Filetypes for Annotation & Alignment

J Fass | 20 June 2017

Filetypes

- Fasta
- Fastq
- GTF / GFF
- SAM / BAM / CRAM
- BED

Fasta

```
>sequenceName | plus other junk | few maintain a standard here
AGTGGAGCAAGCACAGAGAAGAAACTGCAGTCAGGACATAAAGTAAAGTA
ATTAATCTAAAAAATAGTCTGAGCAGTCTTCTCTGCTGAANNNNNNNNNN
NNNNNNNNNNNNNNNNNNNAATTTCCTTACTAGGAGGTCTTTAGTACAGA
TTCCTGATATGTAATTAATCACTAAATGTCTTTAATGGGATCTCTTTCTA
TTGAGATATTTGTAAACTTTCTTCATGTGATTGGTTTACAGATATTCAGG
TTTCTGCAAATGGGTGCTGTCTATATTATAGAATTTTTAGTTGAAATTTT
CAAAATACTCTTTGagtattctcttgtaattatattactttacaaggttt
gtggggcatctctttcatttgtgattacatggttgcagtattctttttgt
tcttagtcagactgtataattgtctgtgaagtccagtaaacttttgaaag
```

Assembly gap?

Soft-masked repetitive sequence? Low confidence assembly?

Fastq

```
@K00188:264:HG3WJBBXX:1:1101:6289:1595 1:N:0:TAGCTT
GGACTGCCTTTCAGCCCGTCGCAGAGGGAATGGGAGCCTCTGGAGCGGGTGCAGAGGCTCAGCAG
+
AAFFFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJAJJJJJJ

@header
<sequence>
+(sometimes header?)
<base qualities>
```

Fastq

```
@K00188:264:HG3WJBBXX:1:1101:6289:1595 1:N:0:TAGCTT
GGACTGCCTTTCAGCCCGTCGCAGAGGAATGGGAGCCTCTGGAGCGGGTGCAGAGGCTCAGCAG
+
AAFFFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJAJJJJJJ
```

Oct	Dec	Hex	Char	Oct	Dec	Hex	Char
000	0	00	NUL '\0' (null character)	100	64	40	@
001	1	01	SOH (start of heading)	101	65	41	A
002	2	02	STX (start of text)	102	66	42	B
003	3	03	ETX (end of text)	103	67	43	C
004	4	04	EOT (end of transmission)	104	68	44	D
005	5	05	ENQ (enquiry)	105	69	45	E
006	6	06	ACK (acknowledge)	106	70	46	F
007	7	07	BEL '\a' (bell)	107	71	47	G
010	8	08	BS '\b' (backspace)	110	72	48	H
011	9	09	HT '\t' (horizontal tab)	111	73	49	I
012	10	0A	LF '\n' (new line)	112	74	4A	J
013	11	0B	VT '\v' (vertical tab)	113	75	4B	K

Fastq

Oct	Dec	Hex	Char
112	74	4A	J

```
74 - 33 = 41
```

Probability of error = 10 ^ (-41 / 10) ~ 0.0001

41 is the “phred-scaled Q-value”

Standard FASTQ encodes qualities using “phred + 33” quality characters. See https://en.wikipedia.org/wiki/FASTQ_format for a good graphic about current and older encodings.

Common QC question: “how many reads have average of at least Q30?”

SAM / BAM / CRAM!

http://www.htslib.org/

See also samtools man page: *http://samtools.sourceforge.net/*

SAM spec grew out of 1000 Genomes Project (see Li et al. 2009 *Bioinformatics* 25:2078)

SAM is plain text; BAM is binary, compressed version of SAM; CRAM is further compressed but not widely used / recognizable by many tools.

