Propose a Substitution Model for DNA Data Compression

Ayad E. Korial
Computer Engineering Department University of Technology, Iraq

Ali Kamal Taqi
Computer Engineering Department University of Technology, Iraq

ABSTRACT
One of the most difficult challenges in lossless data compression is finding the right model for the Deoxyribonucleic Acid (DNA) compression. DNA sequences include four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T) where the information in DNA is stored as a code made up of these four chemical bases and these sequences show these are not random, if they are totally random then store them in two bits, this is the most efficient and logical way. This paper proposed an algorithm called A2 for DNA data compression. The proposed algorithm consists of four stages to build a substitutional model. The first stage used a modified version of run-length coding, in second and third stages mapping model for formatting data to be suitable for the final stage fed into Burrows-Wheeler Transform to use permutation technique that group related symbols as possible to improve dictionary coding using Lempel-Ziv (LZ77) and output file stored as (.a2) extension. The A2 algorithm implemented and tested on data from GenBank and shows acceptable file size and processing time ratio.

Keywords
DNA compression, Compression Model, Run-length coding, Mapping Table, BWT, Dictionary Coding, A2 algorithm.

1. INTRODUCTION
Data compression divided into two categories lossy and lossless compression. Lossy data compression allows certain loss of accuracy in exchange for greater compression ratio. This type of compression used in images, audio and videos, while the lossless data compression guaranteed to generate the exact duplicate input data after decompression these. This type of compression used in database record, text files and application files. Building the right model is a critical issue in data compression. In order to reduce the entropy of data, a suitable model should apply to data to be compressed efficiently using entropy coding technique. Molecular sequence databases (e.g. European Molecular Biology

Laboratory (EMBL), Genbank, etc) store hundreds or thousands of DNA sequences reaching up to

thousands of gigabytes and are continuously growing with storage doubling time of 18 months [1].

Due to this large size, they are difficult to download or shared over the network, by anyone except those with a large number of available resources. In such a situation, the data transferred to hard disks from one place to the other, which results in wastage of time and money. Even though one downloads specific data sets, the load on local storage can be very high and such large-scale analysis of these sequences is only possible to those with large computing resources [2].

Lossless compression is a class of data compression algorithms that allows the original data to reconstruct from the compressed data. It can be divided into Entropy encoding, Dictionary encoding, and other types. Entropy encoding compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix-free output code word such as Huffman, Arithmetic, and asymmetric numeral systems coding. Dictionary encoding operates by searching the matches between the text to be compressed and a set of strings contained in a data structure (called the 'dictionary') maintained by the encoder. When the encoder finds such a match, it substitutes a reference to the string's position in the data structure such as Lempel-Ziv-Welch which creates a tokenization by trying to match a current word with a previous occurrence of that same word. LZ77 encodes and decodes from a sliding window over previously seen characters, decompression must always start at the beginning of the input. DEFLATE utilizes LZ and statistical encoding to achieve compression. Zstd combines use of a dictionary-type algorithm (LZ77) and Finite State Entropy (tANS) stage of entropy coding. Brotli is a generic-purpose lossless compression algorithm that compresses data using a combination of a modern variant of the LZ77 algorithm. Huffman coding and 2nd order context modeling, GenCompress, for genetic sequences, based on searching for approximate repeats, Other types such as Run-Length which takes advantage of the adjacent clustering of symbols that occur in succession. It replaces a “run” of symbols with a tuple that contains the symbol and the number of times it is repeated, Delta encoding which encode a series of code as the difference from the previous code [2].

Afify et al. [2] presented a differential compression algorithm based on production of difference sequences according to op-code table. Chen et al. [3] presented a DNA compression algorithm, GenCompress, based on approximate matching.

Another researcher used a simple and statistical replacement method used to represent the word of four symbols at a time with its frequency then encoder generates code word using depending on its frequency distribution [4].

In [5] Constructed a reference sequence based on common sequences of another species reference, this achieved by matching the input with the reference and hence, replacing the match found in input by its fingerprints.

Srinivasa et al. [6] proposed an encoding scheme to compress non-repeat regions of DNA sequences, based on dynamic programming approach.

