on the binary representation of DNA sequences. In the first phase, we will use a modified version of DNABIT compression algorithm to compress and convert the DNA sequence into binary representation. Thereafter, we will compress the resulting DNA using the Extended-ASCII encoding through which one character can represent four nucleotides or more. The remarkable compression ratio of our algorithm makes its use interesting.**

Index Terms—**Extended-ASCII code, DNA compression, horizontal compression, DNABIT Compress;**

I. INTRODUCTION

Nowadays, we are living in the era of internet of things due to the use of billions of devices connected together and the significant increase in the dataflow stored and transmitted between them. Thus, every day, a huge quantity of digital information is used, shared and analyzed.

Powerful computers are used to properly analyze and store this data. Consequently, two problems have arisen. First, the encoding of the data and second, the time required to process them. As a result, to reduce data sizes, many data compression methods have been implemented. Compressors such as JPEG and MPEG are lossy compressors that try to remove some information that human being cannot notice in images. Lossless compressors, on the other hand, compress data without any loss of its information. Therefore, they are used for text compression methods and thus for DNA sequences chain.

Technological evolution has led to the birth of bioinformatics research area which processes and analyzes different living beings' data. The essential element in achieving these treatments is the Deoxyribonucleic Acid or DNA, which is a biomolecule present in all cells. This biomolecule contains the genetic information required for the functioning and development of all living beings. Each monomer constituting it is a nucleotide, which is composed of a nitrogenous base; adenine (A), cytosine (C), guanine (G) or thymine (T). GenBank, managed by the International Nucleotide Sequence Database Collaboration, is a free access database that contains a large amount of DNA sequences which are stored in raw format and that may consequently lead to redundant data. For this reason, we aim at proposing DNA sequences compression algorithms that reduce the size and so thoroughly analyze and choose the data that will be stored.

In this article, we will start with a review of existing DNA sequences compression algorithms (Section II). In section III, we will present our approach for DNA sequences compression and explain how it can reduce their sizes. Finally, in section IV, we will illustrate the experimental results and we will draw a comparison of ratio between our algorithm and other existing algorithms.

II. EXISTING DNA SEQUENCES COMPRESSION ALGORITHMS

The compression of DNA sequences is based on text compression algorithms. However, researchers proved that conventional text compression algorithms are not enough for DNA sequences compression and proposed specific compression algorithms. Based on the standard benchmark of DNA sequences data [1] GZIP tool [2] for example has a compression ratio of 2.217 Bit per Base. However, a compression algorithm provides significant results only if the BpB is lower than two because only then the four nucleotides [3] can be represented by two bits.

There are two classes of DNA sequences compression algorithms. The algorithms for DNA compression in horizontal mode and the algorithms for DNA Compression in vertical mode. The first class compresses a single sequence based on its genetic information. For example, Biocompress [4] seeks repetitions and palindromes in a sequence. BIocompress-2[5] uses a Markov model to compress non-repetitive regions of a sequence. By applying these algorithms to the standard benchmark data, the compression ratio is 1.85 BpB for Biocompress and 1.78 BPB for biocompress-2. Therefore, they are better than conventional Lossless compression algorithms since the BpB rate is under two.

Some other DNA sequences compression algorithms are based on the binary representation of the nucleotides (e.g. A = 00, C

Manuscript received January 03, 2016; revised January 28, 2016.

Bacem Saada, Ph.D. Student with Harbin Engineering University, College of Computer Science and Technology, Harbin, China, (email:basssoum@gmail.com).

Jing Zhang, Ph.D. Professor with Harbin Engineering University, College of Computer Science and Technology, Harbin, China, (email: zhangjing@hrbeu.edu.cn).

= 01, G = 10, T = 11). For example, GENBIT [6] divides sequences in blocks of 8 bits each and subsequently makes a 9th bit. If the block is identical to the above, the 9th bit is equal to 1, otherwise to 0. DNABIT [7] divides the sequence into small blocks and compresses them while taking into consideration if they existed previously or not. Saada, B. and Zhang, J reduced the size of the DNA sequence to less than 25% of its initial size by compressing it using the extended-ASCII representation and applying the RLE technique to compress the similar blocks and keep only one block [8].
The second class of DNA sequences compression algorithms analyzes the genetic information of a set of sequences in order that one of them would be representative of the whole set. For instance, DNAZIP package [9] has a series of algorithms that divide a genome into small blocks and compress them. LZ77 [10] proposes a compression technique for several genomes belonging to the same genus. Saada, B. and Zhang, J. use some techniques to convert the DNA sequence to hexadecimal representation and detect regions of similarities between a set of sequences [11]. They also devised an algorithm to detect the longest common chain for a set of sequences of the same genus and use it as representative of the whole set [12].