SAM

```
[...]
@SQ     SN:ctg103993     LN:217
@SQ     SN:ctg103994     LN:222
@SQ     SN:ctg103995     LN:205
@SQ     SN:ctg103996     LN:210
@PG     ID:bwa  PN:bwa  VN:0.7.13-r1126 CL:bwa mem -t 4 -M ../../01_Reference/Transcriptome-Contigs-Build2.fna
../../02-Cleaned/3E/3E_SE.fastq
@PG     ID:bwa-7BC92A6F PN:bwa  VN:0.7.13-r1126 CL:bwa mem -t 4 -M ../../01_Reference/Transcriptome-Contigs-Build2.fna
../../02-Cleaned/3E/3E_R1.fastq ../../02-Cleaned/3E/3E_R2.fastq
K00188:264:HG3WJBBXX:1:1116:14692:35180#0       121     ctg2      128     58      101M =       128     0
AAGTCTCGACCAAGTGGTTCAGATGGTGACACAGATGTTAGCCCCATCCACCATTCAGTTGCCGTTTTGATAGCTGGAAATCCTGTAAACACAAT
GCTGAG   FJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   NM:i:10
K00188:264:HG3WJBBXX:1:1116:14692:35180#0       181     ctg2      128     0       *              =       128     0
TTTAGTTTTAATTTTTGACTTTGAATAGCGGGAGTCCAGATCGTGTGAACACAGCAGACTGAGCACTCCATTGACAGCCTTCTTCTGTACTTTAGC
TATCC   FJFJJFAAJF7F7JJJJAFFFAF<7<AFFJJJFJJJJJJJJJJJJJJJJJJJJFJJJJJJFAJJJJJJJFFFJJJJJJJJJJFFJJJJJJFFFAA   AS:i:0  XS:i:0
K00188:264:HG3WJBBXX:1:1202:11028:9596#0       121     ctg5      45      60      101M   =       45      0
TTCTTTTTTCTACAGTTCATTGTCTGTATAAAGTATGCATCAGGAACAATCTGACTAGGAAGGTAAATAATGTAAAACAGATGATTATTGTATGAAA
GTTG  JJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   NM:i:8
K00188:264:HG3WJBBXX:1:1202:11028:9596#0       181     ctg5      45      0       *              =       45      0
TCAGCTGTATTAGTAATTTAGTAGAAAAGGTCTTGAGAGAATTATGTTTTTTAAAAATCCACATCACTTCAAACAAAAAGCCCCATTAGAATGGAGG
GCCA   FJFJJJJJJFJJJJJJFFJJJJFJAJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFF-JFJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   AS:i:0
[...]
```

Header lines (start with “@”)

Alignment line (one line per alignment)

