Alignment Schema

A high level representation of the alignment looks like this:

id *String*	The read alignment ID. This ID is unique within the read group this alignment belongs to. For performance reasons, this field may be omitted by a backend. If provided, its intended use is to make caching and UI display easier for genome browsers and other lightweight clients.		
readGroupId *String*	The ID of the read group this read belongs to. Every read must belong to exactly one read group.		
fragmentName *String*	The fragment name. Equivalent to QNAME (query template name) in SAM.		
improperPlacement *boolean*	The orientation and the distance between reads from the fragment are inconsistent with the sequencing protocol (inverse of SAM flag 0x2).		
duplicateFragment *boolean*	The fragment is a PCR or optical duplicate (SAM flag 0x400).		
numberReads *int*	The number of reads in the fragment (extension to SAM flag 0x1).		
fragmentLength *int*	The observed length of the fragment, equivalent to TLEN in SAM.		
readNumber *int*	he read ordinal in the fragment, 0-based and less than numberReads. This field replaces SAM flag 0x40 and 0x80 and is intended to more cleanly represent multiple reads per fragment.		
failedVendorQualityChecks *boolean*	The read fails platform or vendor quality checks (SAM flag 0x200).		

alignment

LinearAlignment

A linear alignment describes the alignment of a read to a Reference, using a position and CIGAR array.

position

Position

The position of this alignment: an unoriented base in some Reference. A Position is represented by a reference name and a base number on that reference (0-based).

referenceName *String*	The name of the Reference on which the Position is located.
position *long*	The 0-based offset from the start of the forward strand for that Reference. Genomic positions are non-negative integers less than Reference length.
strand *Strand*	Indicates the DNA strand associate for some data item. **NEG_STRAND**: The negative (-) strand. **POS_STRAND**: The postive (+) strand.

mappingQuality

int

The mapping quality of this alignment, meaning the likelihood that the read maps to this position. Specifically, this is -10 log10 Pr(mapping position is wrong), rounded to the nearest integer.

cigar

List<CigarUnit>

A list of instances of CIGAR operations, i.e.: it represents the local alignment of this sequence (alignment matches, indels, etc) versus the reference.

operation *CigarOperation*	An enum for the different types of CIGAR alignment operations that exist.\\nUsed wherever CIGAR alignments are used. The different enumerated values\\nhave the following usage: **ALIGNMENT_MATCH**: An alignment match indicates that a sequence can be aligned to the reference without evidence of an INDEL. Unlike the SEQUENCE_MATCH and SEQUENCE_MISMATCH operators, the ALIGNMENT_MATCH operator does not indicate whether the reference and read sequences are an exact match. This operator is equivalent to SAM's M. **INSERT**: The insert operator indicates that the read contains evidence of bases being inserted into the reference. This operator is equivalent to SAM's I. **DELETE**: The delete operator indicates that the read contains evidence of bases being deleted from the reference. This operator is equivalent to SAM's D. **SKIP**: The skip operator indicates that this read skips a long segment of the reference, but the bases have not been deleted. This operator is commonly used when working with RNA-seq data, where reads may skip long segments of the reference between exons. This operator is equivalent to SAM's N. **CLIP_SOFT**: The soft clip operator indicates that bases at the start/end of a read have not been considered during alignment. This may occur if the majority of a read maps, except for low quality bases at the start/end of a read. This operator is equivalent to SAM's S. Bases that are soft clipped will still be stored in the read. **CLIP_HARD**: The hard clip operator indicates that bases at the start/end of a read have been omitted from this alignment. This may occur if this linear alignment is part of a chimeric alignment, or if the read has been trimmed (e.g., during error correction, or to trim poly-A tails for RNA-seq). This operator is equivalent to SAM's H. **PAD**: The pad operator indicates that there is padding in an alignment. This operator is equivalent to SAM's P. **SEQUENCE_MATCH**: This operator indicates that this portion of the aligned sequence exactly matches the reference (e.g., all bases are equal to the reference bases). This operator is equivalent to SAM's =. **SEQUENCE_MISMATCH**: This operator indicates that this portion of the aligned sequence is an alignment match to the reference, but a sequence mismatch (e.g., the bases are not equal to the reference). This can indicate a SNP or a read error. This operator is equivalent to SAM's X.
operationLength *long*	The number of bases that the operation runs for.
referenceSequence *String*	It is only used at mismatches (SEQUENCE_MISMATCH) and deletions (DELETE). Filling this field replaces the MD tag. If the relevant information is not available, leave this field as null.

secondaryAlignment *boolean*	Whether this alignment is secondary. Equivalent to SAM flag 0x100. A secondary alignment represents an alternative to the primary alignment for this read. Aligners may return secondary alignments if a read can map ambiguously to multiple coordinates in the genome. By convention, each read has one and only one alignment where both secondaryAlignment and supplementaryAlignment are false.		
supplementaryAlignment *boolean*	Whether this alignment is supplementary. Equivalent to SAM flag 0x800. Supplementary alignments are used in the representation of a chimeric alignment. In a chimeric alignment, a read is split into multiple linear alignments that map to different reference contigs. The first linear alignment in the read will be designated as the representative alignment; the remaining linear alignments will be designated as supplementary alignments. These alignments may have different mapping quality scores. In each linear alignment in a chimeric alignment, the read will be hard clipped. The alignedSequence and alignedQuality fields in the alignment record will only represent the bases for its respective linear alignment.		
alignedSequence *String*	The bases of the read sequence contained in this alignment record (equivalent to SEQ in SAM). It may be shorter than the full read sequence and quality. This will occur if the alignment is part of a chimeric alignment, or if the read was trimmed. When this occurs, the CIGAR for this read will begin/end with a hard clip operator that will indicate the length of the excised sequence.		
alignedQuality *List <int>*	The quality of the read sequence contained in this alignment record (equivalent to QUAL in SAM). It may be shorter than the full read sequence and quality. This will occur if the alignment is part of a chimeric alignment, or if the read was trimmed. When this occurs, the CIGAR for this read will begin/end with a hard clip operator that will indicate the length of the excised sequence.		
nextMatePosition *Position*	The mapping of the primary alignment of the (readNumber + 1) % numberReads read in the fragment. It replaces mate position and mate strand in SAM. **referenceName** *String* — The name of the Reference on which the Position is located. **position** *long* — The 0-based offset from the start of the forward strand for that Reference. Genomic positions are non-negative integers less than Reference length. **strand** *Strand* — Indicates the DNA strand associate for some data item. **NEG_STRAND** — The negative (-) strand. **POS_STRAND** — The postive (+) strand.		
info *Map <String, List <String> >*	A map of additional read alignment information used to store SAM's optional fields (more information at https://samtools.github.io/hts-specs/SAMtags.pdf).		