Journal of The Gujarat Research Society

ISSN: 0374-8588
Volume 21 Issue17,December 2019

COMPRESSION OF GENOMIC SEQUENCES BASED ON BURROWS-WHEELER TRANSFORM: A SURVEY

S.Ranjitha[1], Dr. L.Robert[2]

[1,2]Department of Computer Science, Governments Arts College, Coimbatore, Tamil Nadu, India

Abstract: *With increasingly complete genomes becoming available and the completion of the human genome in the horizon, fundamental questions have been increased regarding the characteristics of these sequences. In this paper, one of the basic questions such as the compressibility of DNA or genomic sequences is discussed since the compression of DNA sequences is a very complex process. The primary objective of this study is investigating the approaches to genomic sequence data compression based on the Burrows-Wheeler Transform (BWT) algorithm. The BWT is an essential data structure of genome indexing which has several fundamental applications, but it is still non-trivial to constrict BWT for the huge collection of genomes. Therefore, this paper highlights some researches related to the compression of genomic or DNA sequence using BWT algorithm and is discussed briefly. In addition, a comparative analysis is carried out to observe the issues in those methods and proposed a modification in BWT to increase the further compression ratio while compressing the genomic sequence data.*

Keywords*- Genome, Genomic sequences, Genomic sequence compression, BWT (Burrow Wheel Transformation), Genome Indexing*

I. INTRODUCTION

The accessibility of the draft sequence of the entire human genome and the entire sequencing of the genome for different model organisms represent a significant objective in molecular biology. It becomes possible with the entire genome for performing genome-wide analysis of whole genomes and cross-genome analysis with entire genomes which could be significant in absolute identification of the genes related to the genetic disorders and in the detection of potential drugs to conflict them. The human genome contains about 3.1647 billion deoxyribonucleic acids (DNA) base pairs. Other mammalian genomes are also of a similar order of magnitude. Therefore, along with the accessibility of entire genomes becomes the exponential growth in the volume of the available biological sequence data. It is approximated that the number of available nucleotide bases doubles approximately every 14 months when genomes are being sequenced at a rate of 15 complete genomes per month.

Nowadays, the most vital challenge is how to analyze this huge amount of data. Normally, the sequences with tens of thousands of base pairs are considered with single gene sequences whereas, with complete genomes, millions or billions of base pairs are required [1]. Also, efficient and effective algorithms are needed for analysis, annotation, interpretation and visualization of this unprecedented mass of biological sequence data. If the information in DNA sequences completely random, then two bits must require coding each nucleotide base pair. Biological sequences are however known to convey significant information between different generations of organisms. From the perspective of compression and sequence understanding, the repetitions inherent in biological sequences imply redundancies which can provide an opportunity for vital compaction. The prediction of such dependencies is the starting point for the biological sequence compression. There are many lossless and lossy compression schemes have been proposed to compress the genomic sequence data during the past few years. Most of the researches on the compression of genomic sequences have been encouraged by the notion that the compressibility of a DNA/genomic sequence could serve as a measure of its information content and thus as a tool for sequence analysis. The outcome of a sequencing experiment typically comprises a large number of short sequences called reads including metadata related to each read and a quality score that estimates the confidence of each base.

The main aim of this article is to review the researches associated with the lossless compression scheme such as BWT algorithm-based genomic sequence data compression. Since the BWT is the basis of many algorithms for compression and indexing of data [2]. However, the cost of computing the BWT of very large string collections has prevented these techniques from being widely applied to the large sets of sequences often encountered as the outcome of DNA sequencing. Thus, an overview of different researches related to the compression of genomic/DNA sequence data using BWT algorithm is discussed briefly. In addition, a comparative analysis is carried out to observe the issues in those methods and proposed a modification in BWT to increase the further compression ratio while compressing the genomic sequence data.

The rest of the article is organized as follows: Section II presents the previous researches related to the genomic sequences compression based on BWT. Section III illustrates the comparative analysis of those methods and Section IV concludes an entire discussion and suggests future enhancement.

