	PL	Platform/technology used to produce the reads. *Valid values*: CAPILLARY, DNBSEQ (MGI/BGI), ELEMENT, HELICOS, ILLUMINA, IONTORRENT, LS454, ONT (Oxford Nanopore), PACBIO (Pacific Biosciences), SOLID, and ULTIMA. This field should be omitted when the technology is not in this list (though the PM field may still be present in this case) or is unknown. The values should be written as described in uppercase, however due to the existance of public data with lowercase values tools should also accept lowercase when decoding.
	PM	Platform model. Free-form text providing further details of the platform/technology used.
	PU	Platform unit (e.g., flowcell-barcode.lane for Illumina or slide for SOLiD). Unique identifier.
	SM	Sample. Use pool name where a pool is being sequenced.
@PG		Program.
	ID*	Program record identifier. Each @PG line must have a unique ID. The value of ID is used in the alignment PG tag and PP tags of other @PG lines. PG IDs may be modified when merging SAM files in order to handle collisions.
	PN	Program name
	CL	Command line. UTF-8 encoding may be used.
	PP	Previous @PG-ID. Must match another @PG header's ID tag. @PG records may be chained using PP tag, with the last record in the chain having no PP tag. This chain defines the order of programs that have been applied to the alignment. PP values may be modified when merging SAM files in order to handle collisions of PG IDs. The first PG record in a chain (i.e., the one referred to by the PG tag in a SAM record) describes the most recent program that operated on the SAM record. The next PG record in the chain describes the next most recent program that operated on the SAM record. The PG ID on a SAM record is not required to refer to the newest PG record in a chain. It may refer to any PG record in a chain, implying that the SAM record has been operated on by the program in that PG record, and the program(s) referred to via the PP tag.
	DS	Description. UTF-8 encoding may be used.
	VN	Program version
@CO		One-line text comment. Unordered multiple @CO lines are allowed. UTF-8 encoding may be used.

1.3.1 Defined sub-sort terms

While the SS sub-sort field allows implementation-defined keywords, some terms are predefined with specific meanings.

lexicographical sort order is defined as a character-based dictionary sort with the character order as defined by the POSIX C locale. For example "abc", "abc17", "abc5", "abc59" and "abcd" are in lexicographical order.

natural sort order is similar to lexicographical order except that runs of adjacent digits are considered to be numbers embedded within the text string, ordered numerically when compared to each other and ordered as single digits when compared to the surrounding non-digit characters. Runs that differ only in the number of leading zeros (thus are numerically tied) are ordered by more-zeros coming before fewer-zeros. The characters '-' and '.' are considered as ordinary characters, so apparently negative or fractional values are not treated as part of an embedded number. For example, "abc", "abc+5", "abc-5", "abc.d", "abc03", "abc5", "abc008", "abc08", "abc8", "abc17", "abc17.+", "abc17.2", "abc17.d", "abc59" and "abcd" are in natural order.

umi is a lexicographical sort by the UMI tag. The MI tag should be used for comparing UMIs. The RX tag may be used in its absence but is not guaranteed to be unique across multiple libraries.

1.3.2 Reference MD5 calculation

The M5 tag on @SQ lines allows reference sequences to be uniquely identified through the MD5 digest of the sequence itself. As the digest is based on the sequence and nothing else, it can help resolve ambiguities with