The Sequence Alignment/Map Format and SAMtools

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th>Citation</th>
<th>Li, Heng, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25(16): 2078-2079.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Published Version</td>
<td>doi://10.1093/bioinformatics/btp352</td>
</tr>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:10246875">http://nrs.harvard.edu/urn-3:HUL.InstRepos:10246875</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>
Sequence analysis

The Sequence Alignment/Map format and SAMtools

Heng Li1,†, Bob Handsaker2,†, Alec Wysoker2, Tim Fennell2, Yue Ruan3, Nils Homer4, Gabor Marth5, Goncalo Abecasis6, Richard Durbin1,∗ and 1000 Genome Project Data Processing Subgroup7

†Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, UK, 2Broad Institute of MIT and Harvard, Cambridge, MA 02141, USA, 3Beijing Institute of Genomics, Chinese Academy of Science, Beijing 100029, China, 4Department of Computer Science, University of California Los Angeles, Los Angeles, CA 90095, 5Department of Biology, Boston College, Chestnut Hill, MA 02467, 6Center for Statistical Genetics, Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA and 7http://1000genomes.org

Received on April 28, 2009; revised on May 28, 2009; accepted on May 30, 2009

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: The Sequence Alignment/Map (SAM) format is a generic alignment format for storing read alignments against reference sequences, supporting short and long reads (up to 128 Mbp) produced by different sequencing platforms. It is flexible in style, compact in size, efficient in random access and is the format in which alignments from the 1000 Genomes Project are released. SAMtools implements various utilities for post-processing alignments in the SAM format, such as indexing, variant caller and alignment viewer, and thus provides universal tools for processing read alignments.

Availability: http://samtools.sourceforge.net
Contact: rd@sanger.ac.uk

1 INTRODUCTION

With the advent of novel sequencing technologies such as Illumina/Solexa, AB/SOLiD and Roche/454 (Mardis, 2008), a variety of new alignment tools (Langmead et al., 2009; Li et al., 2008) have been designed to realize efficient read mapping against large reference sequences, including the human genome. These tools generate alignments in different formats, however, complicating downstream processing. A common alignment format that supports all sequence types and aligners creates a well-defined interface between alignment and downstream analyses, including variant detection, genotyping and assembly.

The Sequence Alignment/Map (SAM) format is designed to achieve this goal. It supports single- and paired-end reads and combines reads of different types, including color space reads from AB/SOLiD. It is designed to scale to alignment sets of 1011 or more base pairs, which is typical for the deep resequencing of one human individual.

In this article, we present an overview of the SAM format and briefly introduce the companion SAMtools software package. A detailed format specification and the complete documentation of SAMtools are available at the SAMtools web site.

2 METHODS

2.1 The SAM format

2.1.1 Overview of the SAM format

The SAM format consists of one header section and one alignment section. The lines in the header section start with character ‘@’, and lines in the alignment section do not. All lines are TAB delimited. An example is shown in Figure 1b.

SAM, in each alignment line has 11 mandatory fields and a variable number of optional fields. The mandatory fields are briefly described in Table 1. They must be present but their value can be a ‘*’ or a zero (depending on the field) if the corresponding information is unavailable. The optional fields are presented as key-value pairs in the format of ‘TAG:TYPE:VALUE’. They store extra information from the platform or aligner. For example, the ‘RG’ tag keeps the ‘read group’ information for each read. In combination with the ‘@RG’ header lines, this tag allows each read to be labeled with metadata about its origin, sequencing center and library. The SAM format specification gives a detailed description of each field and the predefined ‘TAG’.

2.1.2 Extended CIGAR


These operations are defined in Figure 1 for example alignments. The extended CIGAR string is the space for different types of alignments.

2.1.3 Binary Alignment/Map format

To improve the performance, we designed a companion format Binary Alignment/Map (BAM), which is the binary representation of SAM and keeps exactly the same information as SAM. BAM is compressed by the BGZF library, a generic library developed by us to achieve fast random access in a zlib-compatible compressed file.

An example alignment of 112 Gbp of Illumina GA data requires 116 GB of disk space (1.0 byte per input base), including sequences, base qualities and all the meta information generated by MAQ. Most of this space is used to store the base qualities.