II. Survey On Genomic Sequences Compression Based On Burrows-Wheeler Transform

Yang et al. [3] proposed the Burrows-Wheeler similarity distribution between biological sequences based on BWT. The main aim of this method was proposing Burrows-Wheeler similarity distribution of two sequences with distance measures to compare two sequences. Few distance measures were normally followed by the distribution. The expectation and entropy of the similarity distribution were used for constructing the phylogenetic trees on two

independent datasets. Cox et al. [4] proposed a large-scale compression of genomic sequence databases with the BWT. In this algorithm, the BWT of human genome-scale data was computed and the redundancy present in the large-scale genomic sequence datasets was enabled by generic second-stage compressors such as bzip2 and 7-zip. The effect of genome coverage, sequencing error and read length on the level of compression was investigated. Also, the effects of reads trimming and selections of second-stage compressors on the level of compression were discussed.

Xin et al. [5] proposed a parallel architecture for DNA sequence inexact matching with BWT. In this method, hardware architecture was presented for BWT-based inexact sequence mapping algorithm by using the FPGA. It can handle two errors including mismatches and gaps. The original recursive algorithm implementation was dealt with using hierarchical tables and then parallelized to a large extension via a dual-base extension method. Bauer et al. [6] proposed a lightweight algorithm to construct and invert the BWT of string collections in the field of DNA sequencing. This algorithm can reduce the memory to process the number of strings and make an additional utilization of external memory for achieving RAM usage that is constant with respect the number of strings and negligible in size for a small alphabet such as DNA. Initially, two algorithms were presented to recover the strings in a collection of its BWT. Then, the BWT of the original collection was updated if the sequences were added or removed from the collection for obtaining the BWT of the revised collection.

Prochazka & Holub [7] proposed a BIO-FMI compression method based on wavelet tree FM-index optimized for compressing a set of similar biological sequences. This method was proposed based on tracking single changes between every single sequence and the selected reference sequence. The primary occurrences and secondary occurrences were distinguished. The length of the chunks of the searched pattern was the parameter of this proposed self-index that tunes the trade-off between the space of the stored self-index and the time required to locate the occurrences. Further, this method can exploit the knowledge of alignments of single sequences and construct the self-index in an extremely short duration.

Li et al. [8] proposed a BWT-based method for DNA sequence comparison. Initially, BWT algorithm was introduced based on the linear and circular permutation. After that, matrix representations were constructed for a DNA sequence by means of a subtraction between the sequence and its BWT sequence to characterize the DNA sequence by a 24-D vector whose entries are the spectral norms of these matrices.Kimura & Koike [9] analyzed the genomic rearrangements by using the BWT of short-read (reads) data. In this analysis, a new method was proposed for sensitive detection of genomic rearrangements by using the BWT of reads with three processes. Initially, breakpoint regions which are combined together by rearrangement were predicted from the discordant pairs by using a type of the conjugate gradient method. Then, reads partially matching the breakpoint regions were collected from the BWT of reads. After that, breakpoints were detected as branching points among the collected reads and their precise positions were determined.

Arram et al. [10] proposed a new mapping algorithm known as FPGA acceleration of reference-based compression for genomic data based on the FM-index search operation with algorithmic optimizations targeting compression ratio and speed. Also, hardware was designed based on a highly optimized version of the FM-index search optimization which is performed based on the BWT algorithm. Liu et al. [11] proposed a novel parallel BWT construction approach named de Bruijn branch-based BWT (deBWT) constructor for a large collection of genomes. This approach was used for representing and organizing the suffixes of input sequence with a novel data structure known as de Bruijn branch encoding. In this data structure, the advantage of the de Bruijn graph was taken to facilitate the comparison between the suffixes with a long common prefix that breaks the bottleneck of the BWT construction of repetitive genomic sequences. Also, the redundant comparisons between suffixes were reduced by this de Bruijn graph structure.

