

the definitions of the template mapped start and end. Thus the exact definitions are implementation-defined.¹⁵

10. **SEQ**: segment SEQUENCE. This field can be a '*' when the sequence is not stored. If not a '*', the length of the sequence must equal the sum of lengths of M/I/S/=/X operations in CIGAR. An '=' denotes the base is identical to the reference base. No assumptions can be made on the letter cases.
11. **QUAL**: ASCII of base QUALity plus 33 (same as the quality string in the Sanger FASTQ format). A base quality is the phred-scaled base error probability which equals $-10 \log_{10} \Pr\{\text{base is wrong}\}$. This field can be a '*' when quality is not stored.¹⁶ If not a '*', SEQ must not be a '*' and the length of the quality string ought to equal the length of SEQ.

1.5 The alignment section: optional fields

All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string that matches `/[A-Za-z][A-Za-z0-9]/`. Within each alignment line, no TAG may appear more than once and the order in which the optional fields appear is not significant. A TAG containing lowercase letters is reserved for end users. In an optional field, TYPE is a single case-sensitive letter which defines the format of VALUE:

Type	Regexp matching VALUE	Description
A	[!~]	Printable character
i	[+]?[0-9]+	Signed integer ¹⁷
f	[+]?[0-9]*\.[0-9]+([eE][+]?[0-9]+)?	Single-precision floating number
Z	[!~]*	Printable string, including space
H	([0-9A-F][0-9A-F])*	Byte array in the Hex format ¹⁸
B	[cCsSiIf](,[+]?[0-9]*\.[0-9]+([eE][+]?[0-9]+)?)*	Integer or numeric array

For an integer or numeric array (type 'B'), the first letter indicates the type of numbers in the following comma separated array. The letter can be one of 'cCsSiIf', corresponding to `int8_t` (signed 8-bit integer), `uint8_t` (unsigned 8-bit integer), `int16_t`, `uint16_t`, `int32_t`, `uint32_t` and `float`, respectively.¹⁹ During import/export, the element type may be changed if the new type is also compatible with the array.

Predefined tags are described in the separate *Sequence Alignment/Map Optional Fields Specification*.²⁰ See that document for details of existing standard tag fields and conventions around creating new tags that may be of general interest. 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 these specifications.

¹⁵The earliest versions of this specification used 5' to 5' (in original orientation, TLEN#1; dashed parts of the reads indicate soft-clipped bases) while later ones used leftmost to rightmost mapped base (TLEN#2). Note: these two definitions agree in most alignments, but differ in the case of overlaps where the first segment aligns beyond the start of the last segment.



¹⁶Note an ambiguity exists for the unlikely case of a non-'*' sequence of length 1 with base quality 9 (ASCII '*'). Given the ambiguity, the quality should always be deemed to be unavailable in this scenario.

¹⁷The number of digits in an integer optional field is not explicitly limited in SAM. However, BAM can represent values in the range $[-2^{31}, 2^{32})$, so in practice this is the realistic range of values for SAM's 'i' as well.

¹⁸For example, the six-character Hex string '1AE301' represents the byte array `[0x1a, 0xe3, 0x1]`.

¹⁹Explicit typing eases format parsing and helps to reduce the file size when SAM is converted to BAM.

²⁰See `SAMtags.pdf` at <https://github.com/samtools/hts-specs>.