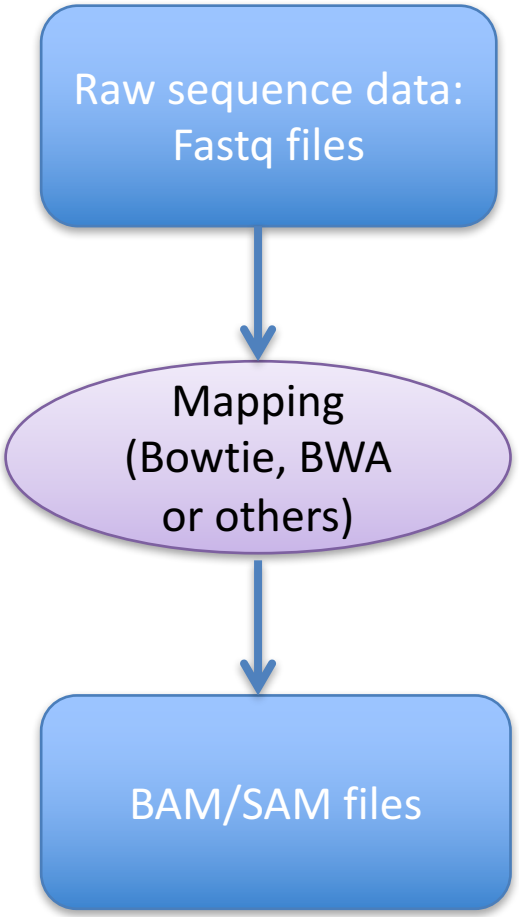
Amel Ghouila, Claudia Chica, Emna Achouri & Fatma Guerfali
C3BI Hands-on NGS course – IPP – 23rd Nov 2016

