

# BAM format

Alistair Ward  
USTAR Center for Genetic Discovery  
University of Utah

# What are we going to discuss?

## BAM format

- Quick recap - how are BAM files generated?
- Description of format
- Important attributes
- How to manipulate BAM files



# FASTA format

We start with a reference genome to map to

The reference sequence  
(chromosome)

Sequence description

>20 dna:chromosome chromosome:GRCh37:20:1:63025520:1  
NNNNNNNNNTACTTCGATTGCGTATTTACGGACGTAGCGAGTCTTTAGAGTCTTTTAGTCTGTATC  
GTCGTAGTGTCAGTTCGTAGTCTATGTCGTATTCGTAGGCGTACGTAGTCGTGTAGTCAGTCGTGTT

DNA sequence



# FASTQ format

and sequence reads to map

DNA sequence

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
GTCGTAGTGTCAGTTCGTAGTCTATGTCGTATTCGTAGGCGTACGTAGTCGTGTAGTCAGTCGTGTT
+
!#!#!!*!#^^!#(#*&^*$^^#%&$*$&(^&$^^^$)$*&$**$))$*&^$*##$))$)))))
```

Quality scores

$$Q = -10 \log_{10}(P_{\text{err}})$$

<b>EAS139</b>	the unique instrument name
<b>136</b>	the run id
<b>FC706VJ</b>	the flowcell id
<b>2</b>	flowcell lane
<b>2104</b>	tile number within the flowcell lane
<b>15343</b>	x'-coordinate of the cluster within the tile
<b>197393</b>	y'-coordinate of the cluster within the tile
<b>1</b>	the member of a pair, 1 or 2 ( <i>paired-end or mate-pair reads only</i> )
<b>Y</b>	Y if the read is filtered, N otherwise
<b>18</b>	0 when none of the control bits are on, otherwise it is an even number
<b>ATCACG</b>	index sequence

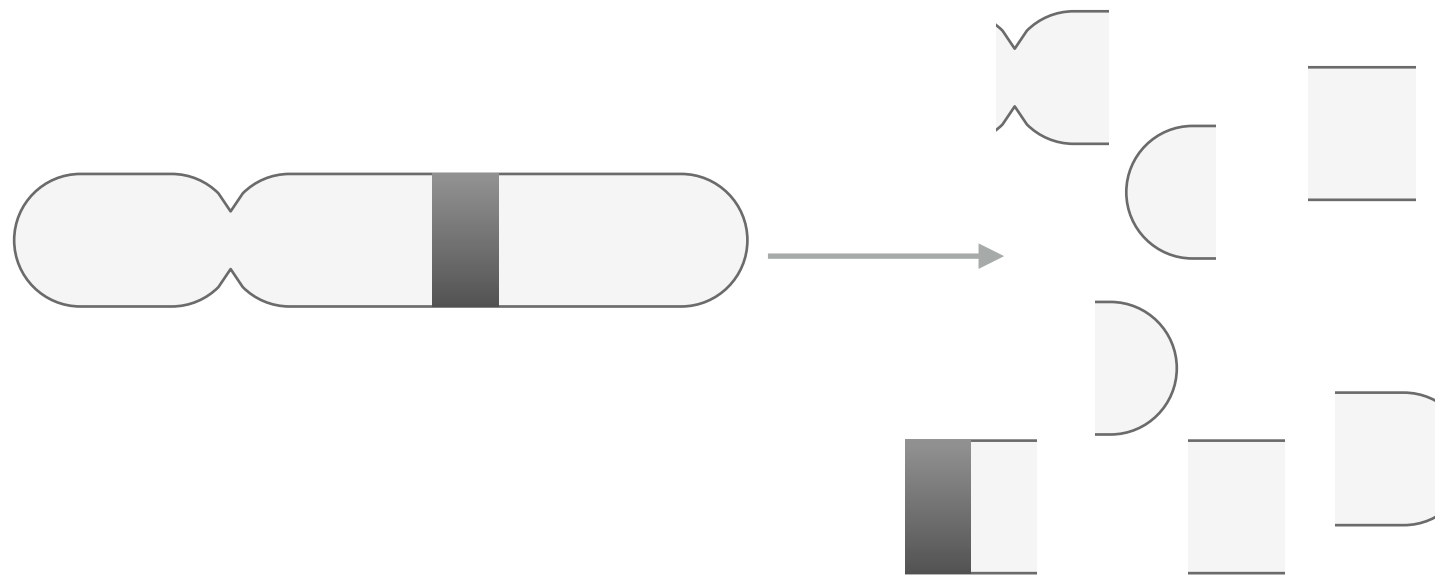


# Generate sequence reads

Fragment a genome —> DNA library

PCR amplification

Sequence reads (ends of DNA fragment for mate pairs)

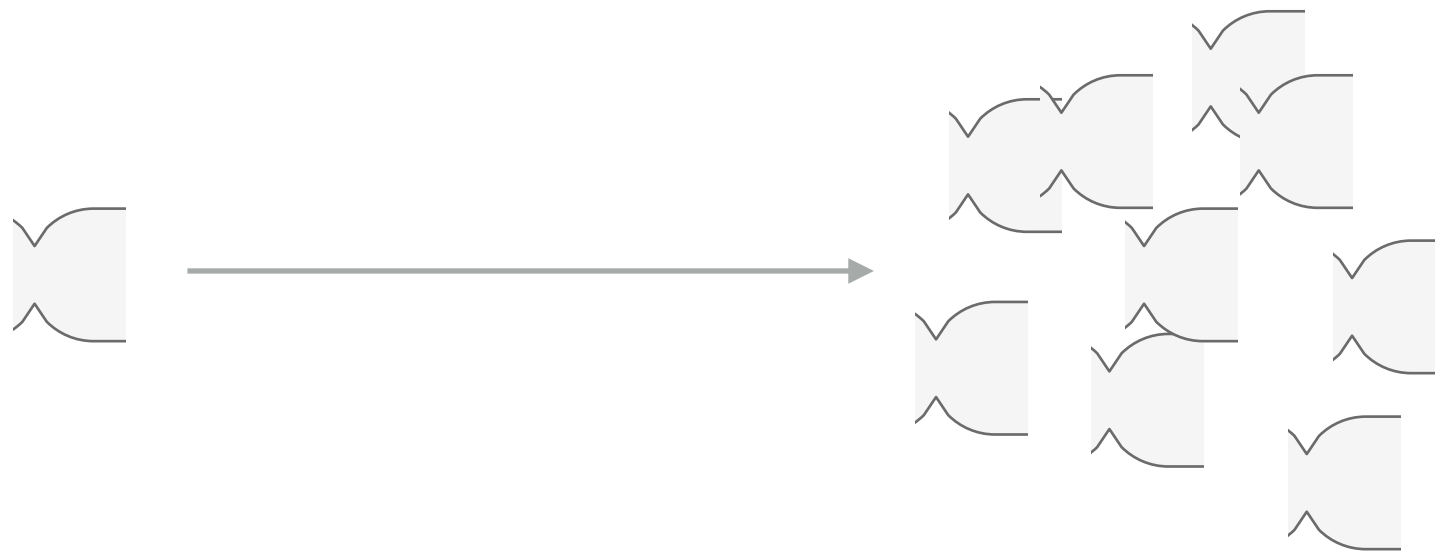