2.1.4 Sorting and indexing

A SAM/BAM file can be unsorted, but sorting by coordinate is used to streamline data processing and to avoid loading extra alignments into memory. A position-sorted BAM file can be indexed. We combine the UCSC binning scheme (Kent et al., 2002) and simple linear indexing to achieve fast random retrieval of alignments overlapping the

© 2009 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
to retrieve alignments in a region.

specified chromosomal region. In most cases, only one seek call is needed or a lowercase letter on the reverse strand.

denotes a base from a read mapped on the forward strand, while a comma denotes a base identical to the reference; a dot or a capital letter of reads covering the position and read bases. In the fifth field, a dot or a

consists of reference name, sorted coordinate, reference base, the number

NM

clipped sequence is not present in the sequence field. The

H

29 on the reverse strand. The hard clipping operation

number of mismatches. Read

r004

by the

N

and would typically be twice as slow as conversion. Slower than decompression. External sorting writes temporary BAM files to main memory. Conversion is slower mainly because compression with zlib is

10 h to convert from the MAQ format and 40 min to index with <30 MB

the example alignment of 112 Gbp Illumina GA data, SAMtools took about

generate per-position information in the pileup format (Fig. 1c), call SNPs and short indel variants, and show alignments in a text-based viewer. For

alignments in the SAM/BAM format. It is able to convert from other

SAM tools are two separate implementations, one in C and the other in Java, with slightly different functionality.

3 CONCLUSIONS
We designed and implemented a generic alignment format, SAM, which is simple to work with and flexible enough to keep most information from various sequencing platforms and read aligners. The equivalent binary representation, BAM, is compact in size and supports fast retrieval of alignments in specified regions. Using positional sorting and indexing, applications can perform stream-based processing on specific genomic regions without loading the entire file into memory. The SAM/BAM format, together with SAMtools, separates the alignment step from downstream analyses, enabling a generic and modular approach to the analysis of genomic sequencing data.

ACKNOWLEDGEMENTS
We are grateful to James Bonfield for the comments on indexing and to SAMtools users for testing the software as it has matured.

Funding: Wellcome Trust/077192/Z/05/Z; NIH Hapmap/1000 Genomes Project grant (U54HG002750 to B.H.).

Conflict of Interest: none declared.

REFERENCES

SAM and SAMtools

Fig. 1. Example of extended CIGAR and the pileup output. (a) Alignments of one pair of reads and three single-end reads. (b) The corresponding SAM file. The ‘@SQ’ line in the header section gives the order of reference sequences. Notably, r001 is the name of a read pair. According to FLAG 163 (=1 + 2 + 32 + 128), the read mapped to position 7 is the second read in the pair (128) and regarded as properly paired (1 + 2); its mate is mapped to 37 on the reverse strand (32). Read r002 has three soft-clipped (unaligned) bases. The coordinate shown in SAM is the position of the first aligned base. The CIGAR string for this alignment contains a ‘P’ (padding) operation which correctly aligns the inserted sequences. Padding operations can be absent when an aligner does not support multiple sequence alignment. The last six bases of read r003 map to position 9, and the first five to position 29 on the reverse strand. The hard clipping operation it indicates that the clipped sequence is not present in the sequence field. The 10t tag gives the number of mismatches. Read r004 is aligned across an intron, indicated by the ‘I’ operation. (c) Simplified pileup output by SAMtools. Each line consists of reference name, sorted coordinate, reference base, the number of reads covering the position and read bases. In the fifth field, a dot or a comma denotes a base identical to the reference; a dot or a capital letter denotes a base from a read mapped on the forward strand, while a comma or a lowercase letter on the reverse strand.

Table 1. Mandatory fields in the SAM format

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QNAME</td>
<td>Query NAME of the read or the read pair</td>
</tr>
<tr>
<td>2</td>
<td>FLAG</td>
<td>bitwise FLAG (pairing, strand, mate strand, etc.)</td>
</tr>
<tr>
<td>3</td>
<td>RNAME</td>
<td>Reference sequence NAME</td>
</tr>
<tr>
<td>4</td>
<td>POS</td>
<td>1-based leftmost POSition of clipped alignment</td>
</tr>
<tr>
<td>5</td>
<td>MAPQ</td>
<td>MQMapping Quality (Phred-scaled)</td>
</tr>
<tr>
<td>6</td>
<td>CTAG</td>
<td>Extended CIGAR string (operations: MIDSHIP)</td>
</tr>
<tr>
<td>7</td>
<td>MRNM</td>
<td>Mate Reference NaMe (‘=’ if same as RNAME)</td>
</tr>
<tr>
<td>8</td>
<td>HOPO</td>
<td>1-based lefmost Mate POsition</td>
</tr>
<tr>
<td>9</td>
<td>SIERS</td>
<td>Inferred Insert SIZE</td>
</tr>
<tr>
<td>10</td>
<td>SEQ</td>
<td>Query SEQuence on the same strand as the reference</td>
</tr>
<tr>
<td>11</td>
<td>QUAL</td>
<td>Query QUALity (ASCII-33=Phred base quality)</td>
</tr>
</tbody>
</table>

Table 1. Mandatory fields in the SAM format