Sequence Alignment/Map Optional Fields Specification

The SAM/BAM Format Specification Working Group

14 Jul 2017

The master version of this document can be found at https://github.com/samtools/hts-specs. This printing is version dab57f4 from that repository, last modified on the date shown above.

This document is a companion to the Sequence Alignment/Map Format Specification that defines the SAM and BAM formats, and to the CRAM Format Specification that defines the CRAM format. Alignment records in each of these formats may contain a number of optional fields, each labelled with a tag identifying that field’s data. This document describes each of the predefined standard tags, and discusses conventions around creating new tags.

1 Standard tags

Predefined standard tags are listed in the following table and described in greater detail in later subsections. Optional fields are usually displayed as TAG:TYPE:VALUE; the type may be one of A (character), B (general array), f (real number), H (hexadecimal array), i (integer), or Z (string).

<table>
<thead>
<tr>
<th>Tag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>i</td>
<td>The smallest template-independent mapping quality of segments in the rest</td>
</tr>
<tr>
<td>AS</td>
<td>i</td>
<td>Alignment score generated by aligner</td>
</tr>
<tr>
<td>BC</td>
<td>Z</td>
<td>Barcode sequence identifying the sample</td>
</tr>
<tr>
<td>BQ</td>
<td>Z</td>
<td>Offset to base alignment quality (BAQ)</td>
</tr>
<tr>
<td>BZ</td>
<td>Z</td>
<td>Phred quality of the unique molecular barcode bases in the OX tag</td>
</tr>
<tr>
<td>CC</td>
<td>Z</td>
<td>Reference name of the next hit</td>
</tr>
<tr>
<td>CG</td>
<td>B,I</td>
<td>BAM only: CIGAR in BAM’s binary encoding if (and only if) it consists of &gt;65535 operators</td>
</tr>
<tr>
<td>CM</td>
<td>i</td>
<td>Edit distance between the color sequence and the color reference (see also NM)</td>
</tr>
<tr>
<td>CQ</td>
<td>Z</td>
<td>Color read base qualities</td>
</tr>
<tr>
<td>CS</td>
<td>Z</td>
<td>Color read sequence</td>
</tr>
<tr>
<td>CT</td>
<td>Z</td>
<td>Complete read annotation tag, used for consensus annotation dummy features</td>
</tr>
<tr>
<td>E2</td>
<td>Z</td>
<td>The 2nd most likely base calls</td>
</tr>
<tr>
<td>FI</td>
<td>i</td>
<td>The index of segment in the template</td>
</tr>
<tr>
<td>FS</td>
<td>Z</td>
<td>Segment suffix</td>
</tr>
<tr>
<td>FZ</td>
<td>B,S</td>
<td>Flow signal intensities</td>
</tr>
<tr>
<td>GC</td>
<td>?</td>
<td>Reserved for backwards compatibility reasons</td>
</tr>
<tr>
<td>GQ</td>
<td>?</td>
<td>Reserved for backwards compatibility reasons</td>
</tr>
<tr>
<td>GS</td>
<td>?</td>
<td>Reserved for backwards compatibility reasons</td>
</tr>
<tr>
<td>H0</td>
<td>i</td>
<td>Number of perfect hits</td>
</tr>
<tr>
<td>H1</td>
<td>i</td>
<td>Number of 1-difference hits (see also NM)</td>
</tr>
<tr>
<td>H2</td>
<td>i</td>
<td>Number of 2-difference hits</td>
</tr>
<tr>
<td>HI</td>
<td>i</td>
<td>Query hit index</td>
</tr>
<tr>
<td>IH</td>
<td>i</td>
<td>Number of stored alignments in SAM that contains the query in the current record</td>
</tr>
</tbody>
</table>

1.1 Additional Template and Mapping data

**AM:i** The smallest template-independent mapping quality of segments in the rest.

**AS:i** Alignment score generated by aligner.

**BQ:Z:qualities** Offset to base alignment quality (BAQ), of the same length as the read sequence. At the \(i\)-th read base, \(BAQ_i = Q_i - (BQ_i - 64)\) where \(Q_i\) is the \(i\)-th base quality.

