A Novel DNA Sequence Compression Scheme Using Both Intra and Inter Sequences Correlation

K.O. Cheng*, N.F. Law† and W.C. Siu#
*†#Centre for Signal Processing, Department of Electronic and Information Engineering, the Hong Kong Polytechnic University, Hong Kong
*E-mail: encko@polyu.edu.hk Tel: +852-2766 6201
†E-mail: ennflaw@polyu.edu.hk Tel: +852-2766 4746
#E-mail: enwcsiu@polyu.edu.hk Tel: +852-2766 6229

Abstract— **Classical DNA sequence compression algorithms consider only intra-sequence similarity, i.e., similar subsequences within the DNA sequence are found and encoded together. In this work, in addition to the intra-sequence similarity, we exploit the inter-sequence similarities in that similar subsequences are found within the DNA sequence as well as from other reference sequences. Hence, highly similar sequences from the same population or partially similar chromosome sequences of the same species can be compressed together to reduce the storage space. Experimental results show that the proposed scheme achieves good compressibility for both partially similar chromosome sequences and highly similar population sequences.**

I. Introduction

The growth of public databases storing DNA sequences has been stimulated by advancement of technologies and the usefulness of DNA in areas such as forensics applications in recently years [1]. In order to reduce the storage space and lessen the transmission load, effective compression algorithms are needed to compress these sequences. Traditional DNA compression algorithms found similar subsequences within the DNA sequence which are then encoded together. Examples include BioCompress [2], GenCompress [3], DNACompress [4] and DNAPack [5]. While these algorithms employ different strategies to exploit the intra-sequence similarity, the average bit per base (bpb) can only be reduced from 2 to 1.73 for benchmark DNA sequences [4].

Studying the DNA sequences stored in public databases revealed that similarities can be found among a number of DNA sequences. For example, the chromosome sequences of one species are partially similar to each other; similar subsequences can be found between two different chromosome sequences [6, 7]. Besides, DNA sequences from different individuals of the same species/population are highly similar to each other [8]. Hence, great saving in bpb can be achieved if a sequence is encoded with respect to another sequence. The work in [8] encoded the base-to-base differences between two DNA sequences while that in [9] used a suffix array to store similar subsequences which was followed by differential coding of the suffix array data. Classical Lempel-Ziv compression has also been modified to improve the compression performance in RLZ-opt [10], GDC [11] and FRESCO [12]. These algorithms [8–12] are very effective in compressing the highly similar population sequences. Despite that, their strategies would not be effective in characterizing the partially similar chromosome sequences. The objective of this study is to develop a compression scheme that exploits intra-sequence and inter-sequence similarities and provides effective solution to both highly similar and partially similar DNA sequences.

II. Background

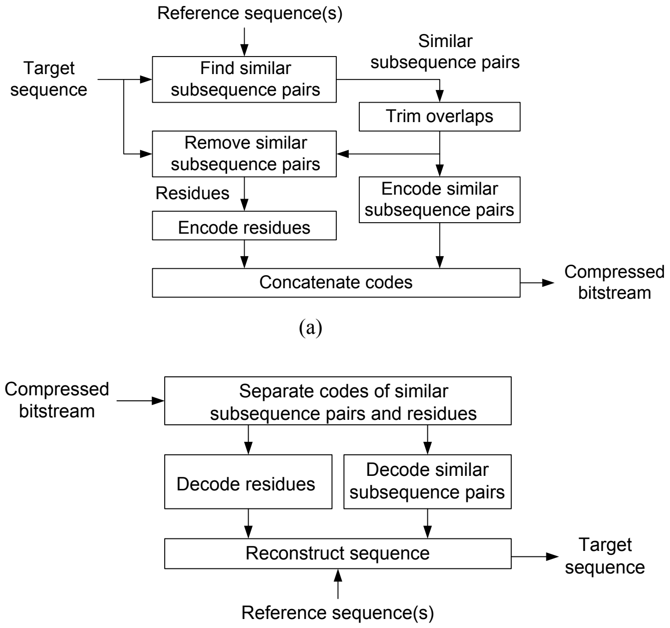
Classical DNA sequence compression algorithms [2-5] find similar subsequences within the DNA sequence to be compressed and then encoded them together. The similar subsequences can be in the form of exact or approximate repeats and reverse complementary repeats [2-4, 13]. The two subsequences in the approximate repeats have similar composition of bases but with some mismatches in certain positions. The mistmatches are characterized through base substitution, deletion or insertion. The reverse complementary repeats refer to the matching of the two subsequences by first replacing bases with their complementary bases and then matching in the reverse order [6].

