Tag	Type	Description
H1	i	Number of 1-difference hits (see also NM)
H2	i	Number of 2-difference hits
HI	i	Query hit index
IH	i	Query hit total count
LB	Z	Library
MC	Z	CIGAR string for mate/next segment
MD	Z	String encoding mismatched and deleted reference bases
MF	?	Reserved for backwards compatibility reasons
MI	Z	Molecular identifier; a string that uniquely identifies the molecule from which the record was derived
ML	B,C	Base modification probabilities
MM	Z	Base modifications / methylation
MN	i	Length of sequence at the time MM and ML were produced
MQ	i	Mapping quality of the mate/next segment
NH	i	Number of reported alignments that contain the query in the current record
NM	i	Edit distance to the reference
OA	Z	Original alignment
OC	Z	Original CIGAR (deprecated; use OA instead)
OP	i	Original mapping position (deprecated; use OA instead)
OQ	Z	Original base quality
OX	Z	Original unique molecular barcode bases
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
Q2	Z	Phred quality of the mate/next segment sequence in the R2 tag
QT	Z	Phred quality of the sample barcode sequence in the BC tag
QX	Z	Quality score of the unique molecular identifier in the RX tag
R2	Z	Sequence of the mate/next segment in the template
RG	Z	Read group
RT	?	Reserved for backwards compatibility reasons
RX	Z	Sequence bases of the (possibly corrected) unique molecular identifier
S2	?	Reserved for backwards compatibility reasons
SA	Z	Other canonical alignments in a chimeric alignment
SM	i	Template-independent mapping quality
SQ	?	Reserved for backwards compatibility reasons
TC	i	The number of segments in the template
TS	A	Transcript strand
U2	Z	Phred probability 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:`score` The smallest template-independent mapping quality of any segment in the same template as this read. (See also SM.)

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, $\mathrm{BAQ}_i = Q_i - (\mathrm{BQ}_i - 64)$ where Q_i is the i-th base quality.

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

PT:Z:*annotag*`(\|`*annotag*`)*` where each *annotag* matches *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 annotation 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` ~~`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`.

1.7 Base modifications

Base modifications, including base methylation, are represented as a series of edits from the primary unmodified sequence as originally reported by the sequencing instrument. This potentially differs to the sequence stored in the main SAM SEQ field if the latter has been reverse complemented, in which case SAM FLAG 0x10 must be set. This means modification positions are also recorded against the original orientation (i.e. starting at the 5' end), and count the original base types.

Each modified base prediction listed also has a quality value associated with it. Given the unmodified base already has a phred likelihood, this base modification quality should be interpreted as the likelihood of this modification being correct given an assumption the original call is correct.

MM:Z:`([ACGTUN][-+]([a-z]+|[0-9]+)[.?]?(,[0-9]+)*;)*`
The first character is the unmodified "fundamental" base as reported by the sequencing instrument for the top strand. It must be one of '`A`', '`C`', '`G`', '`T`', '`U`' (if RNA) or '`N`' for anything else, including any IUPAC ambiguity codes in the reported SEQ field. Note '`N`' may be used to match any base rather than specifically an '`N`' call by the sequencing instrument. This may be used in situations where the base modification is not a derivation of a standard base type. This is followed by either plus or minus indicating the strand the modification was observed on (relative to the original sequenced strand of SEQ with plus meaning same orientation),[3] and one or more base modification codes.

Following the base modification codes is a recommended but optional '`.`' or '`?`' describing how skipped seq bases of the stated base type should be interpreted by downstream tools. When this flag is '`?`' there is no information about the modification status of the skipped bases provided. When this flag is not present, or it is '`.`', these bases should be assumed to have low probability of modification.[4]

This is then followed by a comma separated list of how many seq bases of the stated base type to skip, stored as a delta to the last and starting with 0 as the first (or next) base, starting from the

[3] Hence a tool that may reverse complement sequences does not need to understand how to manipulate the `MM` and `ML` tags.

[4] The decision whether a base is assumed to be unmodified or has a probability explicitly provided is up to the modification calling program. Some programs will elide calls with modification probabilities below a threshold to provide a more compact modification tag.