III. Our Proposed Algorithm

A. Description of the algorithm

Our algorithm is based on the binary representation of nucleotides. First, our algorithm attempts to detect adjacent similar or palindrome blocks and compresses them to a binary representation. Thereafter, to reduce the size of the output sequence, the bits will be converted to Extended ASCII coding which has an 8-bit character code.

B. Presentation of the algorithm

Our algorithm is an extended version of the DNABIT compression algorithm to which we added more compression techniques to better reduce the size of the binary representation of the DNA sequence. We detect similarity in the DNA sequence by using two phases:

- Even Bit technique for non-repetitive blocks,
- Odd Bit techniques for similar blocks of the DNA sequence.

1. Even Bit technique

The four nucleotides {A, C, G, T} will be encoded as follows:

A=00, C=01, G=10, T=11.

Two bits are assigned to each base on non-repetitive block.

2. Odd Bit techniques

In this phase, our algorithm looks for the similar blocks in the DNA sequence. Many techniques are used in this phase depending on the size of the similar detected blocks. Our algorithm uses four techniques:

- 3-Bit Technique,
- 5-Bit Technique,
- 7-Bit Technique,
- 9-Bit Technique.

a) 3-Bit Technique

In the DNA sequence, if there are two or three similar bases which are adjacent, this technique is applied. The first bit is allocated as “0” if the base repeated number is two; this bit is allocated as “1” if the occurrence of the base is three. The two other bits depend on the nucleotide value code (figure1).

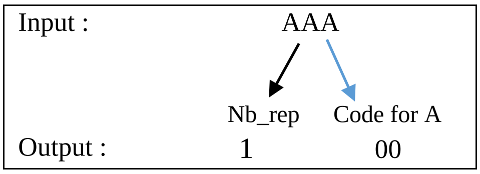

Fig. 1. 3_bit technique

b) 5-Bit Technique

This technique is used if there are more than three and less than eight repetitions of the nucleotide base. The first three bits represent the number of repeated nucleotides. The last two binary digits represent the nucleotide value code (figure 2).

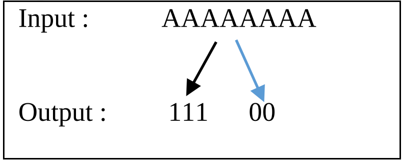

Fig. 2. Exact base 5_bit technique

This technique is also used if there is a reverse similarity of a 2-bits block. The first bit is represented as 0 if there is only one reverse block, and represented as 1 if there are two. The two bases are encoded in the four remaining bits (figure 3).

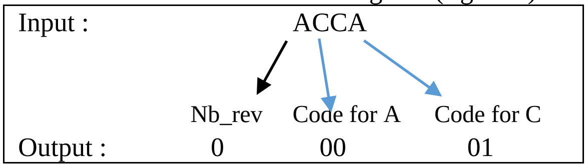

Fig. 3. Reverse 2 bases 5_bit technique

c) 7-Bit Technique

This technique is used if there is a two nucleotides base repeated more than one and less or equal to 8. The first three bits represent the number of repeated nucleotides. The last two binary digits represent the nucleotide value code (figure 4).

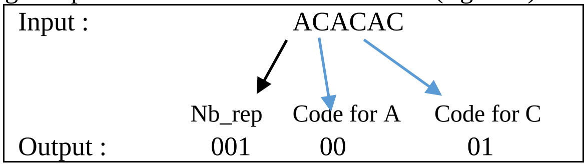

Fig. 4. Reverse 2 bases 7_bit technique

d) 9-Bit Technique

This technique is used if there is a reverse similarity of a 3-bits block. The first 3 bits represent the number of successive palindrome blocks. The three bases are encoded in the six remaining bits (figure 5).

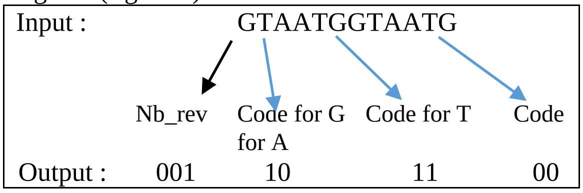

Fig. 5. Exact 3 bases 9_bit technique

This technique is used also if there is a repetition of four nucleotides exact chain or four nucleotides palindrome chain. The first bit will be equal to “0” if the detected chain is an exact repetition (figure 6).