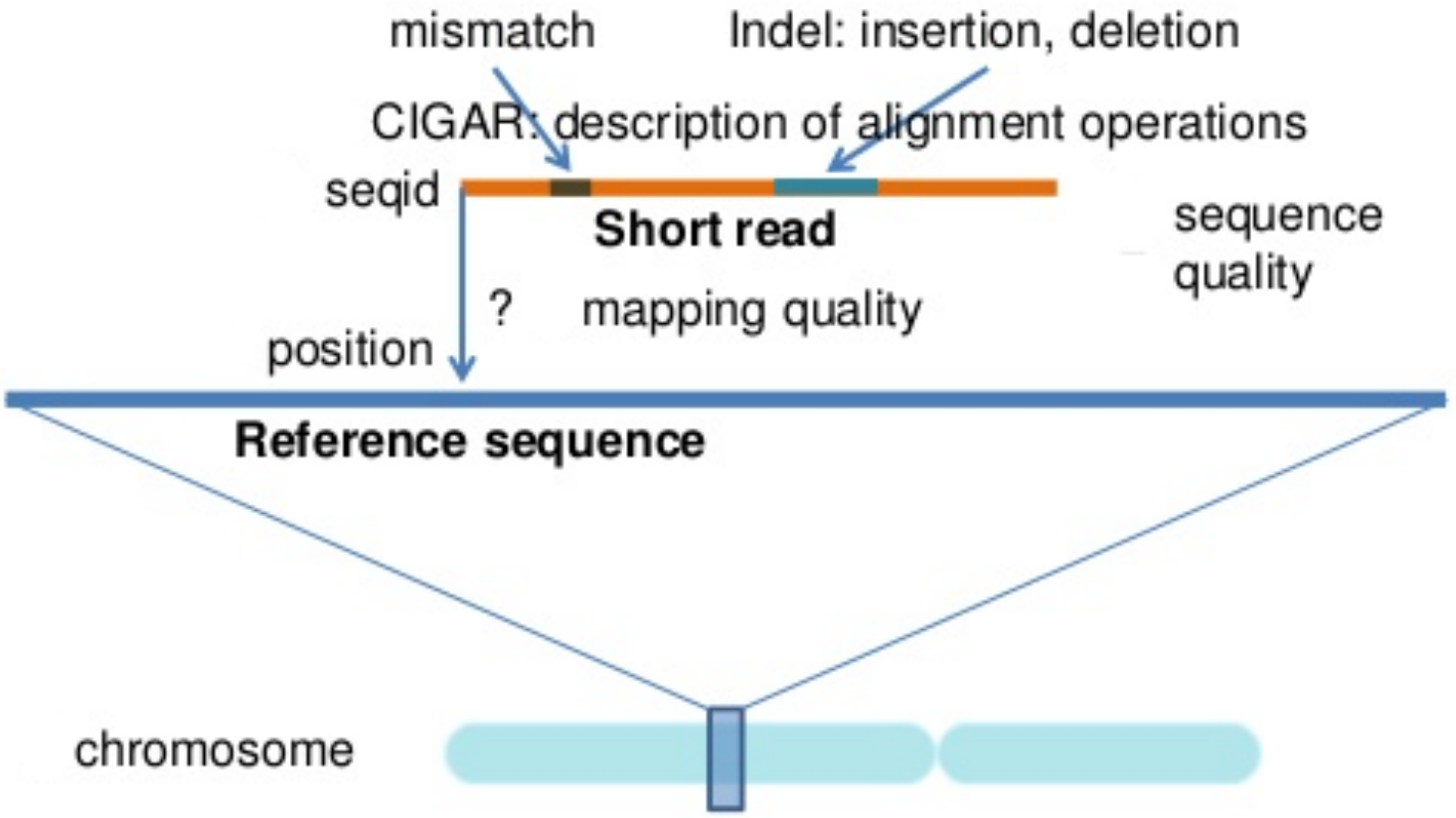
SAM and BAM formats

SAM, BAM formats

- After mapping the FASTQ file to the reference genome you will end up with a SAM or BAM alignment file
- SAM stands for Sequence Alignment/Map format
- A single SAM file can store mapped, unmapped, and even QC-failed reads from a sequencing run, and indexed to allow rapid access. This means that the raw sequencing data can be fully recapitulated from the SAM/BAM file.



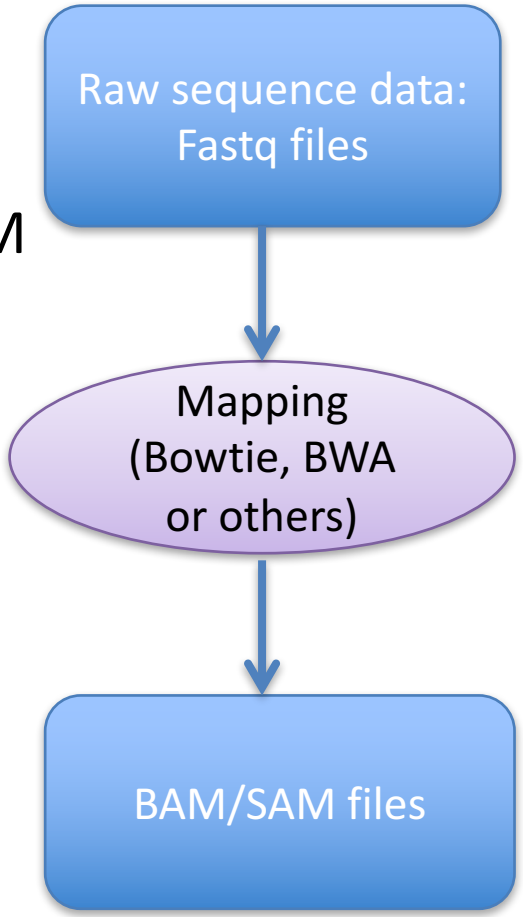
SAM Format



Li Shen, 2014

SAM, BAM formats

- SAM is rarely helpful and really takes up too much space which is why we use only the BAM in principle
- A BAM file (.bam) is the binary version of a SAM file (saving storage and faster manipulation)

