Data Compression Strategy for Reference-Free Sequencing FASTQ Data

Hsu Mon Lei Aung, Swe Zin Hlaing
University of Information Technology, Yangon, Myanmar
hsumonleiaung@uit.edu.mm, swezin@uit.edu.mm

Abstract

Today, Next Generation Sequencing (NGS) technologies play a vital role for many research fields such as medicine, microbiology and agriculture, etc. The huge amount of these genomic sequencing data produced is growing exponentially. These data storages, processing and transmission becomes the most important challenges. Data compression seems to be a suitable solution to overcome these challenges. This paper proposes a lossless data compression strategy to process reference-free raw sequencing data in FASTQ format. The proposed system splits the input file into block files and creates a dynamic dictionary for reads. Afterwards, the transformed read sequences and dictionary are compressed by using appropriate lossless compression method. The performance of the proposed system was compared with existing state-of-art compression algorithms for three sample data sets. The proposed system provides up to 3% compression ratio of other compression algorithms.

Keywords- Genomic Sequencing data, lossless compression, reference-free sequence, reference-based sequence.

1. Introduction

The utilization of next generation sequencing data has greatly increased on genomics analysis, hereditary disease diagnosis, therapy and food security, etc. The rate of storage space capacity, processing and data transferring of large genomic sequencing data are rapidly grown up. It becomes a challenge on the storage of sequencing data. Without any efficient compression, the storage and transmit size of sequences will be large.

The compression of genomic sequences is divided into two main categories: reference-free compression methods and reference-based methods [6]. The main idea of reference-free sequence compression is exploiting the structural and statistical properties of data and storing the sequencing reads with specific compressive encoding schema.

Reference-based compression is exploiting the similarity between a target sequence and a reference sequence. The target is aligned (mapped) to the reference. The reference-based methods do not encode the original read data but the mismatches between these sequences are encoded.

Lossless data compression is used when the original data source files are so important that cannot accept any loss in details. The popular general-purpose compression algorithms are bzip2 and gzip. The bzip2 is an open-source file compression program that uses the standard Burrows– Wheeler (BWT) algorithm. It performs block-based sorting for text transformation. The transformation is reversible, without needing to store any additional data [5]. The gzip is based on deflate algorithm, which is a combination of LZ77 and Huffman's coding. However, the general-purpose algorithms cannot compress the biological sequencing data well because they are not taken into account the characteristics of DNA data. The dictionary-based method [3] is substituting the words by indices that relate to the dictionary of words [8].

Although several specific sequencing data compression methods have been proposed, they have many trade-off of the compression performance parameters and some methods have limitation about type of genome sequences. And the availability of assembled reference sequences for all the organisms are difficult.

There are many different file formats, such as FASTA, Multi-FASTA, FASTQ and SAM/BAM to store biological sequencing data. FASTQ, modified version of traditional FASTA file, is the defacto standard for storing data from next generation sequencing platform. It is a text-based format which represents biological sequences and their corresponding quality scores with ASCII characters. This information is stored in the form of read blocks. Example of a read block is shown in Figure 1.

```
@SRR1063349.15409 15409 length=202
TCCGCCTGCAACCTTATGACGTGGTGTATGTCACCACCGCCCCGGTTTCCCG
CTGGAACCGTCTGATCAATCAGTTGCTGCCAACTATTAGCGGTGTTGCACGGC
CCGACGCGATACTGGTAATTCGCGATCTCACTTTCCAGCGTCATATTGGGGCG
CGCTACGTTGGGTCGTGGGCGTAATTGATCAATCAGGCGCGGGG
+SRR1063349.15409 15409 length=202
CCCFFFFFHHHHHJJJEHHIJGHECGHIIJJJIJJIJJIIJJJBGBEHGHHHFFDCEDDDD
DDDDDDDCCCD>C4>ACDCDDCDDDDDECEDDD5>5<<>4@@@FFFFFHHGH
HJIJJJEDHIIJJJIIIJIGJGJJJJHHCHHF?CBECF>ACDDDBDDDDBD<8?BBD?AB@
BB>BD>BBDEDDEDEDCDDCCDDDD@BB
```

Figure 1. Example of a read block in FASTQ

ISBN 978-99971-0-381-9 © 2017 ICAIT

A read block contains four parts. These are sequence identifier, raw read sequence, description field (optional) and quality scores. A description field is also called second header line that starts with "+" .The information is the same as the identifier, but it can also be blank. So, this field can be omitted.

The remainder of the paper is organized as follows: Section II reviews the methods proposed for the compression of biological sequences in FASTQ file format. Section III describes the proposed method in detail. Section IV presents the performance results for the proposed method. Finally, Section V describes conclusions and directions for further work.