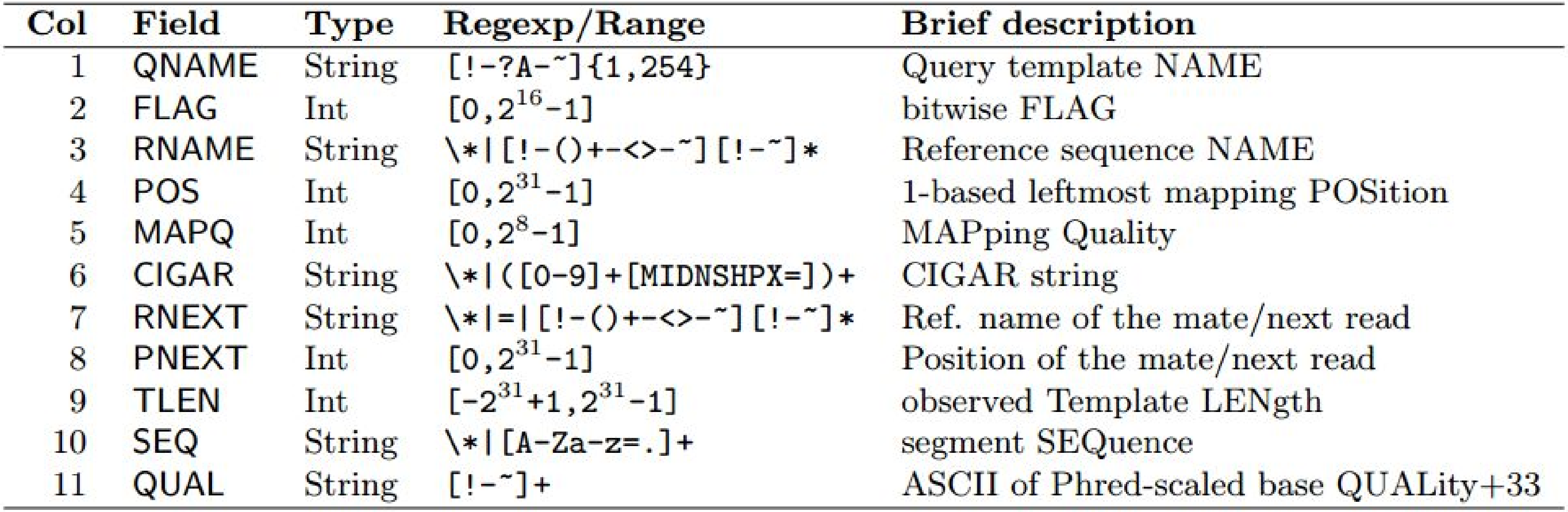
SAM

```
[...]
@SQ     SN:ctg103993     LN:217
@SQ     SN:ctg103994     LN:222
@SQ     SN:ctg103995     LN:205
@SQ     SN:ctg103996     LN:210
@PG     ID:bwa  PN:bwa  VN:0.7.13-r1126 CL:bwa mem -t 4 -M ../../01_Reference/Transcriptome-Contigs-Build2.fna
../../02-Cleaned/3E/3E_SE.fastq
@PG     ID:bwa-7BC92A6F PN:bwa  VN:0.7.13-r1126 CL:bwa mem -t 4 -M ../../01_Reference/Transcriptome-Contigs-Build2.fna
../../02-Cleaned/3E/3E_R1.fastq ../../02-Cleaned/3E/3E_R2.fastq
K00188:264:HG3WJBBXX:1:1116:14692:35180#0      121     ctg2      128     58      101M =       128     0
AAGTCTCGACCAAGTGGTTCAGATGGTGACACAGATGTTAGCCCCATCCACCATTCAGTTGCCGTTTTGATAGCTGGAAATCCTGTAAACACAAT
GCTGAG   FJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   NM:i:10
K00188:264:HG3WJBBXX:1:1116:14692:35180#0      181     ctg2      128     0       *            =       128     0
TTTAGTTTTAATTTTTGACTTTGAATAGCGGGAGTCCAGATCGTGTGAACACAGCAGACTGAGCACTCCATTGACAGCCTTCTTCTGTACTTTAGC
TATCC   FJFJJFAAJF7F7JJJJAFFFAF<7<AFFJJJFJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJFAJJJJJJJJFFFJJJJJJJJJJJFFJJJJJJFFFAA   AS:i:0  XS:i:0
K00188:264:HG3WJBBXX:1:1202:11028:9596#0       121     ctg5      45      60      101M   =       45      0
TTCTTTTTTCTACAGTTCATTGTCTGTATAAAGTATGCATCAGGAACAATCTGACTAGGAAGGTAAATAATGTAAAACAGATGATTATTGTATGAAA
GTTG  JJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   NM:i:8
K00188:264:HG3WJBBXX:1:1202:11028:9596#0       181     ctg5      45      0       *            =       45      0
TCAGCTGTATTAGTAATTTAGTAGAAAAGGTCTTGAGAGAATTATGTTTTTTAAAAATCCACATCACTTCAAACAAAAAGCCCCATTAGAATGGAGG
GCCA   FJFJJJJJFJJJJJFFJJJJFJAJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFF-JFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   AS:i:0
[...]
```

Read Pair (identical headers)

SAM

```
K00188:264:HG3WJBBXX:1:1116:14692:35180#0       121     ctg2      128     58      101M      =       128     0
AAGTCTCGACCAAGTGGTTCAGATGGTGACACAGATGTTAGCCCCATCCACCATTCAGTTGCCGTTTTGATAGCTGGAAATCCTGTAAACACAATGCTGAG
FJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFFFAA   NM:i:10
```