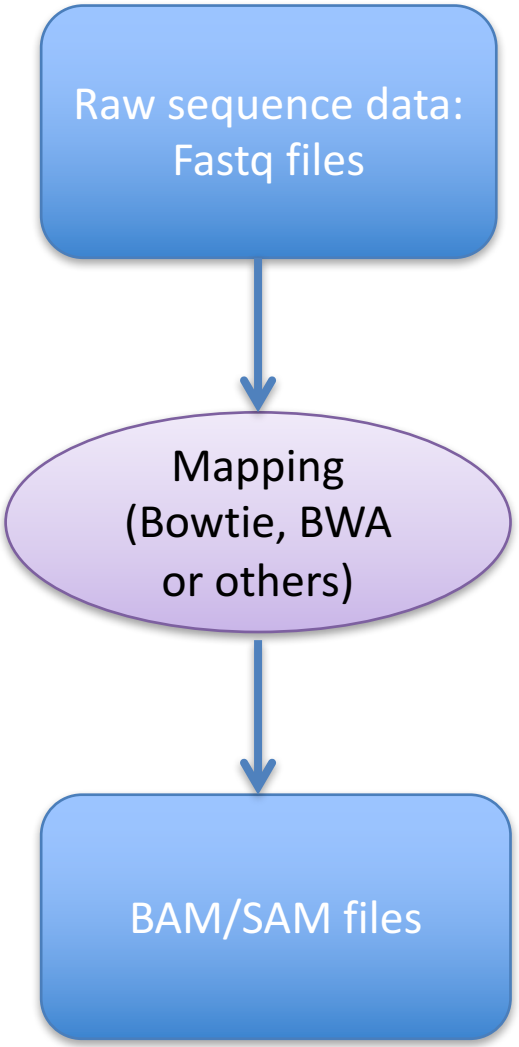


SAM, BAM formats

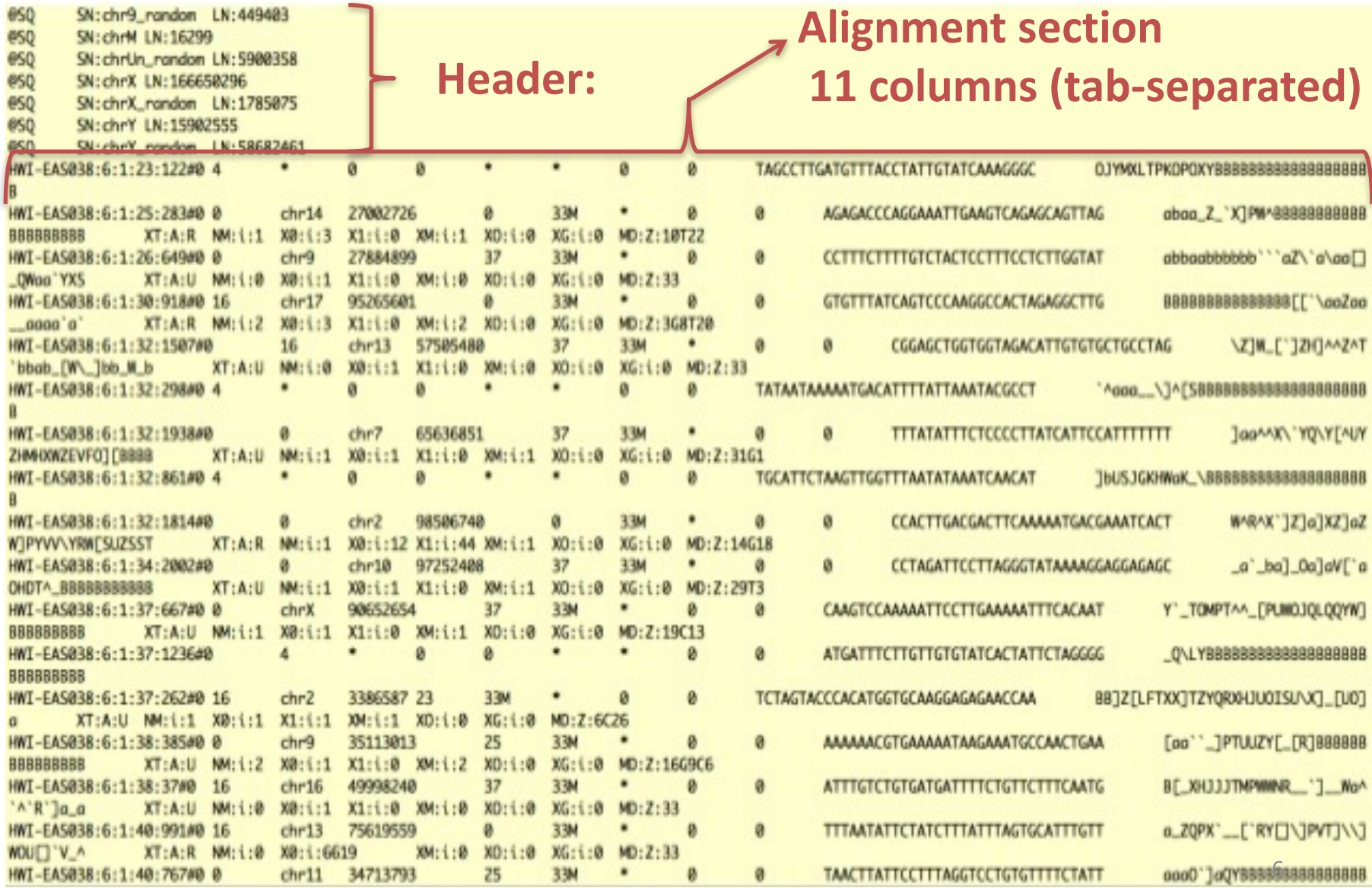
- A SAM file (.sam) is a tab-delimited text file that contains sequence alignment data
- SAM files can be opened using a text editor or viewed using the UNIX "more" command

