Example of SAM/BAM file format

Example SAM/BAM/CRAM header section (abbreviated)

Example SAM/BAM/CRAM alignment section (only 10 alignments shown)

Introduction to the SAM/BAM format

- The specification
  - http://samtools.sourceforge.net/SAM1.pdf
- SAM is uncompressed text data
- BAM is a compressed version of SAM
  - lossless BGZF format
- BAM files are usually 'indexed'
  - A '.bai' file will be found beside the '.bam' file
  - Indexing provides fast retrieval of alignments overlapping a specified region without going through all alignments.
  - BAM must be sorted by the reference ID and then the leftmost coordinate before indexing

SAM/BAM header section

- Used to describe source of data, reference sequence, method of alignment, etc.

- Each section begins with character '@' followed by a two-letter record type code. These are followed by two-letter tags and values:

  - @HD The header line
    - VN: format version
    - SO: Sorting order of alignments

  - @SQ Reference sequence dictionary
    - SN: reference sequence name
    - LN: reference sequence length
    - SP: species

  - @RG Read group
    - ID: read group identifier
    - CN: name of sequencing center
    - SM: sample name

  - @PG Program
    - PN: program name
    - VN: program version

SAM/BAM alignment section

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	$[0,2^{29}-1]$	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 segment
8	PNEXT	Int	$[0,2^{29}-1]$	Position of the mate/next segment
9	TLEN	Int	$[-2^{29}+1,2^{29}-1]$	observed Template LENgth
10	SEQ	String	*|[A-Za-z=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

Example values

```
1    QNAME    e.g.   HWI-ST495_129147882:1:2302:10269:12362
2    FLAG     e.g.   99
3    RNAME    e.g.   1
4    POS      e.g.   11623
5    MAPQ     e.g.   3
6    CIGAR    e.g.   100M
7    RNEXT    e.g.   =
8    PNEXT    e.g.   11740
9    TLEN     e.g.   217
10   SEQ      e.g.   CCTGTTTCTCCACAAAGTGTTTACTTTTGGATTTTTGCCAGTCTAACAGGTGAAGCCCTGGAGATTCTTATTAGTGATTTGGGCTGGGGCCTGGCCATGT
11   QUAL     e.g.   CCCFFFFFHHHHHJJIJFIJJJJJJJJJJJJHIJJJJJJJJIJJJJJJGGHIJHIJJJJJJJJJJGHGGIJJJJJJJIJEEHHHHFFFFCDCDDDDDDDB@ACDD
```

SAM/BAM flags explained

- 12 bitwise flags describing the alignment

- Stored as a binary string of length 12 instead of 12 columns of data

- Value of '1' indicates the flag is set. e.g. 001000000000

- All combinations can be represented as a number from 0 to 4095 (i.e. $2^{12}-1$). This number is used in the BAM/SAM file.

- You can specify 'required' or 'filter' flags in samtools view using the '-f' and '-F' options respectively

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

Note that to maximize confusion, each bit is described in the SAM specification using its hexadecimal representation (i.e., '0x10' = 16 and '0x40' = 64).

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

CIGAR strings explained

- The CIGAR string is a sequence of base lengths and associated 'operations' indicating which bases align to the reference (either a match or mismatch), are deleted, are inserted, represent introns, etc.

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

' that

- e.g. 81M859N19M

  - A 100 bp read consists of: 81 bases of alignment to reference, 859 bases skipped (an intron), 19 bases of alignment

CRAM files

- CRAM is an ultra-compressed version of a BAM file
  - Usually between 30-60% smaller than the corresponding BAM

- Stores "diffs" from the reference genome
  - requires the matching reference genome to restore original data!

- Base quality binning may be used as well

- Some tools still require conversion back to bam

Quality Score Bins	Example of Empirically Mapped Quality Scores*
N (no call)	N (no call)
2–9	6
10–19	15
20–24	22
25–29	27
30–34	33
35–39	37
≥ 40	40

By replacing the quality scores between 19 and 25 with a new score of 22, data storage space is conserved.

*The mapped quality score of each bin (except "N") is subject to change depending on individual Q-tables.

Introduction to the BED format

- When working with BAM files, it is very common to want to examine a focused subset of the reference genome
    - e.g. the exons of a gene

- These subsets are commonly specified in 'BED' files
    - https://genome.ucsc.edu/FAQ/FAQformat.html#format1

- Many BAM manipulation tools accept regions of interest in BED format

- Basic BED format (tab separated):
    - Chromosome name, start position, end position
    - Coordinates in BED format are 0 based

# Manipulation of SAM/BAM and BED files

- Several tools are used ubiquitously in sequence analysis to manipulate these files