In addition, Fast Compression introduced an improved run length encoding and delta modulation that has lower computation by improving the running time [7].

More convenient traditional Chinese medicine quality information carrier has been studied. Chen et al. [8] found that the internal transcribed spacer 2 (ITS2) sequence code is more appropriate for identifying medicinal plant species. Liu et al.

[9] studied the conversion of the Chinese herbal medicine DNA sequence into a quick response code (QR code).

Kumar et al. [10] studied the conversion of DNA barcoding into PDF417. Cai et al. [11] studied the conversion of Chinese medicine chemical fingerprints into a QR code. However, the two-dimensional code storage space is very limited; the original data of the normal P. ginseng DNA sequence and chemical fingerprints far exceed the capacity of the current two-dimensional codes. Therefore, compression algorithms for P. ginseng DNA sequence and chemical fingerprint data are very important.

Moreover, Converting Panax ginseng DNA and chemical fingerprints into two-dimensional barcode by converting ginseng ITS2 into bytes then combine chemical fingerprint data and ginseng ITS2 to compress it using Zlib to convert data into QR code [12].

The aim of this paper to propose an algorithm called A2 for DNA data compression. A new compression algorithm presented to compress DNA sequences by using the four stages: run-length, pre-mapping, post-mapping and BWT with LZ77. The proposed algorithm which consists of four stages to build a substitutional model. A2 algorithm implemented and tested on data from GenBank and shows acceptable file size and processing time ratio.

2. PROPOSED ALGORITHM

In this paper, we propose a new algorithm called A2 algorithm. This algorithm work in two phases, the first phase in Encoding phase with four stages, and the second phase is decoding with reverse stages of the first phase. The general A2 block diagram is shown in Figure (1).

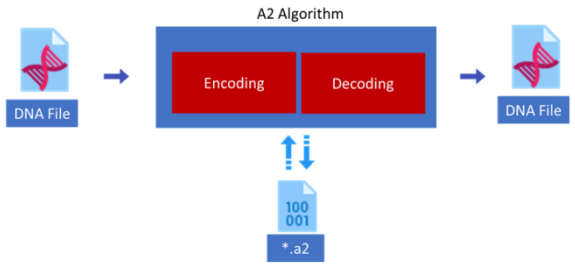

Figure (1): General A2 block diagram

The Encoding phase: this part contains stages, run-length, pre-mapping, post-mapping and BWT with LZ77. The encoding phase shown in Figure (2).

Figure (2): Encoding phase

In the beginning, a Preprocess filtering is applied using a window to extract gene (A, T, C, G) from the file, this accomplished by removing all metadata associated from the sequence file from GenBank database.

In the first stage, creating a modified version of the run length method. A modified version of run-length of each repeated character with repetition length more than or equal to three can be written in the form of "X#" where X is one of DNA character that mean either A, C, G or T, and # represents the number of repetition of that character. In the case of character with repetition length equal to two the next stage will resolve it.

Pre-mapping is the stage two, at this stage resolve the remaining part of stage one, for repetition length two, we use Table (1) for this mapping:

The following scenario considered in stage three. Firstly, when two symbols repeat as the duplicate of the previous two, the (+) character used to represent the two characters, secondly when the duplicate is the inverse of the first scenario repeats used the first scenario duplicate with (-) character. lastly by conclusion of the previous stages and scenarios few probabilities remain to express other sequences such that (ATA) would have either (G) or (C) as prefix so that one character is used to represent it (#) and using inverse character (!) to represent the inverted sequences.

The reason behind using these substitution characters with inverse technique and its order is to improve the representation of DNA sequence data to be more suitable for the entropy and dictionary coding in the final stage. The Post-Mapping table applied in order as shown in the Table (2).