- Most alignment programs will supply:

- a header: describing the format version, sorting order of the reads, genomic sequences to which the reads were mapped
- an alignment section: contains the information for each sequence about where/how it aligns to the reference genome



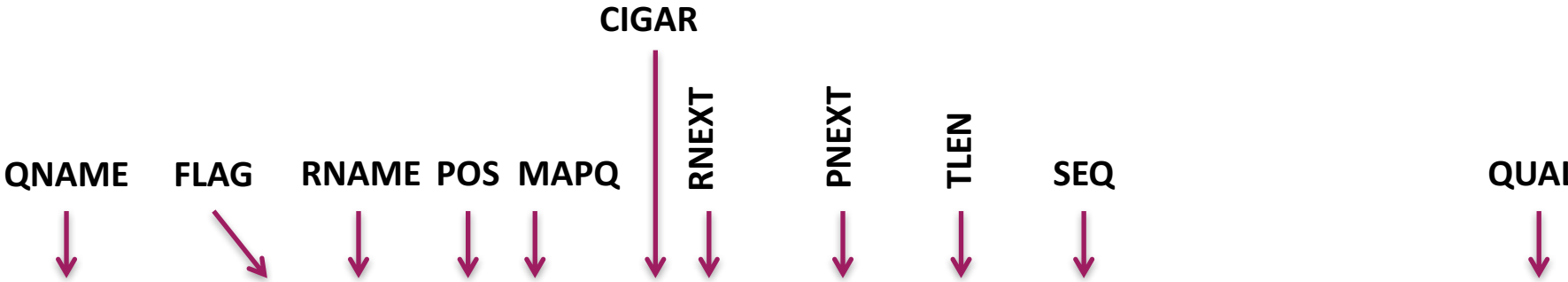
SAM, BAM formats




```
@SQ     SN:chr9_random  LN:449403
@SQ     SN:chrM LN:16299
@SQ     SN:chrUn_random LN:5900358
@SQ     SN:chrX LN:166650296
@SQ     SN:chrX_random  LN:1785075
@SQ     SN:chrY LN:15902555
@SQ     SN:chrY_random  LN:58682461
HWI-EAS038:6:1:23:122#0 4       *       0       0       *       *       0       0       TAGCCTTGATGTTTACCTATTGTATCAAAGGGC        OJYMXLTPKDPOXYBBBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:25:283#0 0       chr14   27002726        0       33M     *       0       0       AGAGACCCAGGAAATTGAAGTCAGAGCAGTTAG        abaa_Z_`X]PW^BBBBBBBBBB
BBBBBBBBB       XT:A:R  NM:i:1  X0:i:3  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:18T22
HWI-EAS038:6:1:26:649#0 0       chr9    27884899        37      33M     *       0       0       CCTTTCTTTTGTCTACTCCTTTCCTCTTGGTAT        abbaabbbbbb```aZ\`a\aa[]
_QWaa`YX5       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:30:918#0 16      chr17   95265601        0       33M     *       0       0       GTGTTTATCAGTCCCAAGGCCACTAGAGGCTTG        BBBBBBBBBBBBBB[[`\aaZaa
__aaaa`a`       XT:A:R  NM:i:2  X0:i:3  X1:i:0  XM:i:2  XO:i:0  XG:i:0  MD:Z:3G8T20
HWI-EAS038:6:1:32:1507#0        16      chr13   57505480        37      33M     *       0       0       CGGAGCTGGTGGTAGACATTGTGTGCTGCCTAG        \Z]W_[`]ZH[]^^Z^T
`bbab_[W\_]bb_W_b       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:32:298#0 4       *       0       0       *       *       0       0       TATAATAAAAATGACATTTTATTAAATACGCCT        `^aaa__\]^[SBBBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:32:1938#0        0       chr7    65636851        37      33M     *       0       0       TTTATATTTCTCCCCTTATCATTCCATTTTTTT        ]aa^^X\`YQ\Y[^U\Y
ZHHXWZEVFQ][BBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:31G1
HWI-EAS038:6:1:32:861#0 4       *       0       0       *       *       0       0       TGCATTCTAAGTTGGTTTAATATAAATCAACAT        ]bUSJGKHWaK_\BBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:32:1814#0        0       chr2    98506740        0       33M     *       0       0       CCACTTGACGACTTCAAAAATGACGAAATCACT        W^R^X`]Z]a]XZ]aZ
W]PYVV\YRW[SUZSST       XT:A:R  NM:i:1  X0:i:12 X1:i:44 XM:i:1  XO:i:0  XG:i:0  MD:Z:14G18
HWI-EAS038:6:1:34:2002#0        0       chr10   97252488        37      33M     *       0       0       CCTAGATTCCTTAGGGTATAAAAGGAGGAGAGC        _a`_ba]_Oa]aV[`a
CHDT^_BBBBBBBBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:29T3
