

The Sequence Alignment/Map (SAM) Format and SAMtools

Heng Li^{1,*}, Bob Handsaker^{2,*}, Alec Wysoker², Tim Fennell², Jue Ruan³, Nils Homer⁴, Gabor Marth⁵, Goncalo Abecasis⁶, Richard Durbin^{1,†} and 1000 Genome Project Data Processing Subgroup

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, USA, ⁵Department of Biology, Boston College, Chestnut Hill, MA 02467, USA, ⁶Center for Statistical Genetics, Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA

Associate Editor: Prof. 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 128Mbp) 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 postprocessing 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 (Li et al., 2008; Langmead et al., 2009) 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 SAM format is designed to achieve this goal. It supports single-end and paired-end reads and combining reads of different types, including color space reads from AB/SOLiD. It is designed to scale to alignment sets of 10^{11} or more base-pairs, which is typical for the deep resequencing of one human individual.

In this paper, 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 website.

2 METHODS

2.1 The SAM format

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.

Table 1. Mandatory fields in the SAM format.

#	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-based leftmost POSITION of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-based leftmost Mate POSition
9	ISIZE	inferred Insert SIZE
10	SEQ	query SEQUENCE on the same strand as the reference
11	QUAL	query QUALity (ASCII-33=Phred base quality)

In SAM, 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

*The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

†To whom correspondence should be addressed.

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.

Extended CIGAR The standard CIGAR description of pairwise alignment defines three operations: 'M' for match/mismatch, 'I' for insertion compared to 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.

(a)

```

coord 12345678901234 5678901234567890123456789012345
ref    AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

r001+      TTAGATAAAGGATA*CTG
r002+      aaaAGATAA*GGATA
r003+      gcttaAGCTAA
r004+      ATAGCT.....TCAGC
r003-      tttagctTAGGC
r001-      CAGCGCCAT

(b)
@SQ SN:ref LN:45
r001 163 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTA *
r002 0 ref 9 30 35M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5H6M * 0 0 AGCTAA * NM:i:1
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 16 ref 29 30 6H5M * 0 0 TAGGC * NM:i:0
r001 83 ref 37 30 9M = 7 -39 CAGCGCCAT *

(c)
ref 7 T 1 . | ref 12 T 3 ... | ref 17 T 3 ...
ref 8 T 1 . | ref 13 A 3 ... | ref 18 A 3 ..-16..
ref 9 A 3 ... | ref 14 A 2 .+2AG. +1G | ref 19 G 2 *
ref 10 G 3 ... | ref 15 G 2 .. | ref 20 C 2 ..
ref 11 A 3 ..C | ref 16 A 3 ... | ...

```

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 in 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 H indicates that the clipped sequence is not present in the sequence field. The NM tag gives the number of mismatches. Read r004 is aligned across an intron, indicated by the N operation. (c) Simplified pileup output by SAMtools. Each line consists of reference name, sorted coordinate, reference base, the number of reads covering the position, 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.

Binary Alignment/Map (BAM) format To improve the performance, we designed a companion format 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 112Gbp of Illumina GA data requires 116GB 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.

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 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 (Figure 1c), call SNPs and short indel variants, and show alignments in a text-based viewer. For the example alignment of 112Gbp Illumina GA data, SAMtools took about 10 hours to convert from the MAQ format and 40 minutes to index with less than 30MB 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.

SAMtools has two separate implementations, one in C and the other in Java, with slightly different functionality.

3 CONCLUSION

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 R.E.H.).

REFERENCES

- Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002). The human genome browser at UCSC. *Genome Res*, 12(6):996–1006.
- Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3):R25.
- Li, H., Ruan, J., and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851–1858.
- Mardis, E. R. (2008). Next-generation DNA sequencing methods. *Annu Rev Genomics Hum Genet*, 9:387–402.