Sequence analysis

The Sequence Alignment/Map format and SAMtools

Heng Li1,†, Bob Handsaker2,†, Alec Wysocki2, Tim Fennell2, Jue Ruan3, Nils Homer4, Gabor Marth5, Goncalo Abecasis6, Richard Durbin1,*, and 1000 Genome Project Data Processing Subgroup7

1Wellcome 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

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 for 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 sequencing 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.

SA1

© 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.

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.

In SAM, each alignment line has 11 mandatory fields and a variable number of optional fields. The mandatory fields are the same as the header fields, but 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 TAGs.

2.1.2 Extended CIGAR The standard CIGAR description of pairwise alignment defines three operations: "M" for match/mismatch, "I" for insertion compared with the reference and "D" for deletion. The extended CIGAR proposed in SAM added four more operations: "N" for skipped bases on the reference, "S" for soft clipping, "H" for hard clipping and "P" for padding. These support splicing, clipping, multi-part and padded alignments. Figure 1 shows examples of CIGAR strings 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 a
(a) 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 r004 map to position 9, and the first five to position 29 on the reverse strand. The hard clipping operation indicates that the clipped sequence is not present in the sequence field. The 10th tag gives the number of mismatches. Read r004 is aligned across an intron, indicated by the S 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 denotes a base from a read mapped on the reverse strand. A dot or a capital letter indicates that the clipped sequence is not present in the sequence field. The 10th tag gives the number of mismatches. Read r004 is aligned across an intron, indicated by the S 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 character on the reverse strand.

specified chromosomal region. In most cases, only one seek call is needed to retrieve alignments in a region.

### 2.2 SAMTools software package

SAMTools is a library and software package for parsing and manipulating alignments in the SAM/BAM format. It is able to convert from other alignment formats, sort and merge alignments, remove PCR duplicates, 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 the example alignment of 112 Gbp Illumina GA data, SAMTools took about 10 h to convert from the MAQ format and 40 min to index with <30 MB memory. Conversion is slower mainly because compression with zlib is slower than decompression. External sorting writes temporary BAM files and would typically be twice as slow as conversion.

### 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>MAPping Quality (Phred-scaled)</td>
</tr>
<tr>
<td>6</td>
<td>CTGAR</td>
<td>Extended CIGAR string (operations: H</td>
</tr>
<tr>
<td>7</td>
<td>HB</td>
<td>Mate Reference NaMe (‘-’ if same as RNAME)</td>
</tr>
<tr>
<td>8</td>
<td>HPOS</td>
<td>1-Based leftmost Mate POSition</td>
</tr>
<tr>
<td>9</td>
<td>ISIZ</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>QQUAL</td>
<td>Query QUALity (ASCII-33=Phred base quality)</td>
</tr>
</tbody>
</table>

SAMTools has 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