<table>
<tr><th>Col</th><th>Field</th><th>Type</th><th>Regexp/Range</th><th>Brief description</th></tr>
<tr><td>1</td><td>QNAME</td><td>String</td><td>[!-?A-~]{1,254}</td><td>Query template NAME</td></tr>
<tr><td>2</td><td>FLAG</td><td>Int</td><td>[0,2^{16}-1]</td><td>bitwise FLAG</td></tr>
<tr><td>3</td><td>RNAME</td><td>String</td><td>*|[!-()+-&lt;&gt;-~][!-~]*</td><td>Reference sequence NAME</td></tr>
<tr><td>4</td><td>POS</td><td>Int</td><td>[0,2^{31}-1]</td><td>1-based leftmost mapping POSition</td></tr>
<tr><td>5</td><td>MAPQ</td><td>Int</td><td>[0,2^{8}-1]</td><td>MAPping Quality</td></tr>
<tr><td>6</td><td>CIGAR</td><td>String</td><td>*|([0-9]+[MIDNSHPX=])+</td><td>CIGAR string</td></tr>
<tr><td>7</td><td>RNEXT</td><td>String</td><td>*|=|[!-()+-&lt;&gt;-~][!-~]*</td><td>Ref. name of the mate/next read</td></tr>
<tr><td>8</td><td>PNEXT</td><td>Int</td><td>[0,2^{31}-1]</td><td>Position of the mate/next read</td></tr>
<tr><td>9</td><td>TLEN</td><td>Int</td><td>[-2^{31}+1,2^{31}-1]</td><td>observed Template LENgth</td></tr>
<tr><td>10</td><td>SEQ</td><td>String</td><td>*|[A-Za-z=.]+</td><td>segment SEQuence</td></tr>
<tr><td>11</td><td>QUAL</td><td>String</td><td>[!-~]+</td><td>ASCII of Phred-scaled base QUALity+33</td></tr>
</table>

SAM

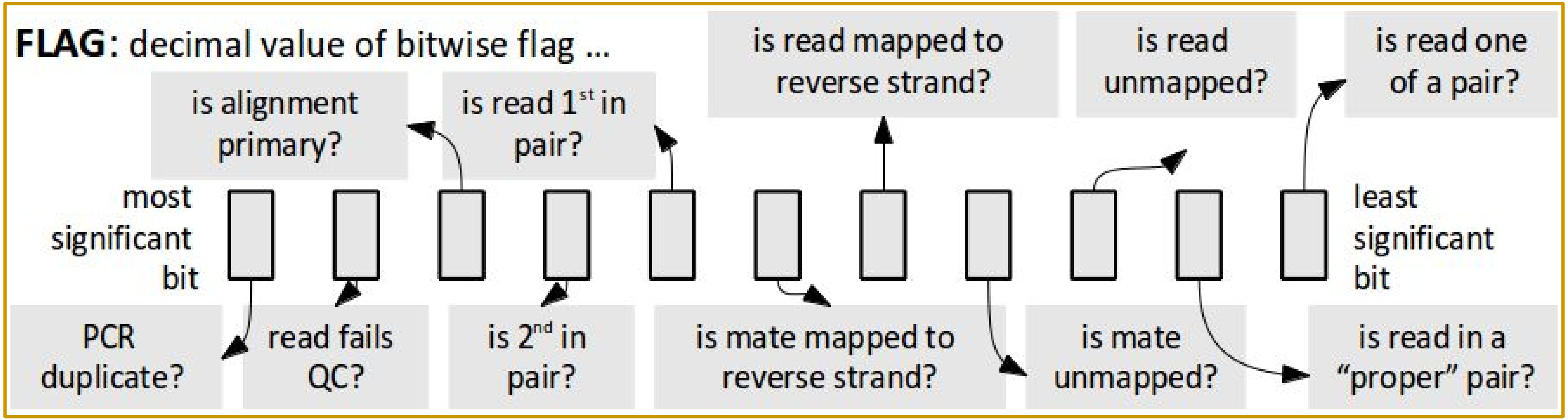
QNAME: Query name

Read IDs are truncated at first whitespace (spaces / tabs), which can make them *non-unique*. Illumina reads with older IDs have trailing “/1” and “/2” stripped (this information is recorded in the next field). Illumina reads with newer IDs have second block stripped (read number is recorded in the next field).