After applying substitution technique in previous stages, a permutation technique applied using Burrows-Wheeler Transform (BWT) as the final stage. The BWT used to group correlated characters as possible where this stage consist of total of 17 characters (A T C G ( ) [ ] + - # @ $ % = & !) and ten possible digit ranging from 0 to 9 as the output of run length, then the output of BWT used in dictionary coding LZ77 to construct the final output file with extension of (.a2). Pseudocode for A2 compression steps shown below:

A2 Compression Algorithm
Input: DNA Sequence File **Output:** DNA Compressed File
Begin: Read Input File Apply Filter Window Genes Data Extraction **While** (Not reach to EOF): process read file with run-length Apply Pre-Mapping Substitutional Model Apply Post-Mapping Substitutional Model Apply BWT+LZW Permutation Technique **End**

The Decoding phase: this part contains a reverse of the four stages in the first phase. The-decoding phase shown in Figure (3).

Figure (3): The Decoding phase

Pseudocode for A2 decompression steps shown below:

A2 Decompression Algorithm
Input: DNA Compressed File **Output:** DNA Sequence File
Begin: Read Input File Reverse BWT+LZW Permutation Technique **While** (Not reach to EOF): Reverse Post-Mapping Substitutional Reverse Pre-Mapping Substitutional Model Reverse file with run-length Model **End**

Table (1): Pre-Mapping

Sequence	Coding
AA	)
CC	(
GG	]
TT	[

Table (2): Post-Mapping

Sequence	Coding	Sequence	Coding	Sequence	Coding	Sequence	Coding
ATAT	AT+	TATA	AT-	TAT	#	TGT	%
ACAC	AC+	CACA	AC-	ATA	#!	GTG	%!
AGAG	AG+	GAGA	AG-	GCG	@	AGA	=
TCTC	TC+	CTCT	TC-	CGC	@!	GAG	=!
TGTG	TG+	GTGT	TG-	ACA	$	CTC	&
CGCG	CG+	GCGC	CG-	CAC	$!	TCT	&!

3. IMPLEMENTATION

In order to implement the compressing stages, the following DNA sequence was considered as example:

ACCTCAAAAAAATGGGGTTCTCGAAGTGTACAGAGA
TCGAGGCAACCAGCCCAATAGGTATAGTGTCAC

Firstly, apply the modified Run-length coder. The only character with repetition length more than or equal to three was coded, other repetitions of two character remain same to be processed in the next stage.

ACCTCA7TG4TTCTCGAAGTGTACAGAGATCGA
GGCAACCAGC3AATAGGTATAGTGTCAC

Secondly, using the Pre-Mapping table to process the remaining double repetition of the run-length stage:

A (TCA7TG4 [CTCG) GTGTACAGAGATCGA] C) (AGC3) TA] TATAGTGTCAC

Thirdly, using the Post-Mapping table to substitute the duplicate, inverse duplicate, and other possible correlation, the mapping applied in sequential order:

A (TCA7TG4 [& G) TG - $ G + ATCGA] C) (AGC3) TA] AT-TG- $!

The Run-length coding and mapping process is shown in the Figure (4):

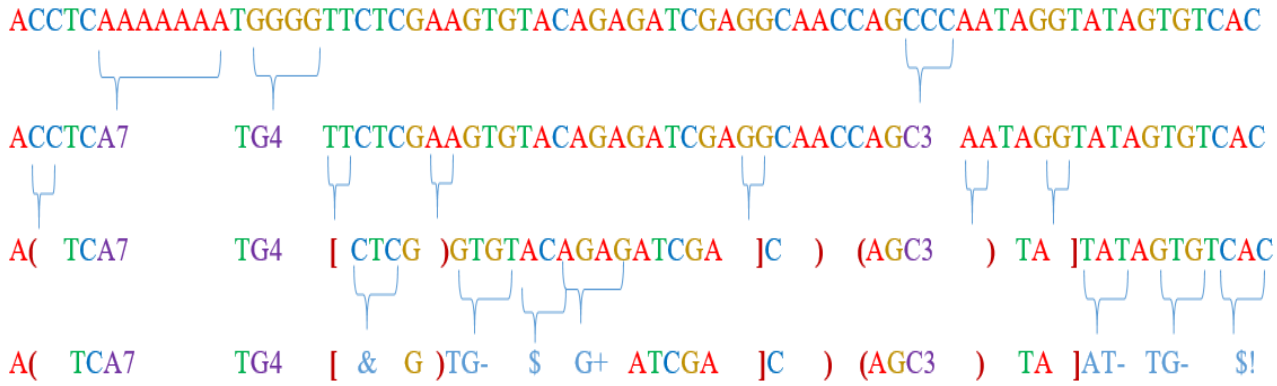

Figure (4): Run-length and Mapping process

The output of mapping then fed into Burrows-Wheeler Transform used permutation technique, which group related symbols as possible to improve dictionary coding using LZ77 and output file stored as (.a2) extension.