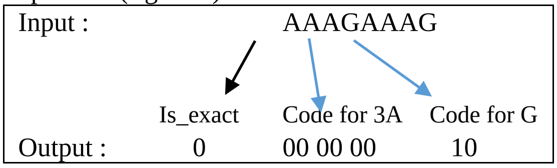

Fig. 6. Exact 4 bases 9_bit technique

The first bit is equal to 1 if the repeated chain is a palindrome (figure 7).

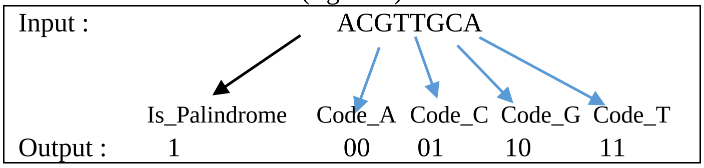

Fig. 7. Reverse 4 bases 9_bits technique

This technique is used also if there are 1 to 8 similarities of the 3-bits block. The first three bits represent the number of repetitions (figure 8).

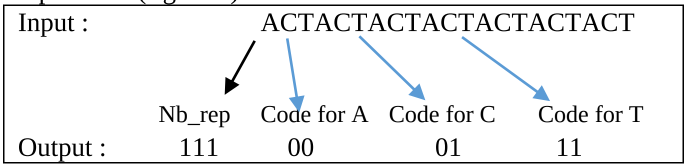

Fig. 8. Exact 3 bases 9_bits technique

3.Additional Structure for storing the type of the used technique.

There are 7 Odd Bit techniques in addition to the Even Bit technique. That’s why we need to use an additional data structure to know which technique is used and how many bits are used for the next nucleotide block. We introduced a data structure (figure 9) based on these techniques as follows:

- 0: Even technique,
- 1:3-Bit Technique, exact 2 repeat bases,
- 2: 5-Bit Technique, exact base,
- 3: 5-Bit Technique, reverse 2-bits block,
- 4: 7-Bit Technique, exact 2-bits block,
- 5: 9-Bit Technique, reverse 3-bits block,
- 6: 9-Bit Technique, exact/reverse 4-bits block,
- 7: 9-Bit Technique, exact 3-bits block.

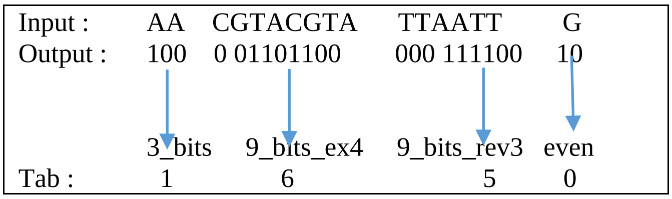

Fig. 9. Additional data structure

4. Compression of the binary output to extended-ASCII representation

To better reduce the size of the data stored in the databases, we will convert the binary representation to an extended-ASCII representation. The benefit from the use of this technique is that one extended-ASCII character encodes 8 binary digits. The output result will be reduced to 12.5% of the initial binary representation (figure 10.).

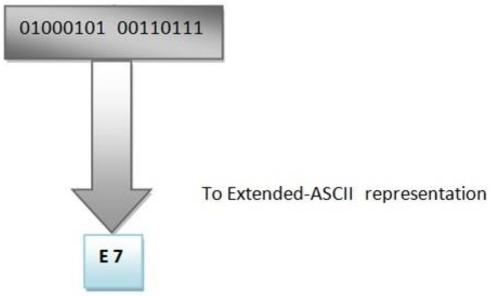

Fig. 10. Conversion to extended-ASCII representation

IV. Experimental Results

A. Evaluation Metrics

To measure the performance of our approach, we use entire genomes in order to calculate the contribution of our algorithm, in terms of compression ratio, to the genomes which have a large number of nucleotides.

B. Performance in terms of data compression

To achieve our experimental study, we used the Human Globin Gene (HUMHBB), the Human Sequence of Contig (HUMHDABCD) the Mitochondrial genome (MPOMTCG) and the Vaccinia Virus genome (VACCG) whose size is about 190000 nucleotides.

As indicated in table I, applying our techniques helped reduce the compression ratio of those genomes. The results demonstrate that most of the existing algorithms have a compression ratio higher than 1.7 BpB. Our algorithm provides better results and has a compression ratio equal to 1.54 BpB for the compression of the genome MPOMTCG.