2. Related Work

Fqzcomp and Fastqz methods [1] are proposed to compress the files in FASTQ format. The first method uses a byte-wise arithmetic coder and context models. The second method breaks the input file into three separate streams and uses libzpaq compression library for specifying the context models in ZPAQ format based on PAQ, which combines the bit-wise predictions of several context models. The performance trade-off of two methods depends on the use case.

The XM (eXpert Model) method [2] compresses eachsymbol by estimating the probability distribution based on previous symbols information. After the symbol's probability distribution is determined, it is encoded using arithmetic coding. The method maintains three types of expert models such as Markov expert, Context Markov expert and Repeat expert that can provide a probability distribution of symbols and considers the next symbols. The algorithm's compression speed is very slow when it is applied on large genome data sets.

In [4], the DNACompact algorithm is encoded by word-based tagged code. It is a two phases-compression method. The first phase is transformation that transformed four symbol space into a three symbol space. The second phrase is encoding scheme that encoded by WBTC.

The DNA-COMPACT [7] is a two-pass lossless DNA compression based on a pattern-aware contextual modeling technique.

In the first pass, there are two schemes for with and without reference sequences. In the second pass, non-sequential contextual models are used. For the non-aligned sequences, this method is not suitable to get good compression performance.

In [9], FASTQ compression method based on concept of prediction by partial matching (PPM) is proposed. This method develops context based models customized to each FASTQ field. The models are coupled with an adaptive arithmetic coder (AAC) to compress data. The proposed method gains in overall storage space on a sample dataset. The limitation of the proposed method is needed to get faster performance.

DMcompress method [10] firstly analyzes the raw sequence data by calculating first order entropy and then determines the Markov model orders according to the Laplace estimator. Finally, the data is compressed by arithmetic coding. It can get effective compression performance for the latest bacteria genome sequences with around 0.02 bps reduction.

Lossless light-weight reference-based compression algorithm, LW-FQZip [11], identifies and eliminates redundancy information independently from other three components of a FASTQ file. Then incremental method is applied for identifier and run-length limited encoding method is used to encode the quality scores. And then, a light-weight mapping model is used which maps them against external reference sequences to encode short reads. Finally, all the processed data streams are packed together by applying LZMA algorithm. LW-FQZip can get optimal compression ratio, but its processing time and memory utilization are very high.

In this paper, a new lossless compression strategy of biological sequences without reference sequences for FASTQ data is proposed. The proposed method can create dynamic dictionary and preprocessed sequences by finding matching symbols in sequence itself. For unmatched symbols, the appropriate compression algorithm is applied. The compressed files are decompressed and transformed into original files by using reversible transformation and compression.

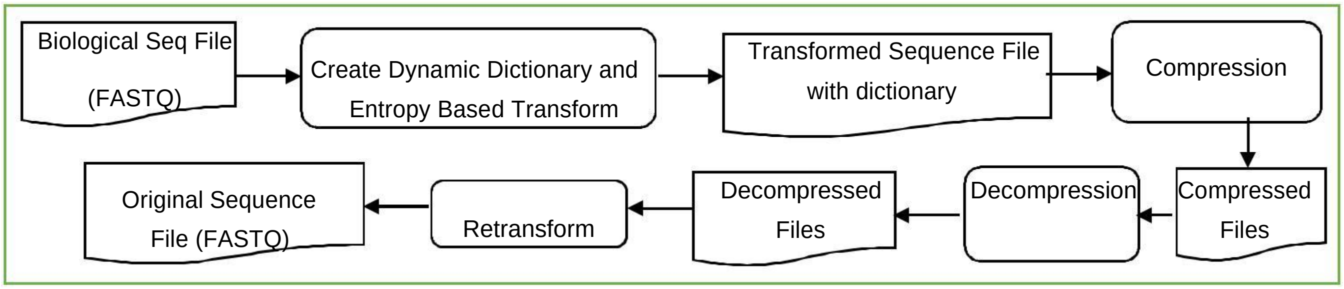

Figure 2. Overview of proposed system design

3. Proposed System

The overview design of the proposed system is shown in Figure 2. Firstly, input file is transformed and created dictionary based on entropy calculation. Secondly, the transformed sequence is compressed by BWCA that is a combination of BWT with compression techniques Move-To-Front transform (MTF), Run-Length Encoding (RLE) and the Huffman Coding.

The flow of a compression process is shown in Figure 3.Firstly, the proposed system splits the FASTQ file into two files. These files will process in parallel. Each read block contains four lines. The third line can be omitted for compression. The three lines (Sequence Identifier, Read Sequences, Quality Score) can be compressed.