@FCC6889ACXX:5:1101:8446:45501#CGATGTATC/1 ⇒ @FCC6889ACXX:5:1101:8446:45501

@HISEQ:153:H8ED7ADXX:1:1101:1368:2069 1:N:0:ATCACG ⇒ @HISEQ:153:H8ED7ADXX:1:1101:1368:2069

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```



FLAG: decimal value of bitwise flag ...

most significant bit

PCR duplicate?

read fails QC?

is alignment primary?

is 2nd in pair?

is read 1st in pair?

is mate mapped to reverse strand?

is read mapped to reverse strand?

is mate unmapped?

is read unmapped?

is read in a “proper” pair?

is read one of a pair?

least significant bit

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

99 (decimal) = 00001100011 (binary) (0 / NO .. 1 / YES)

… so, (from right to left): read is in a pair; the pair is proper; read *is* mapped (double neg); mate *is* mapped (double neg); read is mapped to forward strand (double neg); mate is mapped to reverse strand; read is 1st in pair … *remaining bits not used*

SAM

FLAG: still confused?

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

Common flags for SR (single reads): 0, 4, 16, sometimes 20 (hmm..)

Common flags for PE (paired ends): 99/147, 83/163, 77/141, 65/129, 81/161 ...

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

RNAME: reference sequence name

Reference sequence ID (from fasta header), *possibly truncated at first whitespace (still unique??)*

>chromosome 1
… *becomes* …
chromosome
… (!)

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

POS: 1-based *leftmost* position of (post-clipping) aligned read

```
      ... 4,773,680 4,773,690 4,773,700 4,773,710 ...
                  |         |         |         |
REF:...TACCCAATGGGGATGACATAAGGTGCCATCTGTGGGCTGGTGATTCCATAGTAGAC...
READ:                       GGTGCCATCTGTGGGCTGGTGATCCCATAGTAGAC...
```

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

POS: 1-based *leftmost* position of (post-clipping) aligned read

```
      ... 4,773,680 4,773,690 4,773,700 4,773,710 ...
                  |         |         |         |
REF:...TACCCAATGGGGATGACATAAGGTGCCATCTGTGGGCTGGTGATTCCATAGTAGAC...
READ:                       GGTGCCATCTGTGGGCTGGTGATCCCATAGTAGAC...
```

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
49H101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

POS: 1-based *leftmost* position of (post-clipping) aligned read

```
      ... 4,773,680 4,773,690 4,773,700 4,773,710 ...
                  |         |         |         |
REF:...TACCCAATGGGGATGACATAAGGTGCCATCTGTGGGCTGGTGATTCCATAGTAGAC...
READ:                    GCCGGTGCCATCTGTGGGCTGGTGATCCCATAGTAGAC...
```