We design and develop an application that supports both windows architectures (x86 and x64) and taking into accounts multi-thread processing. To compress a file, firstly load a file in the application in "LOAD FILE" section, the application read the file (DNA DATA) and calculate the original file size.

Secondly, manipulate the file by performing a part of the proposed algorithm starting from run-length, pre-mapping and post-mapping by clicking the "RUN" button. Finally, is the coding process to generate the compressed version of DNA file with extension (a2).

To decompress a file directly click decompress button then select the file that has (a2) extension then the decompression started directly with reverse procedure of the compressing automatically

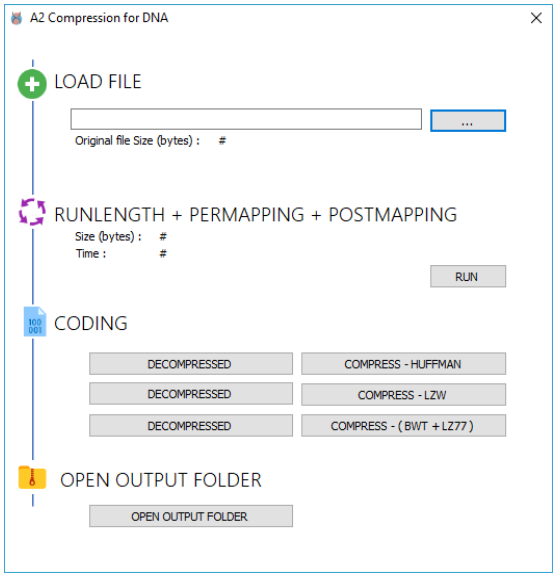

Figure (5): A2 Application

4. RESULTS

GenBank database used to test the proposed algorithm with 20 sequence files [13] after extract the DNA sequences and removing metadata, the following table includes the general-purpose lossless compression compared with the proposed A2 algorithm as shown in Table (3).

In order to compare with special-purpose algorithm for DNA compression, GenCompress algorithm [14] is used.after applying the compression program of the GenCompress algorithm to the files table above, the program shows inefficient compression time and throw an exception, in order to use the program, the following sequences with small size is used as shown in Table (4).

The table shows better compression ratio of GenCompress only with small size sequences while any file more than (2MB) compression ratio drop down and consumption time increased extensionally as shown in Table (5).

Figure (6) shows compression size comparison and Figure (7) shows compression ratio comparison

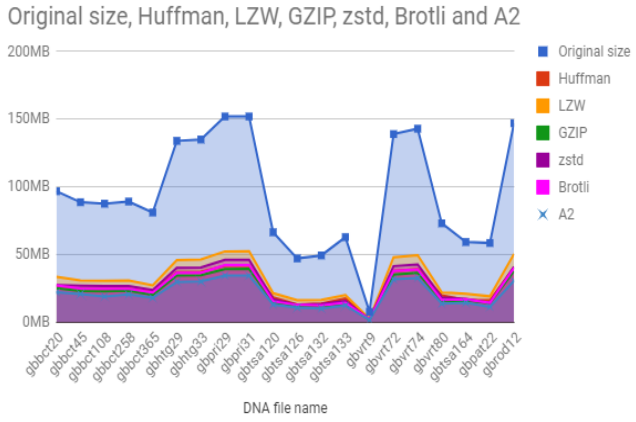

Figure (6): Compression size comparison

The following equations used in order to calculate the compression ratio:

$$CR = \frac{Uncompressed\ \ Size}{Compressed\ \ Size} \quad \ldots (1)$$

Where $CR$is compression ratio, Figure (7) shows compression ratio comparison

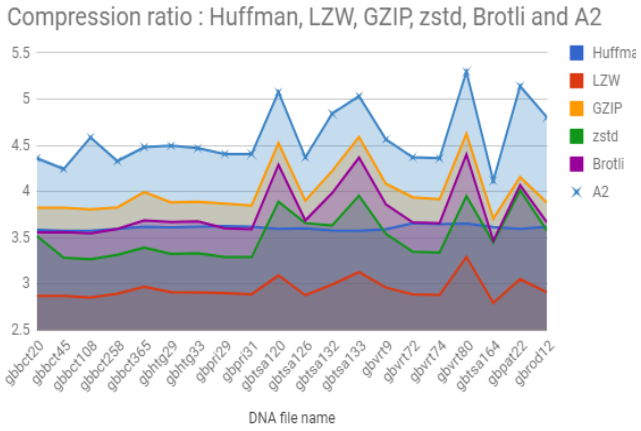

Figure (7): Compression ratio comparison

The saving space calculated by the following equation:

$$SS = 1 - \left(\frac{1}{CR}\right) \qquad (2)$$

Where $SS$is saving space, Figure (8) shows saving space comparison.

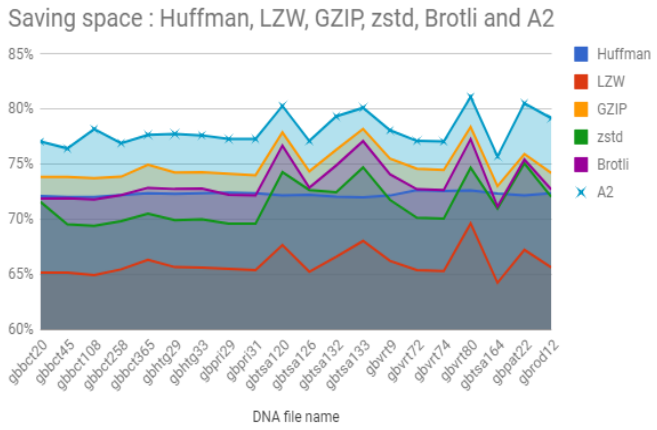

Figure (8): Saving space comparison

Table (3): Comparison with general purpose algorithms

No.	DNA file name	Original size (MB)	Huffman	LZW	GZIP	zstd	Brotli	A2 proposed
1.	gbbct20	96.8	27.0	33.7	25.3	27.5	27.2	22.2
2.	gbbct45	88.7	24.8	30.9	23.2	27.0	24.9	20.9
3.	gbbct108	87.6	24.5	30.7	23.0	26.8	24.7	19.1
4.	gbbct258	89.2	24.8	30.8	23.3	26.9	24.8	20.6
5.	gbbct365	81.1	22.4	27.3	20.3	23.9	22.0	18.1
6.	gbhtg29	134.0	37.1	46.0	34.5	40.3	36.5	29.8
7.	gbhtg33	135.0	37.3	46.4	34.7	40.5	36.7	30.2
8.	gbpri29	152	41.9	52.4	39.3	46.2	42.2	34.5
9.	gbpri31	152	42.0	52.6	39.5	46.2	42.3	34.5
.10	gbtsa120	66.5	18.5	21.5	14.7	17.1	15.5	13.1
11.	gbtsa126	47.2	13.1	16.4	12.1	12.9	12.8	10.8
12.	gbtsa132	49.4	13.8	16.5	11.7	13.6	12.4	10.2
13.	gbtsa133	62.9	17.6	20.1	13.7	15.9	14.4	12.5
14.	gbvrt9	7.76	2.16	2.62	1.90	2.19	2.01	1.70
15.	gbvrt72	139	38.0	48.1	35.3	41.5	37.9	31.8
16.	gbvrt74	143	39.2	49.6	36.5	42.8	39.1	32.8
17.	gbvrt80	73.1	20.0	22.2	15.8	18.5	16.6	13.8

18.	gbtsa164	59.3	16.4	21.2	16.0	17.2	17.1	14.4
19.	gbpat22	58.6	16.3	19.2	14.1	14.6	14.4	11.4
20.	gbrod12	147	40.6	50.5	37.9	41.1	40.1	30.6