Unmodified base	Code	Abbreviation	Name	ChEBI
C	m	5mC	5-Methylcytosine	27551
C	h	5hmC	5-Hydroxymethylcytosine	76792
C	f	5fC	5-Formylcytosine	76794
C	c	5caC	5-Carboxylcytosine	76793
C	C		Ambiguity code; any C mod	
T	g	5hmU	5-Hydroxymethyluracil	16964
T	e	5fU	5-Formyluracil	80961
T	b	5caU	5-Carboxyluracil	17477
T	T		Ambiguity code; any T mod	
U	U		Ambiguity code; any U mod	
A	a	6mA	6-Methyladenine	28871
A	A		Ambiguity code; any A mod	
G	o	8oxoG	8-Oxoguanine	44605
G	G		Ambiguity code; any G mod	
N	n	Xao	Xanthosine	18107
N	N		Ambiguity code; any mod	

ML:B:C,`scaled-probabilities`

The optional `ML` tag lists the probability of each modification listed in the `MM` tag being correct, in the order that they occur. The continuous probability range 0.0 to 1.0 is remapped in equal sized portions to the discrete integers 0 to 255 inclusively. Thus the probability range corresponding to integer value N is $N/256$ to $(N+1)/256$.

The SAM encoding therefore uses a byte array of type '`C`' with the number of elements matching the summation of the number of modifications listed as being present in the `MM` tag accounting for multi-modifications each having their own probability.

For example '`MM:Z:C+m,5,12;C+h,5,12;`' may have an associated tag of '`ML:B:C,204,89,26,130`'.

If the above is rewritten in the multiple-modification form, the probabilities are interleaved in the order presented, giving '`MM:Z:C+mh,5,12; ML:B:C,204,26,89,130`'. Note where several possible modifications are presented at the same site, the `ML` values represent the absolute probabilities of the modification call being correct and not the relative likelihood between the alternatives. These probabilities should not sum to above 1.0 ($\approx$ 256 in integer encoding, allowing for some minor rounding errors), but may sum to a lower total with the remainder representing the probability that none of the listed modification types are present. In the example used above, the 6th `C` has 80% chance of being `5mC`, 10% chance of being `5hmC` and 10% chance of being an unmodified `C`.

`ML` values for ambiguity codes give the probability that the modification is one of the possible codes compatible with that ambiguity code. For example `MM:Z:C+C,10; ML:B:C,229` indicates a C call with a probability of 90% of having some form of unspecified modification.

MN:i:`length`

The length of the SEQ field at the time the `MM` value was last written.

Some processing of aligned data, such as the use of hard-clipping tools, may alter SEQ sequence data. If the sequence is shortened in this manner then the base offsets in `MM` and `ML` become invalid unless they are also updated accordingly.

Some hard-clipping tools will update `MM`/`ML` but others do not, so the `MN` tag offers a simple sanity check. Software that wishes to validate `MM` should compare the length of the SEQ field with the contents of the `MN` tag—if they differ, the `MM` and `ML` values should be considered out-of-date. The tag is optional, but recommended, and if it is absent then there is an implicit assumption that the `MM` data is valid unless evidence implies otherwise (e.g., by having coordinates beyond the end of the sequence).

2 Draft tags

These are tags which have been proposed and are broadly accepted to become standard tags, but a review or probationary period has been deemed useful. They use the locally-defined tag namespace and processing software should consider that the tags may have local usage for other purposes.

There are currently no tags with draft status.

3 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 Tag History

This appendix lists when standard tags were initially defined or significantly changed, and other historical events that affect how tags are interpreted or what files they may appear in.

September 2024

Added the MN tag for validating base modification tag consistency.

February 2022

Base modification tags changed to use the predefined standard names MM and ML, as their review period has finished. Programs outputting the draft Mm and Ml tags should be changed to use MM and ML instead.

December 2021

Amended draft Mm tag to provide hints about the modification status of skipped sequence bases.

July 2021

Added the Mm and Ml draft tags describing base modifications.

March 2020

Transcript strand tag TS added, equivalent to the locally-defined XS tag produced by several RNA aligners.

January 2019

Added the OA tag for recording original/previous alignment information.
Deprecated the OC and OP tags.

July 2018

Clarified the calculation of NM score.

May 2018

Cellular barcode tags CB, CR, and CY added.
Removed the RT:Z tag, which was a long-deprecated synonym for BC.