In addition to intra-sequence similarity, inter-sequence similarity can always be found between two DNA sequences. Consider the population sequences of human mitochondria. Out of the 3615 sequences, the average number of bases that deviate from the revised Cambridge reference sequence (GenBank accession number: AC_000021) is only 33.8 base pairs [8]. As compared to the sequence length of 16K, the base composition has an average similarity over 99.7%. Therefore, algorithms such as [8-12] that consider the inter-sequence similarity are very effective and have very high compression ratio.

In addition to the highly similar population sequences, studies have found that similarities also exist among different chromosome sequences of the same species [7]. For example, the 16 chromosome sequences of the yeast *S. cerevisiae* contain both intra-sequence and inter-sequence similarities. In chromosome sequence I, the ratio of intra-sequence similarity and inter-sequence similarity is about 1:10. As inter-sequence similarities are significant, incorporating these similarities in DNA sequence compression can certainly improve the compression ratio. However, it should be noted that only certain parts of the two chromosome sequences are similar to each other. Strategies such as base-to-base difference or differential coding would not be effective to handle this type of similarity. A more general scheme to exploit the partial similarity is needed.

III. PROPOSED COMPRESSION SCHEME

A compression scheme that considers both intra-sequence and inter-sequence similarities is proposed. In the proposed scheme, instead of characterizing the base-to-base difference, similar subsequences within two DNA sequences are found and encoded together. Hence, both highly similar and partially similar sequences can be encoded effectively. Fig. 1 shows the encoding and the decoding processes of the proposed compression scheme.


Reference sequence(s)
Target sequence
Find similar subsequence pairs
Similar subsequence pairs
Trim overlaps
Remove similar subsequence pairs
Residues
Encode similar subsequence pairs
Encode residues
Concatenate codes
Compressed bitstream
Compressed bitstream
Separate codes of similar subsequence pairs and residues
Decode residues
Decode similar subsequence pairs
Reconstruct sequence
Target sequence
Reference sequence(s)

(a)

(b)

Fig. 1 An overview of the proposed compression scheme in the (a) encoding and (b) decoding processes.

In the encoding phase, similar subsequences are found between the sequence to be compressed (called the target sequence) and the reference sequence. The reference sequence can be the target sequence or other similar sequence to be compressed together. Sequence alignment algorithms [14] are used to find the similar subsequences pairs. In this paper, PatternHunter [15] is used for searching similar subsequences in two sequences because of its efficiency. Fig. 2 shows an example of the similar subsequence pair in which an approximate repeat is identified. The two subsequences have similar base composition but with some mismatches. Three lists are generated to represent the required operation for matching the two subsequences. They are:

- the operation list: it contains the type of operation in matching the corresponding bases in the two subsequences. The type of operation can be base substitution (S), insertion (I) or deletion (D).
- the offset list: it marks the relative positions at where the bases are different in the two subsequences.
- the base list: it contains the replaced base for the substitution operation and the inserted base for the insertion operation.

For the approximate subsequence in Fig. 2, the first unmatched base is at the 3rd position relative to the beginning of the subsequence in the target sequence. It can be described through insertion. Hence the operation list is I, the offset list is 3 and the base list is A. The second unmatched base can be described by substitution at the 5th position relative to the previous operation. Thus the operation list, the offset list and the base list become IS, 35 and AA respectively.

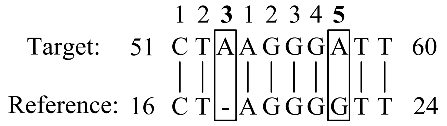

1 2 3 1 2 3 4 5
Target: 51 C T A A G G G A T T 60
Reference: 16 C T - A G G G G T T 24

Fig. 2 An example of the similar subsequence pair. The matched bases are connected by vertical lines while the unmatched bases are marked using rectangular boxes. The number above individual bases shows the offset position.

After identifying all the similar subsequences in the sequences, it is possible that subsequences of the target sequence in two different pairs overlap partially with each other. To solve this problem, our idea is to keep a long subsequence with fewer modifications rather a short subsequence with many modifications. Hence the overlapping parts are kept in the pair which incurs fewer modifications while the corresponding parts in another pair are removed. After solving the overlapping problem, the similar subsequences are removed from the target sequence. This subsequence will be combined with the three lists to form a single compressed bitstream.