Table (4): Comparison with special purpose Algorithms

Sequence Name	Original size (KB)	GenCompress	A2
HUMHBB	59.6	14.1	15.8
HUMHDABCD	47.9	11.2	12.7
HUMDYSTROP	31.5	7.60	9.05
HUMHPRTB	62.9	14.5	16.3
HUMGHCSA	54.2	10.8	12.2

Table (5): Algorithms speed calculation

File Name	Size (MB)	Time Run+Pre +Post (min:sec.ms)	Time BWT + LZ77 (sec)	Total Time (sec)	Speed
gbbct45	88.7	16.32	180	196	0.453 MB/sec
gbbct108	87.6	15.26	180	195	0.455 MB/sec
gbvrt9	7.76	1.40	15.859	17	0.456 MB/sec
gbpri31	152	29.04	223.734	253	0.60 MB/sec
AVG =					0.491 MB/sec

5. CONCLUSION

The need for effective DNA data compression is reasonable in biological applications where network bandwidth (storage and transmission of DNA sequences) are involved. The proposed algorithm builds a substitutional model and uses permutation technique and tested on data from GenBank. The compression results in comparing with different general algorithms show (A2) is more efficient than others general purpose compression algorithm in both compression ratio and saving space. Moreover, in comparing with GenCompress, the output shows faster and more effective compression results on large files with long DNA sequences.

6. REFERENCES

[1] Priyanka Ms., Goel S., "A Compression Algorithm for DNA that uses ASCII Values", in Advance Computing Conference (IACC), 21-22 Feb. 2014 IEEE International, Gurgaon.

[2] Afify H., Islam M., Wahed MA. "DNA Lossless Differnational Compression Algorithm Based on Similarity of Genomic Sequence Database", International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 4, August 2011.

[3] Xin Chen, Sam Kwong, Ming Li, "A Compression Algorithm for DNA Sequences Using Approximate Matching for Better Compression Ratio to Reveal the True Characteristics of DNA", IEEE engineering in medicine and biology, July/August 2001.

[4] Ashutosh Gupta, Kamlesh Kumar Dubey, "Efficien Compressor for Biological Sequences", IEEE International Advance Computing Conference (IACC), 2013.

[5] Kanika Mehta, Satya Prakash Ghrera, "DNA Compression using Referential Compression Algorithm", IEEE, 2015.

[6] Srinivasa K. G, Jagadish M., Venugopal K. R., Patna L. M., "Efficient Compression of non-repetitive DNA sequences using Dynamic Programming", IEEE, 2006.

[7] Ouyang J., Feng P. and Kang J., "Fast Compression of Huge DNA Sequence Data", International Conference on BioMedical Engineering and Informatics, 2012.

[8] Chen SL., Yao H., Han JP., Liu C., Song JY., Shi LC., Zhu YJ., Ma XY., Gao T., Pang XH., et al., "Validation Of The ITS2 Region as a Novel DNA Barcode for Identifying Medicinal Plant Species", . PLoS One 2010; 5:e8613.

[9] Liu C, Shi L, Xu X, Li H, et al., "DNA Barcode Goes Two-Dimensions: DNA QR Code Web Server", PloS One 2012;7: e35146.

[10] Kumar NP., Rajavel A., Jambulingam P., "Application of PDF417 Symbology for 'DNA Barcoding", . Comput Meth Prog Biomed 2008; 90:187e9.

[11] Cai Y, Li XW, Li M, Chen XJ, Hu H, Ni JY, Wang YT. Traceability and quality control in traditional Chinese medicine: from chemical fingerprint to twodimensional barcode. Evid Based Complement Altern Med 2015, 251304. http://dx.doi.org/10.1155/2015/251304. 6 pages, 2015.

[12] Cai Y, et al., "Converting Panax ginseng DNA and Chemical Fingerprints into Two-Dimensional Barcode", Journal of Ginseng Research, 41(3):339-46, 2017.

[13] "FTP Access to GenBank Data." National Center for Biotechnology Information, U.S. National Library of Medicine, Aug. 2017, www.ncbi.nlm.nih.gov/genbank/ftp/.

[14] Chen, Xin, et al. "Welcome to GenCompress!" GenCompress Home Page, www.cs.cityu.edu.hk/~cssamk/gencomp/GenCompress1.htm.