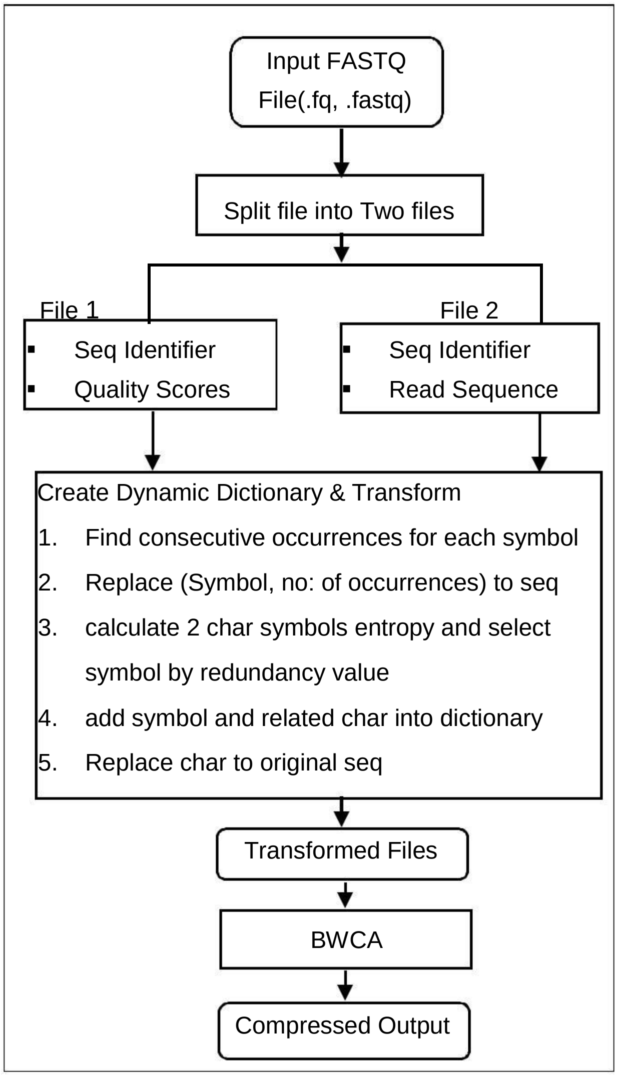
Input FASTQ
File(.fq, .fastq)
Split file into Two files
File 1
File 2
▪ Seq Identifier
▪ Quality Scores
▪ Seq Identifier
▪ Read Sequence
Create Dynamic Dictionary & Transform
1. Find consecutive occurrences for each symbol
2. Replace (Symbol, no: of occurrences) to seq
3. calculate 2 char symbols entropy and select symbol by redundancy value
4. add symbol and related char into dictionary
5. Replace char to original seq
Transformed Files
BWCA
Compressed Output

Figure 3. Flow diagram of compression process

Secondly, the dictionary and transformed sequences are created. In step (1) and (2), the system counts the consecutive occurrences for each symbol that include one character and replaces the (symbols, occurrences) pair into original sequence. Step (3) calculates entropy of symbols including two characters. The entropy calculation equations are shown as follows:

$$H(P) = -\sum_{j=1}^{n} p_j \log p_j \qquad (1)$$

In equation 1, H (P) is the smallest number of bits required to represent the symbols in file. (Pj) represents the probability of each symbol.

The proposed system calculates redundancy by using equation (2). H_{max} (P) is the maximum entropy for same number of states.

$$R = 1 - \frac{H(P)}{H_{m}(P)} \qquad (2)$$

According to the experiments, the numbers of permutations of four symbols taken 2 at a time always have highest redundancy. So, the proposed system, firstly creates two characters symbols dictionary. And then calculate other symbols redundancy. If the value is zero or less than 0.6, it can be added that symbol to dictionary. In step (5), the system replaces the symbols into sequences with char from dictionary. Next the transformed sequences are compressed by using BWCA.

The decompression is the reverse process of compression. In decompression part, the compressed file is firstly decompressed using BWCA and then retransformed it into original file using dictionary.

4. Experimental Result

In this section, two real-world FASTQ data sets and one sample data set are used to test the performance of the proposed system. All file sizes are shown by MB.

Table 1 shows the original file sizes and compressed file sizes of the proposed method that combines with lossless data compression algorithms and popular compression tools such as bzip2 and zip.

Table 1. Comparison of compressed files size

Data Sets	File Size (MB)	bzip2 (MB)	zip (MB)	Propose d +BWCA (MB)	Proposed +Arithmetic Coding (MB)
SRR445780_1.f astq	29.3	5.19	5.94	4.7	10.5
ERR016352_1.f astq	37.6	12.5	14.4	11.2	16.3
SRR1063349.f a stq	30	7	8	6.2	11.4

In Figure 4, the average compression ratios of four methods are compared. The combination of proposed method and BWCA gets the higher ratio than other methods.