The first part of the compressed bitstream is the concatenated operation lists. An additional symbol is added as a delimiter to separate the lists from different subsequence pairs. Arithmetic coding [16] is then used to compress the concatenated operation lists. A first-order character-based model is adopted in probability calculation for the arithmetic coding. The second part of the compressed bitstream contains the concatenated offset lists. There is no delimiter introduced in the offset list since the subsequence boundaries can be deduced from the operation lists. The offset values are encoded primarily using Elias gamma code [8, 17]. However, for long consecutive offset values, run-length coding is used. The third part of the compressed bitstream is the base lists of all repeats. Arithmetic coding is used for its compression. The final part of the compressed bitstream contains the encoded similar subsequence pairs. The decoding process is the reverse of the encoding process. However, there is no need to find the similar subsequence pairs. The target subsequence can simply be reconstructed by decoding the subsequence pairs and the three lists for mismatches in the pairs. The decoding time is thus much less than the encoding time.

IV. EXPERIMENTAL RESULTS

Experiments have been conducted to evaluate the performance of the proposed compression scheme. The first dataset is composed of 3615 *Homo sapiens* mitochondrial sequences. These sequences have small variations across human population. The second dataset consists of the sixteen chromosome sequences of *S. cerevisiae*. All the sequences are available in GenBank (http://www.ncbi.nlm.nih.gov/genbank/). These chromosome sequences are found to be partially similar to each other.

A. *Compression of a Homo Sapiens Sequence with Reference to the Revised Cambridge Reference Sequence*

Our proposed scheme is applied to compress the 3615 *Home sapiens* mitochondrial sequences with an average length of 16k. The revised Cambridge reference sequence with GenBank accession number of AC_000021 is used as the reference sequence in the compression as it is a good representative of all the sequences. Table I summarizes the similarity information of these 3615 sequences with the revised Cambridge reference sequence. The percentage of the similar subsequences has an average of 99.996%. The average number of modifications per base is only 0.0021 bpb. Hence, it is highly effective for using the base-to-base difference coding for this set of sequences. Table II compares the bpb of our proposed algorithms with several existing compression approaches. Our proposed algorithm achieves a bpb of 0.0389 which is lower than those of algorithms RLCSA [9], RLZ-opt [10], GDC [11] and FRESCO [12] which ranges from 0.0395 to 0.1873. The bpb of our proposed algorithm is still comparable to the best bpb attained by the algorithm from Brandon et al. [8], 0.0314. If one considers only intra-sequence similarity, the bpb from GenCompress [3] is 1.9436 which is much higher than the other algorithms that consider inter-sequence similarity. Hence inter-sequence similarity should be considered in compressing population sequences.

TABLE I
SIMILARITY INFORMATION BETWEEN A HOMO SAPIENS MITOCHONDRIAL SEQUENCE AND THE REVISED CAMBRIDGE REFERENCE SEQUENCE

	Average	Minimum	Maximum
Sequence length	16287.4	15436	16584
Repeat length	16286.7	15436	16583
Percentage of repeat	99.996%	99.946%	100%
Number of modifications per base	0.0021	0	0.0068

TABLE II
THE BPBS FOR COMPRESSING THE HOMO SAPIENS MITOCHONDRIAL DATASET USING DIFFERENT ALGORITHMS

Algorithms	bpb
GenCompress	1.9436
Brandon et al.	0.0314
RLCSA	0.0395
RLZ-opt	0.0615
GDC	0.1873
FRESCO	0.0807
Proposed Scheme	0.0389

B. *Compression of a Chromosome Sequence with Reference to Another Sequence*

A chromosome sequence of *S. cerevisiae* is compressed by considering its similarity with another reference chromosome sequence. As evidence in Table III, this set of 16 chromosome sequences is very difficult to compress as the average bpb for considering only intra-sequence similarity is always more than 1.92 for intra-similarity based algorithms such as arithmetic coding [16] and context tree weighting [18] and GenCompress [3]. In Table IV, results from our proposed scheme are shown. Considering only intra-sequence similarity, our proposed scheme achieves an average bpb of 1.9226 which is similar to other existing algorithms based on only intra-similarity. However, if we consider inter-sequence similarity by compressing the sequence with reference to another chromosome sequence, we can see that the average bpb drops to about 1.8434. In fact, by using a reference sequence, the bpb always drops as compared to the case without a reference sequence. The additional savings in bpb depend on the similarity between the target and the reference sequences. As chromosome sequences I and VIII have high similarity, the bpb is able to drop from 1.8409 to 1.6131. For less similar sequences such as chromosome sequences XI and III, the bpb drops by 0.0269 only.

