The

(SAM) file format

Why SAM/BAM

flexible and able to store all the alignment information generated by various alignment programs

easily generated by alignment programs or converted from existing alignment formats

allows most of the operations on the alignment to work on a stream

allows the file to be indexed by genomic position to efficiently retrieve all reads aligning to a locus

BAM/SAM format specification

BAM format is similar to SAM format and both are described in
http://samtools.sourceforce.net/SAM1.pdf

header section (not mandatory but recommended)

- @HD (first line in case a header is present)
  - VN (Format version, mandatory in header section
- @SQ (Reference sequence dictionary, also alignment sorting order)
  - SN (Reference sequence name e.g. chr1 chrM)
  - LN (Reference sequence Length)
- @RG (read group, to identify individual experiments after concatenation of data sets)
- @PG (Program, to record the programs that are applied on the alignment results)
- @CO (comments)

BAM/SAM format specification

alignment section (mandatory fields)

Field	regexp range	description
QNAME	[!-?A-~]1,255	query name (*)
FLAG	[0,2^{16}-1]	bitwise flag
RNAME	*|[!-()+-<>-~][!-~]*	reference name
POS	[0,2^{29}-1]	1-based leftmost position
MAPQ	[0,2^{8}-1]	mapping quality
CIGAR	*|([0-9]+[MIDNSHPX=])+	mapping description
RNEXT	*|=|[!-()+-<>-~][!-~]*	ref name of next segment
PNEXT	[0,2^{29}-1]	pos of next seqment
TLEN	[-2^{29}+1,2^{29}-1]	template length
SEQ	*|[A-Za-z=.]+	segment sequence
QUAL	[!-~]+	Phred-scaled quality+33

*=NA

Bitwise flag

Hex	Dec	Binary	Description
0x01	== 1	== 00000000001	multiple segments
0x02	== 2	== 00000000010	each segment properly aligned
0x04	== 4	== 00000000100	segment unmapped
0x08	== 8	== 00000001000	next segment unmapped
0x10	== 16	== 00000010000	reverse complemented
0x20	== 32	== 00000100000	next segment being reversed
0x40	== 64	== 00001000000	the first segment
0x80	== 128	== 00010000000	the last segment
0x100	== 256	== 00100000000	secondary alignment
0x200	== 512	== 01000000000	not passing quality controls
0x400	==1024	== 10000000000	PCR or optical duplicate

http://picard.sourceforge.net/explain-flags.html

CIGAR

SAM is able to store most types of alignments and the extended CIGAR string is the key to describing these alignments:

extended Compact Idiosyncratic Gapped Alignment Report (CIGAR)

M match	S soft clipping	H hard clipping
I Insertion	N gap	P padding
D deletion	= Match	X mismatch

clipped_alignment

```
REF: AGCTAGCATCGTGTCGCCCGTCTAGCATACGCATGATCGACTGTCAGCTAGTCAGACTAGTCGATCGATGTG
     READ:  gggGTGTCGCC-GTCTAGgggg
```

The CIGAR for this alignment is: 3S8M1D6M4S.

spliced alignments

```
REF: AGCTAGCATCGTGTCGCCCGTCTAGCATACGCATGATCGACTGTCAGCTAGTCAGACTAGTCGATCGATGTG
READ:          GTGTCGCCC.............................TCAGAATA
```

The CIGAR for this alignment is : 9M32N8M.

Nucleotide variation

Mismatching positions are stored in the MD tag in the optional field of the alignment section.

The MD field aims to achieve SNP/indel calling without looking at the reference.

MD:Z:10A5^AC6

(from the leftmost reference base in the alignment)

10 matches

A on the reference (different from aligned read base)

5 matches

2bp deletion (AC) from the reference

6 matches