FICS challenge 2010: Compression of DNA sequencing data

Veli Mäkinen

Department of Computer Science, University of Helsinki, Finland.
{vmakinen}@cs.helsinki.fi

Abstract. Next generation sequencing (NGS) platforms are generating massive amounts of short DNA reads in hundreds of laboratories around the world. The data contains lots of redundancy since the study targets are typically model organisms, whose whole genome sequences are almost completely known. Generic compression methods do not work well enough for short DNA reads, and hence tailored mechanisms are being developed. The proposed challenge is to develop such a compression mechanism for DNA reads. The results are evaluated on compression efficiency, compression time, decompression time, and on time to decompress i-th read.

1 Introduction

High-throughput next generation (short read) sequencing is revolutionizing the way molecular biology is researched [4]. The advances in the technology allow cost-effective ways to read millions of short fragments of DNA or RNA from a given sample. Different enrichement techniques make it possible to prepare samples that contain DNA from targeted areas of genome, such as from the vicinity of transcription factor binding sites. Whole genome assembly is also feasible with the new technology [3].

Although short reads are just an intermediate product towards sophisticated analyses, the databases typically store these read files as the raw data so that further analyses can be repeated. Due to obvious redundancy in the read data, it is important to consider compression mechanism to avoid investing on needless storage space. For whole individual genomes such compression methods have been proposed (see e.g. [2, 1]), and similar techniques work for compressing reads.

2 Challenge

The proposed challenge is to develop a tailored compression scheme for short read data. The compression mechanism can exploit the reference genome for free (one can assume that both the compressor and decompressor have the same reference sequence available). The compression strategy can be arbitrary, but as a hint, it may be a good approach to align the reads to the reference and store somehow the occurrence positions and the list of edit differences.

3 Input and output formats

Reads are typically stored together with their encoded quality values: the sequencing machine associates to each position a quality value telling how likely the nucleotide at that position is correctly measured. For this challenge, we ignore the quality values. It is hence sufficient that the decompressor outputs the original read file without the quality values. Also each read has typically some (redundant) header information which can be ignored for this challenge.

An example of input file for the compressor in fastq-format (`http://www.ncbi.nlm.nih.gov/sra/ERX001170?report=full`):

```
@ERR006459.1 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:1915 length=51
NAGAGAAAGAAGGAACCCTCCCTAAATCATTCCATGAAGCCAGTATCACCC
+ERR006459.1 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:1915 length=51
!IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIF?5II8>IH9@I/:4-.
@ERR006459.2 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:141 length=51
NTGGCGCATTTAAAGTAAGTGTGTGCAGAGACCAAGCCAAATGAGGCCCCA
+ERR006459.2 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:141 length=51
!EII$II2IIIIIII*IIIII3IHI.II5@G6+//-#(?$,.(5'*%#*(+
@ERR006459.3 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:95 length=51
NATCTCCCTGTTAGTGTCTTAAAAAAATCACAATAAAATATGTTGAAATTA
+ERR006459.3 091002_HWUSI-EAS451_0001_42FK7AAXX_K:3:1:0:95 length=51
!IIIIDDIIIIIIIIIIIIIIIIIIIIDI<<I>I66/392<-495//.69
...
```

An example of what the decompressor should output for our challenge problem:

```
NAGAGAAAGAAGGAACCCTCCCTAAATCATTCCATGAAGCCAGTATCACCC
NTGGCGCATTTAAAGTAAGTGTGTGCAGAGACCAAGCCAAATGAGGCCCCA
NATCTCCCTGTTAGTGTCTTAAAAAAATCACAATAAAATATGTTGAAATTA
...
```

3.1 Variants

Notice that keeping the order for the reads is usually essential (if one stores headers and qualities compressed separately). Also for *paired-end* reads the order is essential: The reads are then stored in two files (usually named `*_1.fastq` and `*_2.fastq`) forming pairs of reads (from the same lines) such that one read represents the head and the other read represents the tail of a DNA fragment. The length of the DNA fragment is approximately known and hence the read pair should occur at specific distance from each others in the reference genome.

Therefore the compressor and decompressor should support the following modes for compression:

1. Compressor gets one file at the time. Decompressor outputs the reads in the original order.

2. Compressor gets one file at the time. Decompressor outputs the reads in arbitrary order.
3. Compressor gets two files at the time (paired-end files). Decompressor outputs the reads in the original order.
4. Compressor gets two files at the time (paired-end files). Decompressor outputs the reads in arbitrary order, yet synchronized between the two files.

4 Evaluation

Compression efficiency, compression time, and decompression time are obvious measures that sould be optimized in the challenge. A more advanced objective is to provide random access to compressed file, that is, to allow local decompression. In the case of short reads, the typical operation would be to extract the i-th read. Providing such functionality is considered voluntary in this challenge.

5 Useful tools

It is perfectly ok to use outside software to do parts of the compressor (but not the whole thing of course). For aligning the reads to the reference one can use e.g. `bwa`[1], `bowtie`[2], `SOAP2`[3], or `readaligner`[4]. For coding the occurrence positions, one can use integer codes such as implemented in `Integer Coding Library`[5].

6 Example data

Example read files and reference genome is provided in the summer school.

7 Credits

To obtain credits from the challenge, the student should return a working compressor / decompressor and a short report describing the methods used at most one month after the summer school. The grade depends on how well the method evaluates againts its competitors and on how novel are the methods used.

[1] `http://bio-bwa.sourceforge.net/`
[2] `http://bowtie-bio.sourceforge.net/index.shtml`
[3] `http://soap.genomics.org.cn/soapaligner.html`
[4] `http://www.cs.helsinki.fi/group/suds/readaligner/`
[5] `http://www.di.unipi.it/~ferragin/Libraries/IntegerCoding/index.html`

References

1. M. C. Brandon, D. C. Wallace, and P. Baldi. Data structures and compression algorithms for genomic sequence data. *Bioinformatics*, 25(14):1731–8, 2009.
2. S. Christley, Y. Lu, C. Li, and X. Xie. Human genomes as email attachments. *Bioinformatics*, 25:274–5, 2009.
3. R. Li et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.*, 20:265–271, 2010.
4. M. L. Metzker. Sequencing technologies — the next generation. *Nat. Rev. Genet.*, 11(1):31–46, Jan. 2010.