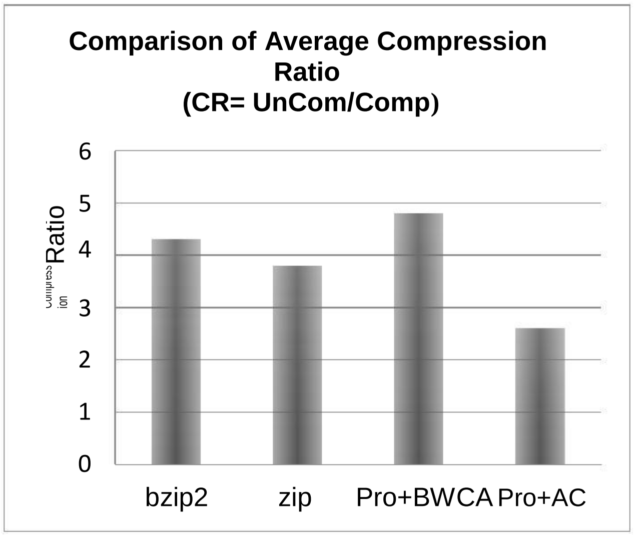

Figure 4. Comparison of average compression ratio

Figure 5 shows the average space saving percentage of four methods. The method (proposed plus BWCA) can save 78% of space, and bzip2 can get 75%. So, the combination of proposed system and BWCA can provide more file compression results.

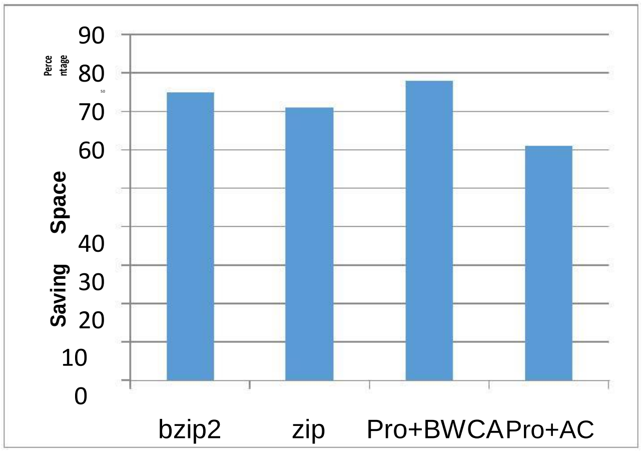

Figure 5. Average Space Saving Percentage of four methods

5. Conclusion

There are many challenges in biological sequence processing and compression. Many different data compression methods have advantages and disadvantage. Some are time-consuming to process but accurate; others are simple to compute but less powerful. Transformation can get better data form before data compression. The proposed system will provide the efficient compression ratio for biological data sequences in FASTQ file format. In future, the proposed method will modify the performance in other sequence file formats. And it also will improve the processing time performance of the proposed system.

6. References

[1] Bonfield, J.; Mahoney M, "Compression of FASTQ and SAM format sequencing data", PLoS ONE, 2013.

[2] Cao, M.; Dix, T.; Allison, L.; Mears, C, "A simple statistical algorithm for biological sequence compression", Data Compression Conference, Snowbird, UT, USA, 2007.

[3] D. Bhattacharya, S. Chakraborty, P. Roy, A. Kairi,"An Advanced Dictionary Based Lossless Compression Technique for English Text Data",CIIT , 2015.

[4] Gupta, A.; Agarwal, S, "A novel approach for compressing DNA sequences using semi-statistical compressor", International Journal of Computers and Applications, 2011.

[5] Hsu Mon Lei Aung, Aye Sandar Win, Swe Zin Hlaing, "Data Transformation for Textual Unstructured Data Compression ", ICCA, 2017.

[6] M. Hosseini, D. Pratas and Armando J. Pinho, "A Survey on Data Compression Methods for Biological Sequences",Information, 2016.

[7] P. Li, S.Wang, J. Kim, H. Xiong, L. Ohno-Machado, and X. Jiang, "DNA-COMPACT:DNA Compression Based on a Pattern-Aware Contextual Modeling Technique," PlosOne, 2013.

[8] R.R.Baruah, V.Deka, M.P. Bhuyan,"Enhancing Dictionary Based Preprocessing for Better Text Compression", IJCTT, 2014.

A. R. Srikanth Mallavarapu, P. Kumar Chinnamalliah., and Ajit S. Bopardikar, "Context based compression of FASTQ data", IEEE, ISCAS, 2016.

B. R. Wang, M. Teng, Y. Bai, "DMcompress: dynamic Markov models for bacterial genome compression", Bioinformatics and Biomedicine BIBM, IEEE, 2016.

C. Y. Zhang. L. Li, Y. Yang, X. Yang, S. He and Z. Zhu,"Light -weight reference-based compression of FASTQ data", BMC Bioinform. 2015.