HWI-EAS038:6:1:37:667#0 0       chrX    90652654        37      33M     *       0       0       CAAGTCCAAAAATTCCTTGAAAAATTTCACAAT        Y`_TOMPT^^_[PUWOJQLQQYW]
BBBBBBBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:19C13
HWI-EAS038:6:1:37:1236#0        4       *       0       0       *       *       0       0       ATGATTTCTTGTTGTGTATCACTATTCTAGGGG        _Q\LYBBBBBBBBBBBBBBBBBB
BBBBBBBBB
HWI-EAS038:6:1:37:262#0 16      chr2    3386587 23      33M     *       0       0       TCTAGTACCCACATGGTGCAAGGAGAGAACCAA        BB]Z[LFTXX]TZYQRXHJUOISU\X]_[UO]
a       XT:A:U  NM:i:1  X0:i:1  X1:i:1  XM:i:1  XO:i:0  XG:i:0  MD:Z:6C26
HWI-EAS038:6:1:38:385#0 0       chr9    35113013        25      33M     *       0       0       AAAAAACGTGAAAAATAAGAAATGCCAACTGAA        [aa``_]PTUUZY[_[R]BBBBBB
BBBBBBBBB       XT:A:U  NM:i:2  X0:i:1  X1:i:0  XM:i:2  XO:i:0  XG:i:0  MD:Z:16G9C6
HWI-EAS038:6:1:38:37#0  16      chr16   49998240        37      33M     *       0       0       ATTTGTCTGTGATGATTTTCTGTTCTTTCAATG        B[_XHJJJTMPWWNR__`]__Wa^
`^`R`]a_a       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:40:991#0 16      chr13   75619559        0       33M     *       0       0       TTTAATATTCTATCTTTATTTAGTGCATTTGTT        a_ZQPX`__[`RY[]\]PVT]\\]
WOU[]`V_^       XT:A:R  NM:i:0  X0:i:6619       XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:40:767#0 0       chr11   34713793        25      33M     *       0       0       TAACTTATTCCTTTAGGTCCTGTGTTTTCTATT        aaa0`]aQYBBBBBBBBBBBBBBBB
```


Amel Ghouila, Claudia Chica, Emna Achouri & Fatma Guerfali
C3BI Hands-on NGS course – IPP – 23rd Nov 2016

SAM Format




```
1:497:R:-272+13M17D24M          113     1       497     37      37M     15      100338662       0       CGGGTCTGACCTGAGGAGAACTGTGCTCCGCCTTCAG   0;==-==9
19:20389:F:275+18M2D19M         99      1       17644   0       37M     =       17919   314     TATGACTGCTAATAATACCTACACATGTTAGAACCAT    >>>>>>>>>>>>>>>>
19:20389:F:275+18M2D19M         147     1       17919   0       18M2D19M        =       17644   -314    GTAGTACCAACTGTAAGTCCTTATCTTCATACTTTGT    ;44999;4
9:21597+10M2I25M:R:-209         83      1       21678   0       8M2I27M =       21469   -244    CACCACATCACATATACCAAGCCTGGCTGTGTCTTCT    <;9<<5><<<<><<<<
```

http://samtools.sourceforge.net/SAM1.pdf
http://genome.sph.umich.edu/wiki/SAM

SAM fomat

The following table gives an overview of the mandatory fields in the SAM format:

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{31}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

(http://samtools.github.io/hts-specs/SAMv1.pdf)

QNAME: Query template NAME. Reads/segments having identical QNAME are regarded to come from the same template. A QNAME '*' indicates the information is unavailable.