- SAM/BAM files
  - samtools
  - bamtools
  - Picard

- BED files
  - bedtools
  - bedops

Common sources of confusion

- Genomic coordinate systems

- Genome builds

- Variant representation

Genomic coordinates – 1 vs 0 based

chr1		T		A		C		G		T		C		A	

1-based: T=1, A=2, C=3, G=4, T=5, C=6, A=7

0-based: 0, 1, 2, 3, 4, 5, 6, 7
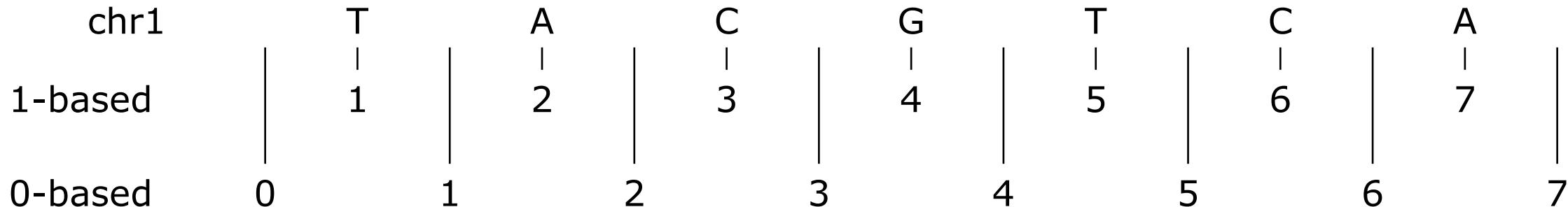
	1-based	0-based
Indicate a single nucleotide	chr1:4-4 G	chr1:3-4 G
Indicate a range of nucleotides	chr1:2-4 ACG	chr1:1-4 ACG
Indicate a single nucleotide variant	chr1:5-5 T/A	chr1:4-5 T/A

- 1-based : Single nucleotides, variant positions, or ranges are specified directly by their corresponding nucleotide numbers
  - GFF, SAM, VCF, Ensembl browser, …

- 0-based: Single nucleotides, variant positions, or ranges are specified by the coordinates that flank them
  - BED, BAM, UCSC browser, …
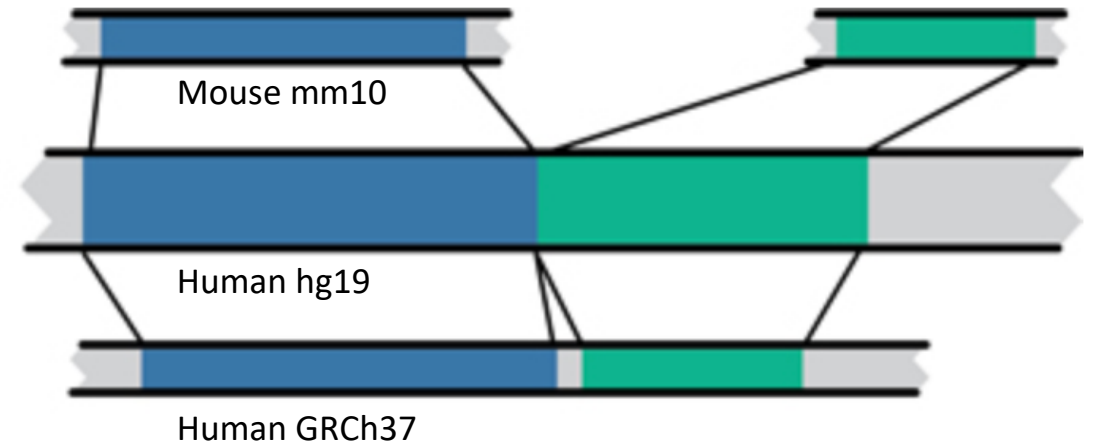
Genome builds

Reference Genome builds

Current human: GRCh38, hg38, b38
     alternate: GRCh38v2_ccdg

Previous human: GRCh37, hg19, b37

Current mouse: GRCm38, mm10

Lift-over

Mouse mm10

Human hg19

Human GRCh37

rnabio.org

Variant shifting (alignment) and parsimony/trimming

Reference and alternative alleles of a CA short tandem repeat (STR)

REF   GGGCACACA**CA**GGG
ALT   GGGCACACAGGG

← **CA** deletion from the reference

Genome Reference

GGGCACACACAGGG

REF	CA
ALT	.
REF	CAC
ALT	C
REF	GCACA
ALT	GCA
REF	GGCA
ALT	GG
REF	GCA
ALT	G

Alleles represented against the human genome reference. Allele pairs are colored the same, all are representations of the same variant.

Variant Call Format

POS	REF	ALT	
8	CA	.	Not left aligned and alternate allele is empty
6	CAC	C	Not left aligned but parsimonious
3	GCACA	GCA	Not right trimmed
2	GGCA	GG	Not left trimmed
3	GCA	G	Normalized (left aligned & parsimonious)

Alleles represented in Variant Call Format, all are representations of the same variant.

Parsimony: representing variant in as few nucleotides as possible without reducing the length of any allele to 0

Left (right) aligning = shifting the start position of a variant as far to the left (right) as possible

How should I sort my SAM/BAM file?

- Generally BAM files are sorted by <u>position</u>
  - This is for performance reasons
    - When sorted and indexed, arbitrary positions in a massive BAM file can be accessed rapidly

- Certain tools require a BAM sorted by <u>read name</u>
  - Usually this is when we need to easily identify both reads of a pair
    - The insert size between two reads may be large
    - In fusion detection we are interested in read pairs that map to different chromosomes

rnabio.org