TABLE III
THE BPBS FOR COMPRESSING THE 16 CHROMOSOME SEQUENCES IN S. CEREVISIAE USING DIFFERENT ALGORITHMS

Algorithms	bpb
Arithmetic coding	1.9513
Context Tree Weighting	1.9454
GenCompress	1.9208
Brandon et al.	NA
RLCSA	5.8430
RLZ-opt	2.2776
GDC	2.0152
FRESCO	1.9959

TABLE IV
THE EXPERIMENTAL RESULTS OF COMPRESSING A SEQUENCE (TARGET SEQUENCE) IN S. CEREVISIAE BY REFERENCING TO ANOTHER CHROMOSOME SEQUENCE (REFERENCE SEQUENCE)

Target sequence	Reference sequence[a]	Without reference (bpb)	With reference(bpb)
I	VIII	1.8409	1.6131
II	VII	1.9501	1.8750
III	XII	1.9534	1.8782
IV	X	1.8855	1.8478
V	XIV	1.9052	1.8510
VI	XIV	1.9538	1.8182
VII	II	1.9069	1.8660
VIII	I	1.9519	1.8518
IX	X	1.9546	1.8470
X	IV	1.9511	1.8535
XI	III	1.9484	1.9215
XII	VII	1.8357	1.7978
XIII	XVI	1.9496	1.8742
XIV	IV	1.9529	1.8756
XV	XVI	1.9218	1.8762
XVI	XV	1.8997	1.8481
Average		**1.9226**	**1.8434**

[a] The best reference sequence found in the experiment is chosen.

Noted that algorithm from Brandon et al. [8], RLCSA [9], RLZ-opt [10], GDC [11] and FRESCO [12] are specifically designed for highly similar sequences. If the variations across the sequences are substantial, they may not be able to compress them effectively. As shown in Table III, we can see that in compressing the 16 chromosome sequences of *S. cerevisiae*, RLCSA, RLZ-opt, GDC and FRESCO has a bpb worse than the intra-similarity based algorithms. The first three algorithms even could not compress the sequences as they have bpb over 2. For Brandon et al. algorithm, differential variants information is not available so the program cannot be applied to compress the 16 chromosome sequences. On the other hand, our proposed scheme can consistently reduce the bpb and has a lower bpb than all the existing algorithms. This shows that our proposed scheme is able to compress sequences with different kinds of similarity structure.

C. Compression of Multiple Chromosome Sequences

As discussed in Section IV.B, the bpb drops by considering similarity between two chromosome sequences. In this part, multiple chromosome sequences of *S. cerevisiae* are compressed together. Table V shows the experimental results. The second column shows the average bpb from single sequence compression while the third column shows the bpb from multiple sequences compression. We can see that the bpb for compressing a number of chromosome sequences together is always smaller than that of compressing them separately. For example, the group of compressing chromosome sequences IV, V, VI, IX, X and XIV together reduces the bpb from 1.9227 to 1.8611. If the group of sequences have high similarity to each other, it is always advantageous to compress together.

TABLE V
THE BPBS OF COMPRESSING MULTIPLE CHROMOSOME SEQUENCES IN S. CEREVISIAE

Chromosome sequences	Single sequence compression	Multiple sequences compression
IV, IX, X	1.9147	1.8702
IV, X, XIV	1.9187	1.8728
IV, IX, X, XIV	1.9232	1.8692
IV, V, IX, X, XIV	1.9207	1.8647
IV, VI, IX, X, XIV	1.9254	1.8652
IV, V, VI, IX, X, XIV	1.9227	1.8611

