BIOINFORMATICS APPLICATIONS NOTE
Vol. 25 no. 2 2009, pages 274–275
doi:10.1093/bioinformatics/btn582

Sequence analysis

Human genomes as email attachments

Scott Christley[1,†], Yiming Lu[1,†], Chen Li[1,*] and Xiaohui Xie[1,2,*]

[1]Department of Computer Science and [2]Institute for Genomics and Bioinformatics, University of California Irvine, Irvine, CA 92697, USA

Received on October 07, 2008; revised on October 31, 2008; accepted on November 06, 2008

Advance Access publication November 7, 2008

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: The amount of genomic sequence data being generated and made available through public databases continues to increase at an ever-expanding rate. Downloading, copying, sharing and manipulating these large datasets are becoming difficult and time consuming for researchers. We need to consider using advanced compression techniques as part of a standard data format for genomic data. The inherent structure of genome data allows for more efficient lossless compression than can be obtained through the use of generic compression programs. We apply a series of techniques to James Watson's genome that in combination reduce it to a mere 4 MB, small enough to be sent as an email attachment.

Availability: Our algorithms are implemented in C++ and are freely available from http://www.ics.uci.edu/∼xhx/project/DNAzip.

Contact: chenli@ics.uci.edu; xhx@ics.uci.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

With the announcement of the 1000 genomes project (1000genomes.org), the day of individual genomes is almost upon us. The project is estimated to generate about 8.2 billion bases per day, the equivalent of more than two human genomes every 24 h, during its 2-year production phase. With the total sequence to exceed 6 trillion nucleotide bases, it represents 60 times more sequence than all the public DNA data that have been deposited over the past 25 years. Despite the availability of cheaper and higher capacity hard drives, management of such a large dataset is not a trivial task. Archival of these datasets is being handled by the National Center for Biotechnology Institute (NCBI) and other international centers, but this does not alleviate the difficulties for researchers to download, copy, share and manipulate the genomic data as part of their analysis. General-purpose compression programs like gzip offer a substantial decrease in the size of the data, but the data can be compressed even further through special purpose algorithms that take the properties of genomic sequence into account. In this article, we present a series of compression techniques when taken together reduces the size of a single genome by orders of magnitude. We apply our algorithm to James Watson's (JW) genome (Wheeler *et al.*, 2008), compressing it to a mere 4 million bytes (MB).

2 METHODS

The key insight is to consider that any two human genomes are more than 99% identical, so it is much more efficient to store genomes as variations from a common reference genome; therefore only that 1% of variation needs to be stored. This technique diverges from the research attempting the challenging problem to compress a single genome (Chen *et al.*, 2002) or recent research on compressing a large database of unrelated sequences (White and Hendy, 2008). We do not consider the process of generating the variations, which can be a challenging problem in itself, but assume that the variation data have been provided. Variation data are comprised of single nucleotide polymorphisms (SNPs) and insertions/deletions of multiple nucleotides (indels). A SNP is stored as a position value on a chromosome and a single nucleotide letter of the polymorphism, an insertion is stored as a position value and the sequence of nucleotide letters to be inserted, and a deletion is stored as a position value and the length of the deletion. We currently treat inversions as indels. Given the variation data for a genome relative to a reference genome, we consider the following compression techniques to reduce the variation data even smaller.

2.1 Variable integers for positions (VINT)

Each position value for a variation indicates the place on a chromosome where the variation occurs. As the chromosomes sizes can be as large as 247 MB, this requires 4 bytes (standard C integer data type) to store the full possible range of position values. For many small values, a full 4 bytes is not needed, so our first compression technique is to use integers with a variable number of bits thus allowing smaller positions to be stored more efficiently. A variable integer (VINT) uses a single bit within 1 byte as an end-of-integer flag; so 1 byte can store values from 0 to 127, 2 bytes can store values from 128 to 16 383, etc. Any number of bytes can be concatenated together in a VINT to optimally store a position value with the one bit flag indicating the end of the VINT.

For the variation data, the position value for SNPs and indels are stored with a VINT. Furthermore, the sequence data for a SNP and an insertion is stored using 2 bits for each letter, and the length of the deletion is stored as a VINT.

2.2 Delta positions (DELTA)

The positions are still stored as absolute values, which means that variations will require increasing storage as they occur further towards the end of the chromosome. The next compression technique we consider is to store the position value as a relative position (DELTA) from the position of the previous variation; that is, the difference between the current and previous position going from the beginning to the end of each chromosome. The relative position will always be less than or equal to the absolute position,

[*]To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

also many variations tend to occur in nearby groups, so the relative values can be quite small.