TABLE I. Comparison with other algorithms

Sequence	Base Pair	GZIP	DNA Compress	DNA Pack	CTW+LZ	DNABIT-2
HUMHBB	73308	2.21	1.79	1.77	1.810	1.6
HUMHDABCD	58864	2.19	1.796	1.74	1.822	1.6
MPOMTCG	186609	2.21	1.90	1.89	1.90	1.54
VACCG	191737	2.17	1.75	1.76	1.76	1.63

By applying the conversion to extended-ASCII representation, the size of the sequence is reduced to less than 11% of its initial size as shown in table II.

TABLE II. PERFORMANCE OF OUR ALGORITHM ON DIFFERENT DNA SEQUENCES AND GENOMES

Sequence Name	Number of Nucleotides	Size after applying the compression techniques	Size after the Extended-ASCII Compression
HUMHBB	73308	58648	7330
HUMHDABCD	58864	47092	5887
MPOMTCG	186609	144621	18078
VACCG	191737	158497	19813

C. Experiments in Time execution

To measure the execution time of our algorithm, we used a computer with an Intel i3-2375M processor cadenced at 1.5 Ghz and a 4GB Ram memory.

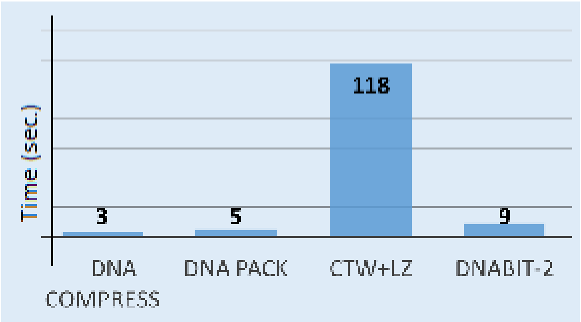

Fig. 11. Execution time comparison between our approach and other algorithms

The previous figure (fig.11) presents the execution time by applying the algorithm on the VACCG genome. It shows that its execution time is less than that of CTW+LZ algorithm and slightly higher than DNA Pack and DNA Compress execution time. To better reduce our algorithm's execution time, it is possible to parallelize its execution.

V. Conclusion and Future Work

The advantage of our algorithm is that it allows to have a compression ratio per base lower than 1.6 BpB thus better than other existing compression algorithms. The algorithm is also easy to implement and interesting to use as the Extended-ASCII representation compresses the initial nucleotide representation to less than 11%

In the future, we will try to associate our algorithm to compression algorithms based on statistical approaches to compress the DNA sequences with a rate higher than the rate of the current existing algorithms.

Acknowledgment

This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

References

[1] S. G rumbach and F. Tahi, "Compression of DNA Sequences," in Proc. of the Data Compression Conf., (DCC '93), 1993, 340–350.

[2] Pierzchala, S. (2004). Compressing Web Content with mod—gzip and mod—deflate. Linux Journal, 1-10.

[3] Matsumoto, T., Sadakane, K., Imai, H., et al., 2000, Can General-Purpose Compression Schemes Really Compress DNA Sequences?, Computational Molecular Biology, Universal Academy Press, 76–77.

[4] Grumbach S. and Tahi F.: Compression of DNA Sequences. In Data compression conference, pp 340-350. IEEE Computer Society Press, 1993.

[5] Korodi, G., Tabus, I., Rissanen, J., et al., 2007, DNA Sequence Compression Based on the normalized maximum likelihood model, Signal Processing Magazine, IEEE, 24(1), 47-53.

[6] Grumbach, S., Tahi, F.: A new Challenge for compression algorithms: genetic sequences. Journal of Information Processing and Management 30, 866–875 (1994).

[7] A.AppaRao, "DNABIT compress-compression of DNA sequences," in Proc. the Bio medical Informatics, 2011.

[8] Saada, B., & Zhang, J. (2015). DNA Sequences Compression Algorithm Based on Extended-ASCII Representation. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2).

[9] Ahmed, S., Brickner, D. G., Light, W. H., Cajigas, I., McDonough, M., Froyshteter, A. B., ... & Brickner, J. H. (2010). DNA zip codes control an ancient mechanism for gene targeting to the nuclear periphery. Nature cell biology, 12(2), 111-118.

[10] Ahmed, S., Brickner, D. G., Light, W. H., Cajigas, I., McDonough, M., Froyshteter, A. B., ... & Brickner, J. H. (2010). DNA zip codes control an ancient mechanism for gene targeting to the nuclear periphery. Nature cell biology, 12(2), 111-118.

[11] Saada, B., & Zhang, J. (2015, November). DNA sequences compression algorithms based on the two bits codation method. In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on (pp. 1684-1686). IEEE.

[12] Saada, B., & Zhang, J. (2015). Vertical DNA Sequences Compression Algorithm Based on Hexadecimal Representation. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2).