V. CONCLUSIONS

In this paper, we extend the idea of intra-sequence similarity to inter-sequence similarity in DNA sequences compression. Instead of finding base-to-base differences between the target and the reference sequences, similar subsequences in the target and the reference sequences are searched and encoded together to achieve compression. In this way, our proposed scheme is applicable to compress a single sequence using only intra-sequence similarity as well as multiple sequences with different kinds of similarity structure. Our proposed compression scheme has been applied to compress the 16 chromosome sequences of *S. cerevisiae* and a set of individual sequences from human mitochondrial data. There is an average improvement of 4.13% for compression of *S. cerevisiae* when compared with single sequence compression. For the human mitochondrial sequences, the bpb of the proposed algorithm is 0.0389 which is comparable to existing algorithms that are specifically designed for compressing highly similar sequences. However, these existing algorithms are not suitable for sequences with partial similarity so that the proposed scheme is general in compressing DNA sequences with different degree of similarity.

ACKNOWLEDGMENT

This work was supported by the GRF Grant, 519012 (project code: BQ32Z), of the Hong Kong SAR Government and the Centre for Signal Processing, The Hong Kong Polytechnic University.

REFERENCES

[1] 1000 Genomes, http://www.1000genomes.org/.

[2] S. Grumbach and F. Tahi, “A new challenge for compression algorithms: genetic sequences,” *Journal of Information Processing and Management*, vol. 30, no. 6, pp. 875-886, Nov. – Dec. 1994.

[3] X. Chen, S. Kwong, and M. Li, "A compression algorithm for DNA sequences," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 4, pp. 61-66, July – Aug. 2001.

[4] X. Chen, M. Li, B. Ma, and J. Tromp, "DNACompress: fast and effective DNA sequence compression," *Bioinformatics*, vol. 18, no. 12, pp. 1696-1698, Dec. 2002.

[5] B. Behzadi and F.L. Fessant, "DNA compression challenge revisited: a dynamic programming approach," *Combinatorial Pattern Matching, Lecture Notes in Computer Science*, vol. 3537, pp. 190-200, 2005.

[6] Paula Wu, N.F. Law, and W.C. Siu, "Cross chromosomal similarity for DNA sequence compression," *Bioinformation*, vol. 2, no. 9, pp. 412-416, July 2008.

[7] W. Li, G. Stolovitzky, P. Bernaola-Galvan, and J.L. Oliver, "Compositional heterogeneity within, and uniformity between, DNA sequences of yeast chromosomes," *Genome Research*, vol. 8, pp. 916-928, Sept. 1998.

[8] M.C. Brandon, D.C. Wallace, and P. Baldi, "Data structures and compression algorithms for genomic sequence data," *Bioinformatics*, vol. 25, no. 14, pp. 1731-1738, July 2009.

[9] V. Mäkinen, G. Navarro, J. Sirén, and N. Välimäki, "Storage and retrieval of highly repetitive sequence collections," *Journal of Computational Biology*, vol. 17, no. 3, pp. 281-308, March 2010.

[10] S. Kuruppu, S.J. Puglisi, and J. Zobel, "Optimized relative Lempel-Ziv compression of genomes," *Proceedings of the Thirty-Fourth Australasian Computer Science Conference*, pp. 91-98, 2011.

[11] S. Deorowicz and S. Grabowski, "Robust relative compression of genomes with random access," *Bioinformatics*, vol. 27, no. 21, pp. 2979-2986, Nov. 2011.

[12] S. Wandelt and U. Leser, "FRESCO: referential compression of highly similar sequences," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 5, pp. 1275-1288, Sept./Oct. 2013.

[13] P. Wu, N.-F. Law, and W.-C. Siu, "Analysis of cross sequence similarities for multiple DNA sequences compression," *International Journal of Computer Aided Engineering and Technology*, vol. 1, no. 4, pp. 437-454, Sept. 2009.

[14] A.W.-C. Liew, H. Yan, and M. Yang, "Pattern recognition techniques for the emerging field of bioinformatics: a review," *Pattern Recognition*, vol. 38, no. 11, pp. 2055-2073, Nov. 2005.

[15] B. Ma, J. Tromp, and M. Li, "PatternHunter: faster and more sensitive homology search," *Bioinformatics*, vol. 18, no. 3, pp. 440-445, March 2002.

[16] A. Said, "Introduction to arithmetic coding - theory and practice," *Hewlett-Packard Laboratories Report*, HPL-2004-76, Palo Alto, CA, April 2004.

[17] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 194-203, March 1975.

[18] CTW (Context Tree Weighting) website: http://www.ele.tue.nl/ctw/