# DNA Data Compression in the Post Genome Era

**Hisahiko Sato**[1] hiko@biochem.s.u-tokyo.ac.jp
**Takashi Yoshioka**[2] yossie@rd.nttdata.co.jp
**Akihiko Konagaya**[3] kona@jaist.ac.jp
**Tetsuro Toyoda**[4] toyop@gsc.riken.go.jp

[1] Dept. of Biophysics and Biochemistry, The University of Tokyo, 2-11-16, Yayoi Bunkyo-ku, Tokyo 108-8639, Japan
[2] Bioinformatics group, NTT DATA, inc, Kayabacho Tower, 1-21-2, Shinkawa, Chuo-ku, Tokyo 104-0033, Japan
[3] Japan Advanced Institute of Science and Technology (JAIST) HokurikuAsahidai 1-1, Tatsunokuchi, Ishikawa 923-1211, Japan
[4] Genomic Sciences Center, RIKEN, 1-7-22 Suehiro-cho, Tsurumi-ku, Yokohama, Kanagawa 230-0045, Japan

**Keywords:** DNA, compression, BLAST, database, LZ77

## 1 Introduction

Today, increasing genome sequence data of organisms lead DNA database size two or three times bigger annually. Thus, it becomes very hard to download and maintain such data in a personal local system. Algorithms for compressing DNA sequences, such as GenCompress [1], Biocompress [2] and fact [5], are available as tools to manage such works. Although these algorisms use characteristics of DNA like reverse complement or point mutation, their compression rate is about 1.74 bits per base (78% in compression ratio) [3]. Therefore, compression of DNA sequences is recognized as a tough task and needs much improvement.

Here we present a new compression method named “GSCompress” (Genome Sequence Compress) based on a fact that the variation of sequences in the same organism is finite, that is, the net size of DNA information of an organism must be equal to or less than that of whole genome. Since the gross size of sequence database of an organism, such as *E. coli* or *S. cerevisiae,* is three or more times bigger than that of its genome, it is expected that the compression ratio must be greatly improved by enlarging LZ77-scheme dictionary size to the genome size. We examined how much the compression ratio would be improved by enlarging the dictionary size. As a result, it was found that the ratio was greatly improved to about 0.30 bits per base (96% in compression ratio).

## 2 Method and Results

### 2.1 Compression Algorithms

Most of the compression methods used today including DNA compression fall into two categories. First is statistical method, which compresses data by replacing a more popular symbol to a shorter code. Second is dictionary-based scheme, which compresses data by replacing long sequences by short pointer information to the same sequences in a dictionary.

In statistical methods, arithmetic coding and CTW are known to compress the DNA data well [3] and Huffman coding is known to compress not very efficiently [4]. In spite of the good compression ratio, arithmetic coding and CTW have disadvantages such as low decompression speed. Thus, in GSCompress, we employ “formatdb” which is used to convert FASTA format to binary codes for

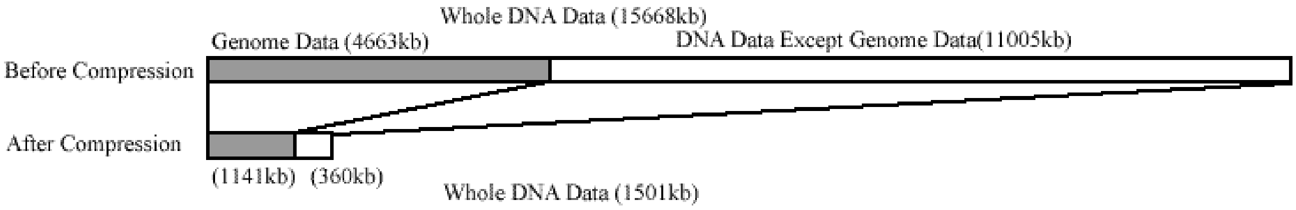

Figure 1: Compression result of *E. coli* DNA data.

BLAST search. "Formatdb" compresses the sequence with a Huffman-like coding method but deals minor symbols very efficiently and can uncompress fast (http://sapiens.wustl.edu/blast/blast/ncbi20ntfmt.html).

For dictionary-based methods, LZ77 scheme is known to be the best method for compressing DNA data so far. Several DNA-oriented algorisms have been tried to make the best of the characteristics of DNA such as reverse complement and point mutation in order to apply LZ77 scheme more efficiently [1].

In GSCompress, we employed LZ77 scheme with reverse complement as a dictionary-based scheme. We assumed that the total size of DNA information of an organism must be limited to that of genome and expected that DNA data would become very compressible by enlarging the dictionary size to genome. Thus, we removed a limitation on dictionary window size of LZ77 scheme and measured the dependency of compression ratio on the dictionary size. Also, we employed arithmetic coding schemes to compress pointer information efficiently to deal characteristics of point mutation. (in detail http://www.gsc.riken.go.jp/GSCompress/)

### 2.2 Result

We tried our compression scheme on the whole DNA data of *E. coli* (*Escherichia coli*) and *S. cerevisiae* (*Saccharomyces cerevisiae*) because of their small genome sizes. We extracted all *E. coli* origin sequence data from DDBJ release 39 for DNA data of *E. coli* and used a sequence data named "yeastGenBank" in the SGD database for *S. cerevisiae.* (http://genome-www.stanford.edu/Saccharomyces/) The results of the compression ratio of DNA data of each organism are shown in Figure 1 and Table 1. The results show that the compression ratios of organisms are about 87-91%. The data except the genome data are efficiently replaced by pointer information to the genome data and compressed at the ratio of more than 96%.

Table 1: Compression ratio of DNA data of *E. coli* and *S. cerevisiae.*

| species | whole DNA data | DNA data except genome |
|---|---|---|
| *E. coli* | 90.4% (0.77 bits/base) | 96.6% (0.27 bits/base) |
| *S. cerevisiae* | 87.2% (1.02 bits/base) | 96.2% (0.30 bits/base) |

## 3 Discussions

Most of DNA compression researches [1, 2, 3, 5] are focused on compressing single sequences. We tackled with the compression problem from a genome point of view. The result shows that DNA data becomes very compressible by enlarging the dictionary size. This method has an advantage in incremental update of a sequence database since additional sequences can be compressed well by using existing sequences in a local database. Also, we expect that DNA data of some organism would be very compressible with the dictionary of evolutionary related organism. Although this method requires time and memory in compression process, we can successfully accelerate the speed by paralleling the process by clustered computers. Decompression speed is much faster than compression.

## References

[1] Chen, X., Kwong, S., and Li, M., A compression algorithm for DNA sequences and its applications in genome comparison, *Genome Informatics* , 10:52–61, 1999.

[2] Grumbach, S. and Tahi, F., A new challenge for compression algorithms: genetic sequences, *Information Processing & Management*, 30:875–886, 1994.

[3] Matsumoto, T., Sadakane, K., and Imai, H., Biological sequence compression algorithms, *Genome Informatics*, 11:43–52, 2000.

[4] Matsumoto, T., Sadakane, K., Imai, H., and Okazaki, T., Can general-purpose compression schemes really compress DNA sequences?, *Currents in Computational Molecular Biology*, Universal Academy Press, 76–77, 2000.

[5] Rivals, E., Delahaye, J.P., Dauchet, M., and Delgrange, O., A guaranteed compression scheme for repetitive DNA sequences, *LIFL Lille I University, technical report*, IT-285, 1995.