Rexline et al. [12] proposed the BWT-based approaches for compressing the biological sequences. In this study, different approaches such as BWT, Move-To-Front encoding (MTF), Run Length Encoding (RLE) and Arithmetic coding (ARI) were explored to compress the DNA sequences. Also, the comparison analysis was performed to identify the best compression method. Based on this analysis, it was concluded that the BWT achieves the best compression ratio to compress the DNA and protein sequences.

Fan et al. [13] proposed an efficient reference-based compression method that incorporates FM-index into complementary contextual models for improving compression performance. Initially, this method can locate the longest match by using the inverse of the reference index in an efficient manner. Then, a self-index was adapted to compress and store the reverse index of reference as a reference sequence for compression and decompression. For the unmatched symbol, the complementary contextual models integrate variable length pattern search for further improving the compression performance in the second stage.

III. RESULTS AND DISCUSSIONS

In this section, a comparative analysis of different researches on compression of genomic sequences using BWT studied in the above section is presented in terms of their merits and demerits. The following Table.1 shows the merits and demerits of the above mentioned BWT-based compression of genomic or a DNA sequence with the dataset used and also highlights the performance evaluation for each paper.

Table.1 Comparison of Different Researches on Compression of Genomic Sequences using BWT

Ref. No.		Merits	Demerits	Dataset	Performance Metrics
[3]		A similarity between two	Random sequences	mtDNA sequences	-Nil-

		sequences was easily estimated using the distance measures.	have little influence on the constructing of the phylogenetic trees.		
[4]		Particular reordering of the sequences may improve the compression.	Processing time was very high.	SRX001540 dataset	CPU time: (Stage 1) BWT-SAP=3520sec; (Stage 2) Bzip2=601sec; PPMd (default)=347sec; PPMd (large)=3116sec; -mx9=11204sec Compression ratio (bits/base) Bzip2=1.40; PPMd (default)=1.21; PPMd (large)=1.28; -mx9=1.34
[5]		Minimized run time and error rate.	Limited by the capacity of Block RAM. Also, resource utilization was high.	Simul-36-002, Simul-36-001, Simul-72-001, Simul-72-0005 Real-36, Real-72	Run time: 32-base=1.9ms; 36-base=2.4ms; 40-base=3.9ms Error rate: Simul-36-002=2%; Simul-36-001=1%; Simul-72-001=1%; Simul-72-0005=0.5%
[6]		High	Space	43 and 85 million read instances i.e.,	0043M: Efficiency=0.98; Memory=0.68GB; Wall clock time=0.75µsec/input

		efficiency.	complexity was high.	0043M and 0085M	base 0085M: Efficiency=0.98; Memory=1.30GB; Wall clock time=0.77µsec/input base
[7]		It able to construct the self-index in an extremely short duration.	Still, it requires further improvement on compression ratio.	Files s001 and s005 from Repetitive corpus	Patterns of length=5: Length of contexts=5: File s001: Compression ratio=2.26%; Localization time per occurrence=2.56µsec File s005: Compression ratio=6.01%; Localization time per occurrence=2.37µsec
[8]		The simplest method to compare two DNA sequences.	The performance efficiency was not analyzed.	Complete β-globin genes of 15 Species	-Nil-
[9]		Efficient detection of genomic rearrangements.	It requires an efficient compression method to compress the large-scale datasets.	Whole-Genome Sequencing (WGS) datasets	Sensitivity=90%; False detection rate=<1%
[10]		Reduced compression time.	Compression ratio was less.	FASTA and FASTQ format data sets	FASTA file: Compression time=0.08sec; Compression ratio=10.937% FASTQ file:

					Compression time=3.1sec; Compression ratio=14.1%
[11]		Efficient and scalable to construct BWT for large dataset.	High time complexity since it requires multiple executions in second-stage.	Three datasets: i. Dataset consists of 10 in silico human genomes; ii. Dataset consists of a set of simulated contigs; iii. Dataset consists of eight primate genomes	Running time with 32 CPU cores: deBWT: Human genomes=134min; Human contigs=129min; Primate genomes=330min deBWT (no conversion): Human genomes=48min; Human contigs=56min; Primate genomes=100min
[12]		Simple and better flexibility.	Less compression ratio.	Four datasets of protein sequences: Haemophilus Influenzae (HI), Human (HS), Methanococcus Janaschii (MJ) and Saccharomyces Cerevisiae (SC)	Average Compression ratio: BWT,MTF,RLE and ARI=4.439bits/sec; BWT,MTF and ARI=4.276bits/sec; BWT and ARI=4.157bits/sec
[13]		Better performance in the compression ratio of DNA sequences.	Similar symbols with different cases were not grouped together.	KOREF_20090131 and YH datasets	Compression size=3300bytes; Compression time=520sec

IV. CONCLUSION

In this paper, a detailed comparative study on different researches on BWT compression algorithm to compress the genomic or DNA sequences is presented. Through this comparative analysis, it is obviously noticed that many researchers have practiced on compression of genomic or DNA sequences with satisfied performance. Even though, the compression ratio is still not increased significantly using those frameworks. Therefore, the future extension of this study would be focused on the improvements on BWT-based compression of genomic sequences with a higher compression ratio than the state-of-the-art compression algorithm.

REFERENCES

[1] Adjeroh, Zhang, Y., Mukherjee, A., Powell, M., & Tim Bell, "DNA sequence compression using the burrows-wheeler transform," In Proceedings. IEEE Computer Society Bioinformatics Conference, pp. 303-313. IEEE,Aug 2002.

[2] Manzini, G. "An analysis of the burrows—wheeler transform", Journal of the ACM (JACM), 48(3), pp. 407-430, May 2001.

[3] Yang, L., Zhang, X., & Wang, T, " The burrows–wheeler similarity distribution between biological sequences based on burrows–wheeler transform", Journal of theoretical biology, 262(4),pp. 742-749, 2010.

[4] Cox, A. J., Bauer, M. J., Jakobi, T., & Rosone, G," Large-scale compression of genomic sequence databases with the burrows wheeler transform.", Bioinformatics, 28(11),pp. 1415-1419, May 2012.

[5] Xin, Y., Liu, B., Min, B., Li, W. X., Cheung, R. C., Fong, A. S., & Chan, T. F." Parallel architecture for DNA sequence inexact matching with burrows-wheeler transform", Microelectronics Journal, 44(8), pp. 670-682, May 2013.

[6] Bauer, M. J., Cox, A. J., & Rosone G, " Lightweight algorithms for constructing and inverting the WT of string collections". Theoretical Computer Science, 483, pp.134-148, 2013.

[7] Prochazka, P., & Holub, J," Compressing similar biological sequences using fm-index. In Data Compression Conference (DCC) ,pp. 312-321, IEEE 2014.

[8] Chun Li, Huan Liu, Junhong Liu, Yuping Qin, & Zhifu Wang," A burrows-wheeler transform based method

for DNA sequence comparison", Computational Biology and Bioinformatics, 2(3), pp. 33-37, May 2014.

[9] Kimura, K., & Koike, A, "Analysis of genomic rearrangements by using the burrows-wheeler transform of short-read data", BMC bioinformatics, 16(18), S5, Sept 2015.

[10] Arram, J., Pflanzer, M., Kaplan, T., & Luk, W." FPGA acceleration of reference-based compression for genomic data", In Field Programmable Technology (FPT), 2015 International Conference on (pp. 9-16). IEEE, 2015.

[11] Bo Liu, Dixian Zhu, & Yadong Wang, "deBWT: parallel construction of burrows–wheeler transform for large collection of genomes with de Bruijn-branch encoding", Bioinformatics, 32(12), i174-i182, 2016.

[12] Rexline, S. J., Gerard, A. R., Lobo, F. T. (2017), "Higher compression from burrows-wheeler transform for DNA sequence", International Journal of Computer Applications, 173(3), 11-15.

[13] Fan, W., Dai, W., Li, Y., & Xiong, H, "Complementary contextual models with FM-index for DNA compression", In Data Compression Conference (DCC), 2017 (pp. 82-91), IEEE April 2017.