**CC:Z:** Reference name of the next hit; '=' for the same chromosome.

**CG:B:encodedCigar** Real CIGAR in its binary form if (and only if) it contains >65535 operations. This is a BAM file only tag as a workaround of BAM’s incapability to store long CIGARs in the standard way. SAM and CRAM files created with updated tools aware of the workaround are not expected to contain this tag. See also the footnote in Section 4.2 of the SAM spec for details.

**CP:i** Leftmost coordinate of the next hit.

**E2:Z:bases** The 2nd most likely base calls. Same encoding and same length as SEQ. See also U2 for associated quality values.

**FI:i** The index of segment in the template.
FS:Z: str Segment suffix.
H0:i: count Number of perfect hits.
H1:i: count Number of 1-difference hits (see also NM).
H2:i: count Number of 2-difference hits.
HI:i: i Query hit index, indicating the alignment record is the i-th one stored in SAM.
IH:i: count Number of stored alignments in SAM that contains the query in the current record.
MC:Z: cigar CIGAR string for mate/next segment.
   The MD field aims to achieve SNP/indel calling without looking at the reference. For example, a string ‘10A5^AC6’ means from the leftmost reference base in the alignment, there are 10 matches followed by an A on the reference which is different from the aligned read base; the next 5 reference bases are matches followed by a 2bp deletion from the reference; the deleted sequence is AC; the last 6 bases are matches. The MD field ought to match the CIGAR string.
MQ:i: Mapping quality of the mate/next segment.
NH:i: Number of reported alignments that contains the query in the current record.
NM:i: Edit distance to the reference, including ambiguous bases but excluding clipping.
PQ:i: Phred likelihood of the template, conditional on both the mapping being correct.
Q2:Z:qualities Phred quality of the mate/next segment sequence in the R2 tag. Same encoding as QUAL.
R2:Z:bases Sequence of the mate/next segment in the template. See also Q2 for any associated quality values.
SA:Z:(rname,pos,strand,CIGAR,mapQ,NM;;) Other canonical alignments in a chimeric alignment, formatted as a semicolon-delimited list. Each element in the list represents a part of the chimeric alignment. Conventionally, at a supplementary line, the first element points to the primary line.
SM:i: Template-independent mapping quality.
TC:i: The number of segments in the template.
U2:Z: Phred probability of the 2nd call being wrong conditional on the best being wrong. The same encoding and length as QUAL. See also E2 for associated base calls.
UQ:i: Phred likelihood of the segment, conditional on the mapping being correct.

1.2 Metadata
RG:Z:readgroup The read group to which the read belongs. If @RG headers are present, then readgroup must match the RG-ID field of one of the headers.
LB:Z:library The library from which the read has been sequenced. If @RG headers are present, then library must match the RG-LB field of one of the headers.
PG:Z: Program. Value matches the header PG-ID tag if @PG is present.
PU:Z:platformunit The platform unit in which the read was sequenced. If @RG headers are present, then platformunit must match the RG-PU field of one of the headers.
CO:Z:text Free-text comments.
1.3 Barcodes

**BC:Z:sequence** Barcode sequence (Identifying the sample/library), with any quality scores (optionally) stored in the QT tag. The BC tag should match the QT tag in length. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the barcodes and places a hyphen (‘-’) between the barcodes from the same template.

**QT:Z:qualities** Phred quality of the sample-barcode sequence in the BC (or RT) tag. Same encoding as QUAL, i.e., Phred score + 33. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the quality strings with spaces (‘ ’) between the different strings from the same template.

**RX:Z:sequence+** Sequence bases from the unique molecular identifier. These could be either corrected or uncorrected. Unlike MI, the value may be non-unique in the file. Should be comprised of a sequence of bases. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the barcodes with a hyphen (‘-’) between the different barcodes.

If the bases represent corrected bases, the original sequence can be stored in OX (similar to OQ storing the original qualities of bases.)

**QX:Z:qualities+** Phred quality of the unique molecular identifier sequence in the RX tag. Same encoding as QUAL, i.e., Phred score + 33. The qualities here may have been corrected (Raw bases and qualities can be stored in OX and BZ respectively.) The lengths of the QX and the RX tags must match. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the quality strings with a space (‘ ’) between the different strings.