2.3 SNP mapping (DBSNP)

Besides most of the genome being common among humans, much of the variation, especially SNPs, is also shared. In addition, the large majority of the SNPs are biallele, existing in only one of the two possible forms. Therefore in addition to having a reference genome, we can also use a reference variation map allowing common variation to be stored more efficiently. We consider using the SNPs deposited in the NCBI dbSNP database as our reference variation map though any list of SNPs could be used. This compression technique (DBSNP) transforms the reference SNP map into an ordered list of bits or bitmap whereby each bit has a value of 1 if that common SNP is in our genome's variation data otherwise it has a value of 0. This technique has the potential of increasing the size if there are very few common SNPs as the bitmap storage would exceed the space saved by storing the individual SNPs, but in practical use we have not encountered this. Furthermore, the size difference can be calculated and the reference SNP map discarded if it would increase the size.

2.4 *K*-mer partitioning (KMER)

The last compression technique is inspired by the observation that the sequence in the human genome is highly non-random, littered with repeat sequences or sequences with low complexity. We partition the insertion sequence data into *k*-mers, then use Huffman coding (Huffman, 1952) to encode all of the *k*-mers. Huffman coding uses the shortest possible representation based upon the frequency of the *k*-mer strings, and the sequence data are stored as a list of Huffman codes. If the length of the sequence data is not an integral number of *k*-mers then the remaining letters are stored using their 2-bit representation.

3 EXAMPLE

We applied our compression algorithm to JW genome (Wheeler *et al*., 2008) using the hg18 release from the UCSC Genome Browser as our reference genome and the UCSC SNP track 129 as our reference SNP map. The results can be seen in Table 1.

Mapping the JW genome to the reference genome results in 3.5 M differences (variations) including 3.3 M SNPs and 220 K indels, which altogether can be packed into a 84 MB file if we keep only the positions of the variations and the corresponding nucleotide changes. Compressing using gzip reduces the file sizes further, but our compression is even better at 4.5 times smaller than the gzip files, resulting in a final data size of 4.1 MB for the JW genome.

There are three notable reductions in compression size in Table 1: (1) using DELTA to encode positions of variations, which leads to a 52% reduction when compared with VINT; (2) DBSNP mapping cuts in half the data size of the SNP variation; and (3) KMER results in 58% reduction in the insertion variations.

The NCBI dbSNP contains about 11 million (M) SNPs, 99% of which are bi-allele. Of these biallele SNPs, 9 M are consistent with the human reference genome in the sense that the alleles in the reference genome belong to one of the two forms in the dbSNPs, and are coded as a bit vector with 1.13 MB . The JW genome contains 3.32 M SNPs with 2.72 M in common with those in dbSNP, leaving 0.61 M as novel SNPs. The final compressed SNP data at 3.2 MB means that each of the 3.32 million SNPs require slightly less than 1 byte per SNP to store with full information of the position in the genome and the polymorphism.

Table 1. Data sizes for compression techniques

Compression	SNPs (KB)	Deletions (KB)	Insertions (KB)	Total (KB)
Entire genome				3 169 831
Map to ref genome	68 519	1741	14 274	84 534
Map to ref genome + gzip	14 803	588	2687	18 078
VINT	13 733	765	803	15 301
VINT/DELTA	6475	507	712	7694
VINT/DELTA/DBSNP	3292	507	712	4511
VINT/DELTA/DBSNP/KMER	3292	507	302	4101

For the insertion data, we experimented with different size *k*-mers to find the *k* giving the best compression (considering both the insertion size distribution and the *k*-mer frequency, see Supplementary Figures). This turned out to be $k = 4$ for the JW genome and corresponds to the results shown in Table 1.

4 CONCLUSION AND DISCUSSION

We have shown that by using a series of techniques we can significantly reduce the size needed to store a human genome. Given that the other human genomes have similar variation, the 1000 human genomes with its 6 trillion bases can be stored in 4–5 GB, easily handled by today's computers. Our technique does require a reference human genome (∼3 GB) and a reference SNP map (∼1.2 GB), but this cost is amortized over the total number of genomes. Our method is easy to implement and runs in time that scales only linearly to the number of variations.

The techniques proposed here can be further improved by considering additional characteristics of the human genome, such as the uneven distribution of allele frequencies at different loci and the structure of linkage disequilibrium (The International HapMap Consortium, 2003). Storing SNP variations using haplotypes rather than individual SNPs is likely able to reduce the data size further.

ACKNOWLEDGEMENTS

We thank David Wheeler for providing access to the sequenced JW genome and its mapping to the human reference genome.

Funding: University of California Irvine; National Science Foundation (in parts).

Conflict of Interest: none declared.

REFERENCES

Chen *et al*. (2002) DNACompress: fast and effective DNA sequence compression. *Bioinformatics*, **18**, 1696–1698.

Huffman,D.A. (1952) A method for the construction of minimum-redundancy codes. *Proc. I.R.E.*, **40**, 1098–1102.

The International HapMap Consortium (2003) The International HapMap Project. *Nature*, **426**, 789–796.

Wheeler,D.A. *et al*. (2008) The complete genome of an individual by massively parallel DNA sequencing. *Nature,* **452**, 872–876.

White,W.T.J. and Hendy,M.D. (2008) Compressing DNA sequence databases with coil. *BMC Bioinformatics*, **9**, 242.