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
3S101M
=
4773721
132
GCCGTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBBBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:GCC101(?)  YT:Z:UU XS:A:-  NH:i:1
```

SAM

MAPQ: mapping quality (phred scaled)

Mapping quality is used by some aligners, in different ways. It’s generally a function of the edit distance (mismatches, indels), and the uniqueness of the alignment. Multiple equivalent best alignments yield a mapping quality of zero; alignments with an edit distance close to the best alignment lower the mapping quality.

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

CIGAR: extended CIGAR string (Compact Idiosyncratic Gapped Alignment Report)

Format: [0-9][MIDNSHP][0-9][MIDNSHP]...
M = match / mismatch (!), I/D = insertion / deletion, N = skipped bases on reference, S/H = soft / hard clip (hard clipped bases no longer appear in the sequence field), P = padding
… e.g. “101M” means that all bases in the read align to bases in the reference, starting with position (4,773,690), always in the order of the reference.

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

MRNM: reference sequence to which the *mate* of this read is aligned

“=” … mate is aligned to the same reference sequence is this read

“*” … this is a single read; no mate exists

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

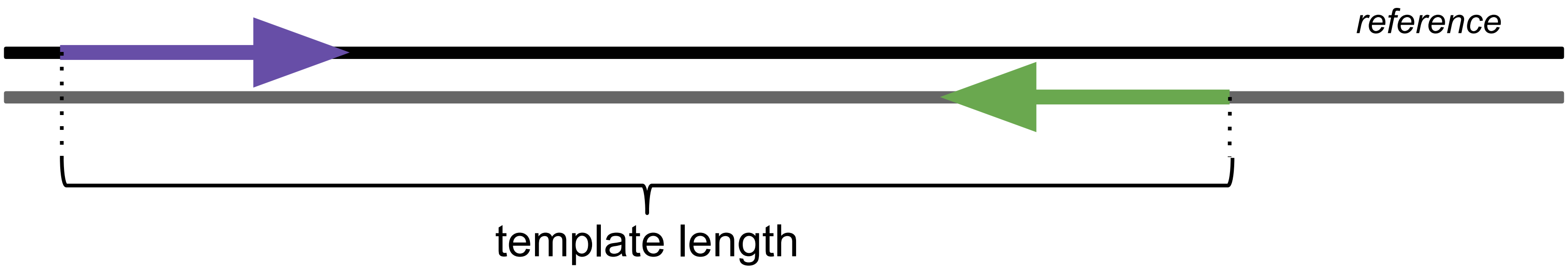
MPOS: 1-based, left-most position of 1st (post-clipping) nucleotide of mate read

“0” … no mate exists

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

TLEN: inferred insert size / template length … “0” if no mate … “-#” if second read(?)



```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM

SEQ and **QUAL:** read’s nucleotides and base qualities, *always in the order of the reference (forward, top) strand!* … includes any insertions, deletions, etc. present in the read.

Reads aligned to reverse strand appear in reverse, with reversed base qualities.