Amel Ghouila, Claudia Chica, Emna Achouri & Fatma Guerfali
C3BI Hands-on NGS course – IPP – 23rd Nov 2016

SAM fomat (2)

The following table gives an overview of the mandatory fields in the SAM format:

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{31}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

(http://samtools.github.io/hts-specs/SAMv1.pdf)

FLAG: FLAG: bitwise FLAG (ideal for compression).

11 boolean flags all stotred in a singe column

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Amel Ghouila, Claudia Chica, Emna Achouri & Fatma Guerfali
C3BI Hands-on NGS course – IPP – 23rd Nov 2016

SAM flag: example

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

SAM file

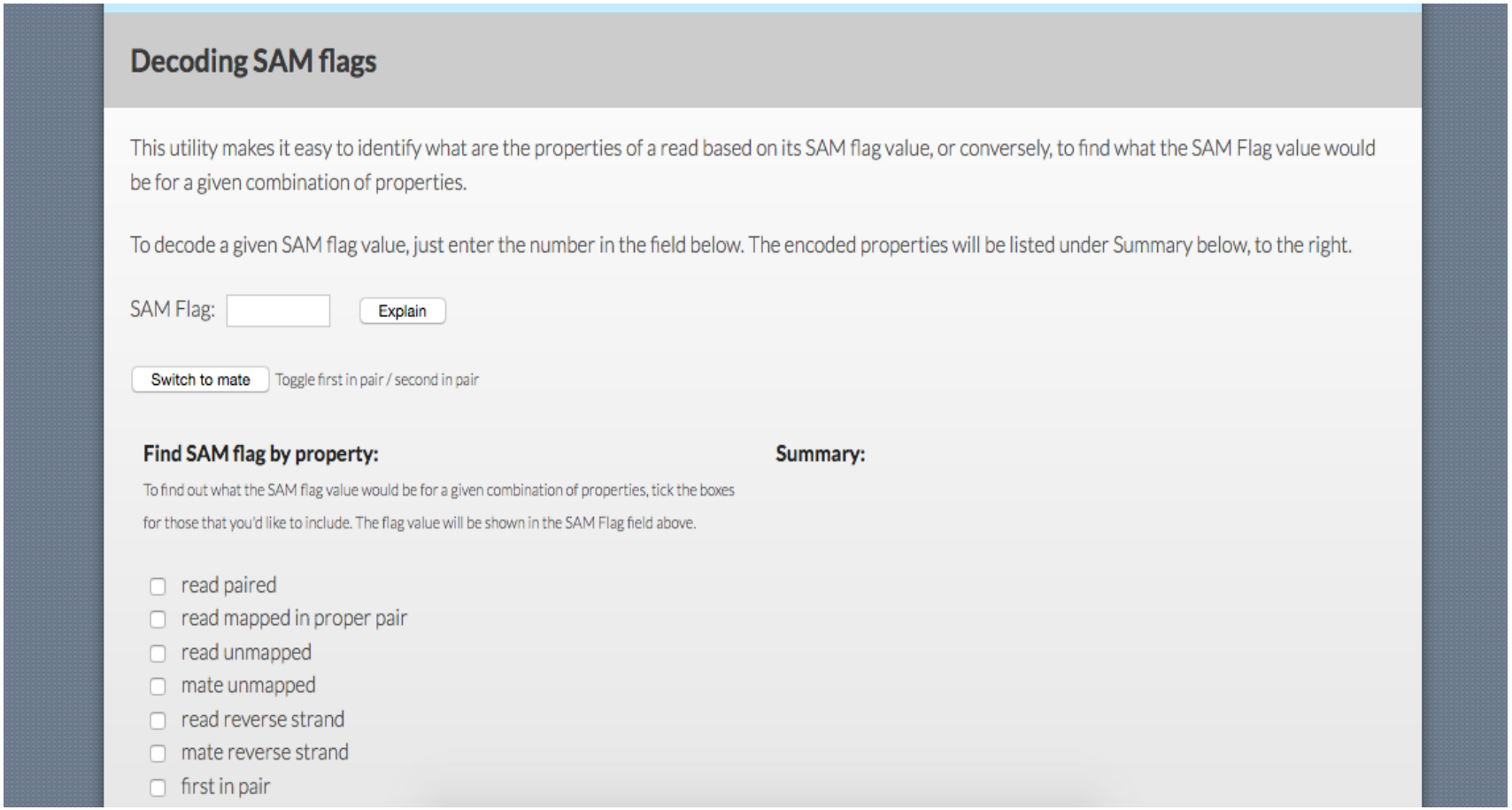
(b)
```
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
```

