Sequence Alignment/Map Optional Fields Specification

The SAM/BAM Format Specification Working Group

4 Apr 2017

The master version of this document can be found at `https://github.com/samtools/hts-specs`.
This printing is version 434cda9 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.[1] 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).

Tag	Type	Description
AM	i	The smallest template-independent mapping quality of segments in the rest
AS	i	Alignment score generated by aligner
BC	Z	Barcode sequence
BQ	Z	Offset to base alignment quality (BAQ)
CC	Z	Reference name of the next hit
CM	i	Edit distance between the color sequence and the color reference (see also `NM`)
CO	Z	Free-text comments
CP	i	Leftmost coordinate of the next hit
CQ	Z	Color read base qualities
CS	Z	Color read sequence
CT	Z	Complete read annotation tag, used for consensus annotation dummy features.
E2	Z	The 2nd most likely base calls
FI	i	The index of segment in the template
FS	Z	Segment suffix
FZ	B,S	Flow signal intensities
GC	?	Reserved for backwards compatibility reasons
GQ	?	Reserved for backwards compatibility reasons
GS	?	Reserved for backwards compatibility reasons
H0	i	Number of perfect hits
H1	i	Number of 1-difference hits (see also `NM`)
H2	i	Number of 2-difference hits
HI	i	Query hit index
IH	i	Number of stored alignments in SAM that contains the query in the current record
LB	Z	Library
MC	Z	CIGAR string for mate/next segment
MD	Z	String for mismatching positions

---

[1]See `SAMv1.pdf` and `CRAMv3.pdf` at `https://github.com/samtools/hts-specs`.

MF	?	Reserved for backwards compatibility reasons
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
OC	Z	Original CIGAR
OP	i	Original mapping position
OQ	Z	Original base quality
PG	Z	Program
PQ	i	Phred likelihood of the template
PT	Z	Read annotations for parts of the padded read sequence
PU	Z	Platform unit
QT	Z	Barcode (BC or RT) phred-scaled base qualities
Q2	Z	Phred quality of the mate/next segment sequence in the R2 tag
R2	Z	Sequence of the mate/next segment in the template
RG	Z	Read group
RT	Z	Barcode sequence (deprecated; use BC instead)
SA	Z	Other canonical alignments in a chimeric alignment
SM	i	Template-independent mapping quality
SQ	?	Reserved for backwards compatibility reasons
S2	?	Reserved for backwards compatibility reasons
TC	i	The number of segments in the template
U2	Z	Phred probility of the 2nd call being wrong conditional on the best being wrong
UQ	i	Phred likelihood of the segment, conditional on the mapping being correct
X?	?	Reserved for end users
Y?	?	Reserved for end users
Z?	?	Reserved for end users

1.1 Additional Template and Mapping data

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

AS:i:`score` 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, $\text{BAQ}_i = Q_i - (\text{BQ}_i - 64)$ where Q_i is the i-th base quality.

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

CP:i:`pos` 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:`int` 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.

MD:Z:[0-9]+(([A-Z]|\^[A-Z]+)[0-9]+)* String for mismatching positions.

> 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 probility 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, with any quality scores stored in the QT tag.

QT:Z:qualities Phred quality of the barcode sequence in the BC (or RT) tag. Same encoding as QUAL.

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`.