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

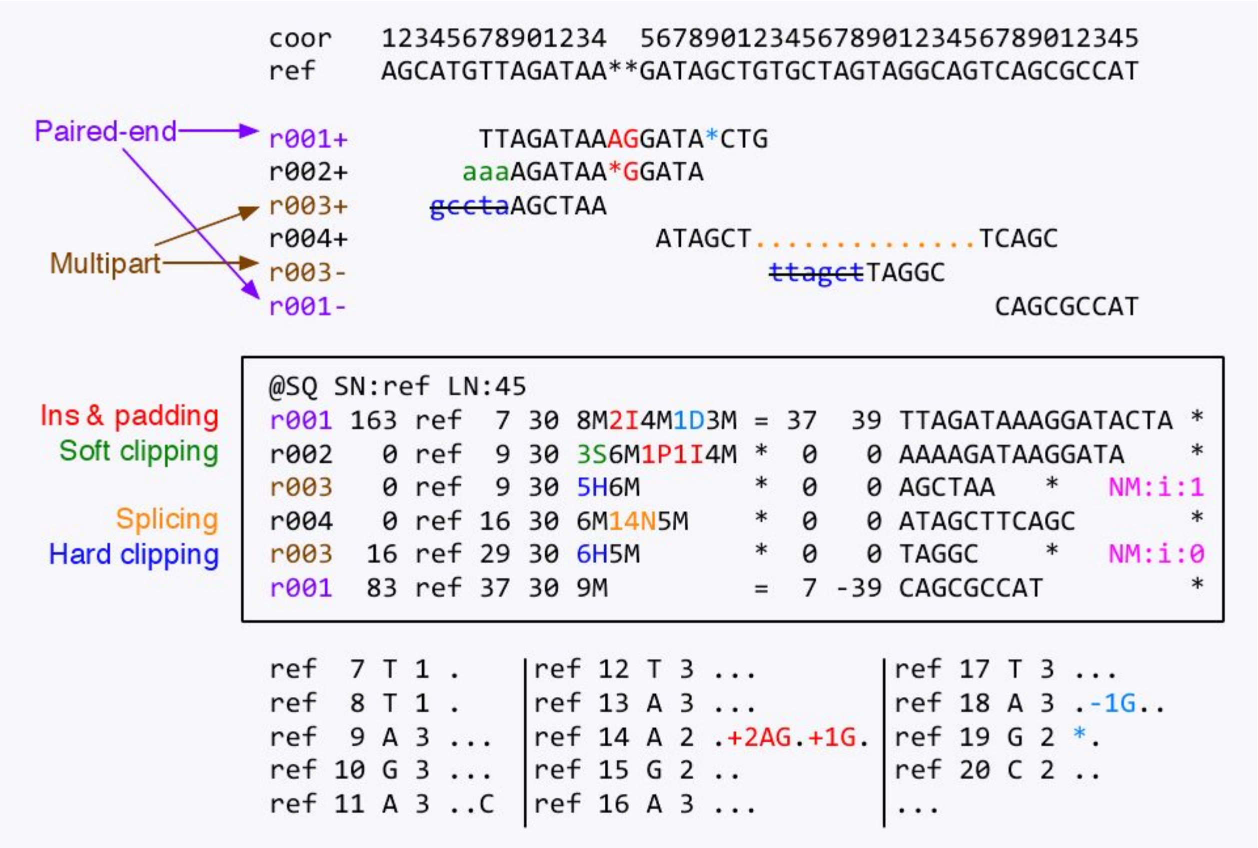
SAM

OPT: various pre-defined and user-defined tags in the format TAG:VTYPE:VALUE … VTYPE is one of [A (printable character); i (signed integer); f (floating point); z (printable string); H (hex string)].

e.g.: NM:i:0 means zero mismatches in this alignment
e.g.: XS:A:- was set by TopHat, RNA that was read was coded by the reverse strand
e.g.: NH:i:1 means that the number of hits for this read was 1 (would be more for repeat)

```
HISEQ:153:H8ED7ADXX:1:1104:8193:69947
99
chr1
4773690
50
101M
=
4773721
132
GTGCCATCTGTGGGCTGGTGATC[...]AGCAGCATGCTCCATGGTCTCTACATG
BBBFFFFFFBFFFFBFFFBFFBB[...]FFBFFBBBBBBBBBBBBBBBBBBBBB<
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:101  YT:Z:UU XS:A:-  NH:i:1
```

SAM - quick summary



```
coor   12345678901234  5678901234567890123456789012345
ref    AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

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

```
@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         *
```

```
ref  7 T 1 .      |ref 12 T 3 ...        |ref 17 T 3 ...
ref  8 T 1 .      |ref 13 A 3 ...        |ref 18 A 3 .-1G..
ref  9 A 3 ...    |ref 14 A 2 .+2AG.+1G. |ref 19 G 2 *.
ref 10 G 3 ...    |ref 15 G 2 ..         |ref 20 C 2 ..
ref 11 A 3 ..C    |ref 16 A 3 ...        |...
```

google "Heng Li slides" - Challenges and Solutions in the Analysis of Next Generation Sequencing Data (2010)

BAM

BAMs are compressed SAMs (so, binary, not human-readable text … don't look directly at them!). They can be indexed to allow rapid extraction of information, so alignment viewers do not need to uncompress the whole BAM file in order to look at information for a particular read or coordinate range, somewhere in the file.

Indexing your BAM file, myCoolBamFile.bam, will create an index file, myCoolBamFile.bam.bai, which is needed (in addition to the BAM file) by viewers and other downstream tools. An occasional downstream tool will require an index called myCoolBamFile.bai (notice that the ".bai" replaces the ".bam", instead of being appended after it).

CRAM

Available as of SAMtools 1.0, and is a binary format like BAM. Uses data-specific compression tools (i.e. compressing letters is different than compressing numbers), *specifically* reference-based compression (e.g. for aligned reads, only *mis-matching* bases need to be stored). Also can employ *lossy* compression of base qualities, which appears to have a negligible effect on, say, variant calling (see Illumina *white paper*).

Indexing your CRAM file, myCoolBamFile.cram, will create an index file, myCoolBamFile.cram.crai, which is needed (in addition to the CRAM file) by viewers and other downstream tools.

This is still a ***relatively recent development***, so it may be a while before many tools are CRAM-capable.