read mapped to position 7:
FLAG 163 (=1 + 2 + 32 + 128):

- **Read is the second read in the pair (128)**
- **Read is properly paired (1 + 2)**
- **its mate is mapped to 37 on the reverse strand (32)**

Decoding SAM flags

Explain flag tool:
https://broadinstitute.github.io/picard/explain-flags.html



SAM fomat (3)

The following table gives an overview of the mandatory fields in the SAM format:

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{31}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

(http://samtools.github.io/hts-specs/SAMv1.pdf)

It equals −10 log10 Pr{mapping position is wrong}, rounded to the nearest integer.

The MAPQ value can be used to figure out how unique an alignment is in the genome.

- ✓ Large number, >10 indicates it's likely the alignment is unique.
- ✓ 255 indicates that the mapping quality is not available

SAM fomat: CIGAR string

- The CIGAR string is a **sequence of numbers and letters** representing the associated **information on bases alignment** used to indicate things like which bases align (either a match/mismatch) with the reference, are deleted from the reference, and if there are insertions that are not in the reference

More information about these formats available here:
http://samtools.sourceforge.net
https://samtools.github.io/hts-specs/SAMv1.pdf

SAM fomat: CIGAR string

Mapped and unmapped reads are imported into SAM/BAM format

The standard CIGAR description of pairwise alignment defines three operations:
'**M**' for **alignment match**, '**I**' for insertion compared with the reference and '**D**' for deletion.

(NB: The POS indicates that the read aligns starting at position 5 on the reference)

The CIGAR :
3M = 3 bases in the read sequence align with the reference.
1I = The next base in the read does not exist in the reference.
1D = The reference base does not exist in the read sequence

```
RefPos:      1  2  3  4  5  6  7  8  9 10 11 12 13
Reference:   C  C  A  T  A  C  T  G  A  A  C  T  G
Read: ACTAGAATG
```

```
RefPos:      1  2  3  4  5  6  7     8  9 10 11 12 13
Reference:   C  C  A  T  A  C  T     G  A  A  C  T  G
Read:                   A  C  T  A  G  A  A     T  G
```

POS: 5
CIGAR: 3M1I3M1D2M

http://genome.sph.umich.edu/wiki/SAM

SAM fomat: CIGAR string

Examples of CIGAR strings for different types of alignments

Alignments

(a)

```
coor    12345678901234  5678901234567890123456789012345
ref     AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

r001+         TTAGATAAAGGATA*CTG
r002+       aaaAGATAA*GGATA
r003+     gcctaAGCTAA
r004+                     ATAGCT..............TCAGC
r003-                           ttagctTAGGC
r001-                                        CAGCGCCAT
```

SAM file

(b)

```
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA   *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC    *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

(Li et al., 2009)

Amel Ghouila, Claudia Chica, Emna Achouri & Fatma Guerfali
C3BI Hands-on NGS course – IPP – 23rd Nov 2016

SAM fomat (5)

The following table gives an overview of the mandatory fields in the SAM format:

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	*|[!-()+-<>-~][!-~]*	Reference sequence NAME
4	POS	Int	[0,2^{31}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	*|([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	*|=|[!-()+-<>-~][!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

(http://samtools.github.io/hts-specs/SAMv1.pdf)

Name of mate (mate pair information for paired-end sequencing)
Position of mate (mate pair information)

Obviously, the chromsome and position are important. The CIGAR string is also important to know where insertions (i.e. introns) might exist in your read.