**MI:Z:str** Molecular Identifier. A unique ID within the SAM file for the source molecule from which this read is derived. All reads with the same MI tag represent the group of reads derived from the same source molecule.

**OX:Z:sequence+** Raw (uncorrected) unique molecular identifier bases, with any quality scores (optionally) stored in the BZ tag. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the barcodes with a hyphen (‘-’) between the different barcodes.

**BZ:Z:qualities+** Phred quality of the (uncorrected) unique molecular identifier sequence in the OX tag. Same encoding as QUAL, i.e., Phred score + 33. The OX tags should match the BZ tag in length. In the case of multiple unique molecular identifiers (e.g., one on each end of the template) the recommended implementation concatenates all the quality strings with a space (‘ ’) between the different strings.

**RT:Z:sequence** Deprecated alternative to BC tag originally used at Sanger.

1.4 Original data

**OC:Z:cigar** Original CIGAR, usually before realignment.

**OP:i:pos** Original mapping position, usually before realignment.

**OQ:Z:qualities** Original base quality, usually before recalibration. Same encoding as QUAL.

1.5 Annotation and Padding

**CT:Z:strand;type (;key (=value))** Complete read annotation tag, used for consensus annotation dummy features.

The CT tag is intended primarily for annotation dummy reads, and consists of a strand, type and zero or more key=value pairs, each separated with semicolons. The strand field has four values as in GFF3, and supplements FLAG bit 0x10 to allow unstranded (‘.’), and stranded but unknown strand (‘?’)
annotation. For these and annotation on the forward strand (strand set to ‘+’), do not set FLAG bit 0x10. For annotation on the reverse strand, set the strand to ‘-’ and set FLAG bit 0x10.

The type and any keys and their optional values are all percent encoded according to RFC3986 to escape meta-characters ‘=’, ‘%’, ‘;’, ‘|’ or non-printable characters not matched by the isprint() macro (with the C locale). For example a percent sign becomes ‘%2C’.

PT:Z:start;end;strand;type(;key=value)*(|start;end;strand;type(;key=value)))* Read annotations for parts of the padded read sequence.

The PT tag value has the format of a series of tags separated by ‘|’, each annotating a sub-region of the read. Each tag consists of start, end, strand, type and zero or more key=value pairs, each separated with semicolons. Start and end are 1-based positions between one and the sum of the M/I/D/P/S/=/X CIGAR operators, i.e. SEQ length plus any pads. Note any editing of the CIGAR string may require updating the ‘PT’ tag coordinates, or even invalidate them. As in GFF3, strand is one of ‘+‘ for forward strand tags, ‘-‘ for reverse strand, ‘.‘ for unstranded or ‘?’ for stranded but unknown strand. The type and any keys and their optional values are all percent encoded as in the CT tag.

1.6 Technology-specific data

FZ:B,S:intensities Flow signal intensities on the original strand of the read, stored as (uint16_t) round(value * 100.0).

1.6.1 Color space

CM:i:distance Edit distance between the color sequence and the color reference (see also NM).

CS:Z:sequence Color read sequence on the original strand of the read. The primer base must be included.

CQ:Z:qualities Color read quality on the original strand of the read. Same encoding as QUAL; same length as CS.

2 Locally-defined tags

You can freely add new tags. Note that tags starting with ‘X’, ‘Y’, or ‘Z’ and tags containing lowercase letters in either position are reserved for local use and will not be formally defined in any future version of this specification.

If a new tag may be of general interest, it may be useful to have it added to this specification. Additions can be proposed by opening a new issue at https://github.com/samtools/hts-specs/issues and/or by sending email to samtools-devel@lists.sourceforge.net.

Appendix A SAM Tags History

This lists the date of each tagged SAM version along with changes that have been made while that version was current.

1.5: 23 May 2013 to current

- Add UMI-related tags (RX, QX, OX, BZ, MI) and clarified usage of sample barcode tag BC. (August 2017)
- SAMtags.txt (this file) created with tags from SAMv1