Sequence Alignment/Map Format Specification

The SAM/BAM Format Specification Working Group

17 May 2023

The master version of this document can be found at https://github.com/samtools/hts-specs.
This printing is version 3b4e874 from that repository, last modified on the date shown above.

1 The SAM Format Specification

SAM stands for Sequence Alignment/Map format. It is a TAB-delimited text format consisting of a header section, which is optional, and an alignment section. If present, the header must be prior to the alignments. Header lines start with '@', while alignment lines do not. Each alignment line has 11 mandatory fields for essential alignment information such as mapping position, and variable number of optional fields for flexible or aligner specific information.

This specification is for version 1.6 of the SAM and BAM formats. Each SAM and BAM file may optionally specify the version being used via the @HD VN tag. For full version history see Appendix B.

SAM file contents are 7-bit US-ASCII, except for certain field values as individually specified which may contain other Unicode characters encoded in UTF-8. Alternatively and equivalently, SAM files are encoded in UTF-8 but non-ASCII characters are permitted only within certain field values as explicitly specified in the descriptions of those fields.[1]

Where it makes a difference, SAM file contents should be read and written using the POSIX / C locale. For example, floating-point values in SAM always use '.' for the decimal-point character.

The regular expressions in this specification are written using the POSIX / IEEE Std 1003.1 extended syntax. For brevity, named character classes are written as [:class:] without an additional pair of brackets.

1.1 An example

Suppose we have the following alignment with bases in lowercase clipped from the alignment. Read r001/1 and r001/2 constitute a read pair; r003 is a chimeric read; r004 represents a split alignment.

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                            ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

[1] Hence in particular SAM files must not begin with a byte order mark (BOM) and lines of text are delimited by ASCII line terminator characters only. In addition to the local platform's text file line termination conventions, implementations may wish to support LF and CR LF for interoperability with other platforms.

Phred scale Given a probability $0 < p \leq 1$, the phred scale of p equals $-10 \log_{10} p$, rounded to the closest integer.

1.2.1 Character set restrictions

Reference sequence names, CIGAR strings, and several other field types are used as values or parts of values of other fields in SAM and related formats such as VCF. To ensure that these other fields' representations are unambiguous, these field types disallow particular delimiter characters.

Query or read names may contain any printable ASCII characters in the range [!-~] apart from ‘@’, so that SAM alignment lines can be easily distinguished from header lines. (They are also limited in length.)

Reference sequence names may contain any printable ASCII characters in the range [!-~] apart from backslashes, commas, quotation marks, and brackets—i.e., apart from ‘\ , "‘ ’ () [] {} <>’—and may not start with ‘*’ or ‘=’.[4]

Thus they match the following regular expression:

```
[0-9A-Za-z!#$%&+./:;?@^_|~-][0-9A-Za-z!#$%&*+./:;=?@^_|~-]*
```

For clarity, elsewhere in this specification we write this set of allowed characters as a character class [:rname:] and extend the POSIX regular expression notation to use ^*= to indicate the omission of ‘*’ and ‘=’ from the character class. Thus this regular expression can be written more clearly as [:rname:^*=][:rname:]*.

1.3 The header section

Each header line begins with the character ‘@’ followed by one of the two-letter header record type codes defined in this section. In the header, each line is TAB-delimited and, apart from @CO lines, each data field follows a format ‘TAG:VALUE’ where TAG is a two-character string that defines the format and content of VALUE. Thus header lines match /^@(HD|SQ|RG|PG)(\t[A-Za-z][A-Za-z0-9]:[-:print:]+)+$/ or /^@CO\t.*/.[5] Within each (non-@CO) header line, no field tag may appear more than once and the order in which the fields appear is not significant.

The following table describes the header record types that may be used and their predefined tags. Tags listed with ‘*’ are required; e.g., every @SQ header line must have SN and LN fields. As with alignment optional fields (see Section 1.5), you can freely add new tags for further data fields. Tags containing lowercase letters are reserved for local use and will not be formally defined in any future version of this specification.[6]

Tag	Description
@HD	File-level metadata. Optional. If present, there must be only one @HD line and it must be the first line of the file.
VN*	Format version. *Accepted format*: /^[0-9]+\.[0-9]+$/.

[3] Chimeric alignments are primarily caused by structural variations, gene fusions, misassemblies, RNA-seq or experimental protocols. They are more frequent given longer reads. For a chimeric alignment, the linear alignments constituting the alignment are largely non-overlapping; each linear alignment may have high mapping quality and is informative in SNP/INDEL calling. In contrast, multiple mappings are caused primarily by repeats. They are less frequent given longer reads. If a read has multiple mappings, all these mappings are almost entirely overlapping with each other; except the single-best optimal mapping, all the other mappings get mapping quality <Q3 and are ignored by most SNP/INDEL callers.

[4] Characters that are *not* disallowed include ‘|’, which historically appeared in reference names derived from NCBI FASTA files, and ‘:’, which appears in HLA allele names. Appendix A describes approaches for parsing *name*[:*begin*-*end*] region notation unambiguously even though *name* may itself contain colons.

[5] [:print:] indicates that header field values contain printable characters, i.e., non-control characters. For fields limited to ASCII, which is the majority, this is equivalent to [-~].

[6] Best practice is to use lowercase tags while designing and experimenting with new data field tags or for fields of local interest only. For new tags that are of general interest, raise an hts-specs issue or email samtools-devel@lists.sourceforge.net to have an uppercase equivalent added to the specification. This way collisions of the same uppercase tag being used with different meanings can be avoided.