Sequence analysis

The Sequence Alignment/Map format and SAMtools

Heng Li1,†, Bob Handsaker2,†, Alec Wyssoker2, 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
Advance Access publication June 8, 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.

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 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: ‘S’ for skipped bases on the reference, ‘H’ for soft clipping, ‘N’ 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
to retrieve alignments in a region.

In most cases, only one seek call is needed to find 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 consists of reference name, sorted coordinate, reference base, the number of mismatches, and the number of 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 H (hard clipping) operation. The equivalent binary representation, BAM, is compact in size and slower than decompression. External sorting writes temporary BAM files to disk, and would typically be twice as slow as conversion.

The SAM/BAM format, together with 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 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.

**REFERENCES**