# Generate sequence reads

Fragment a genome —> DNA library

PCR amplification

Sequence reads (ends of DNA fragment for mate pairs)

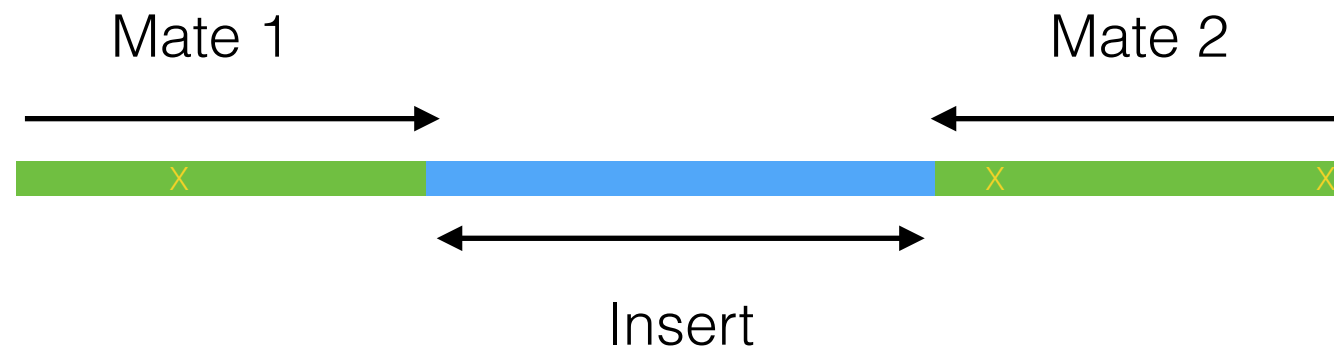


# Generate sequence reads

Fragment a genome —> DNA library

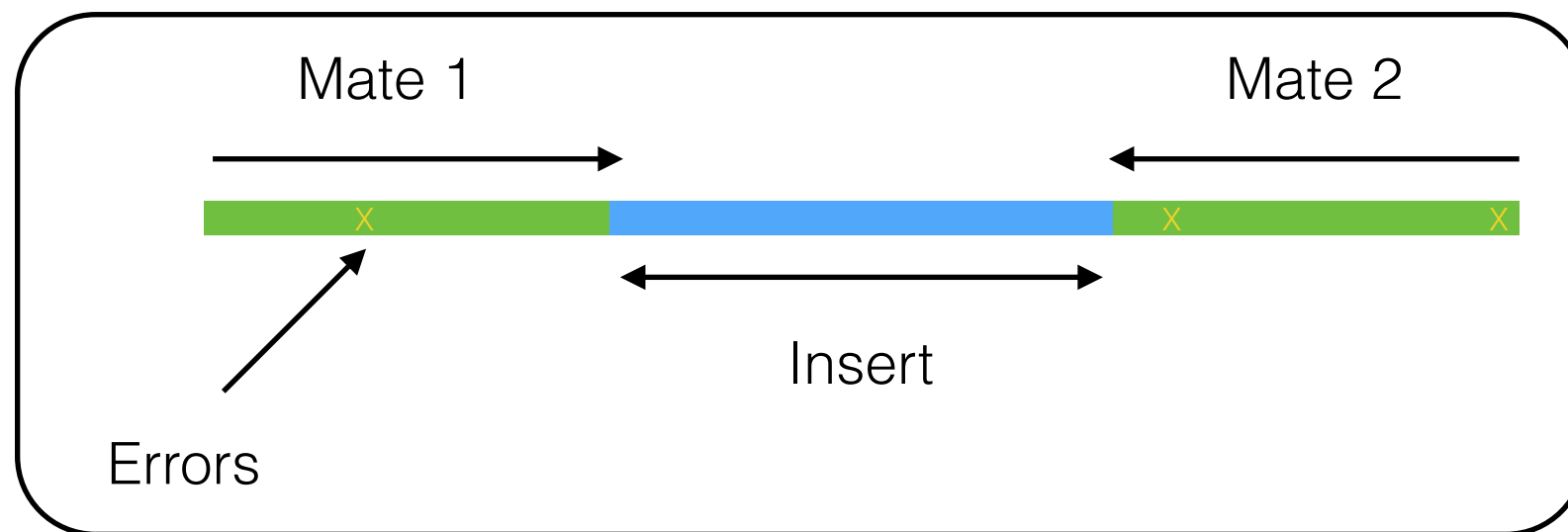
PCR amplification

Sequence reads (ends of DNA fragment for mate pairs)



# Generate sequence reads

We no longer have any positional information or relational information between fragments



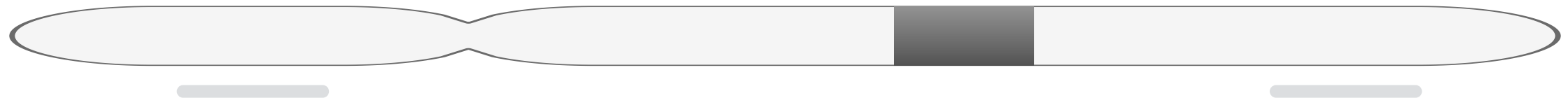
Store the read information in FASTQ file





# Mapping to a reference genome

This is like a jigsaw puzzle

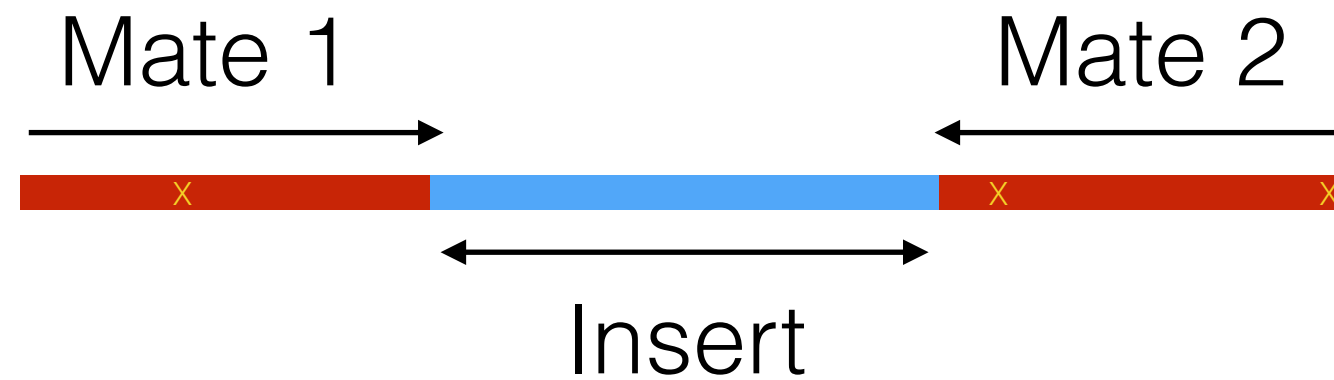


Could fit here - but there are differences




Could fit here as well.



# Paired end reads

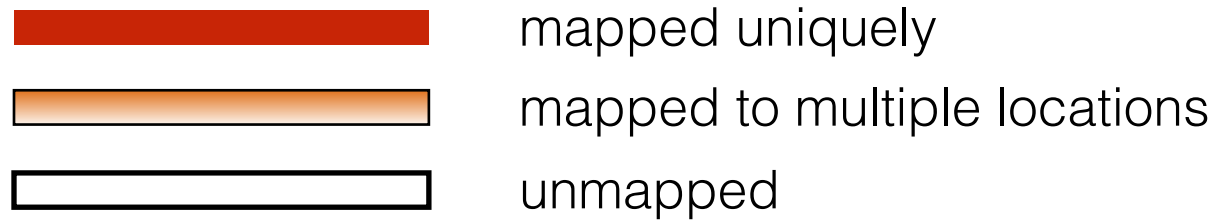
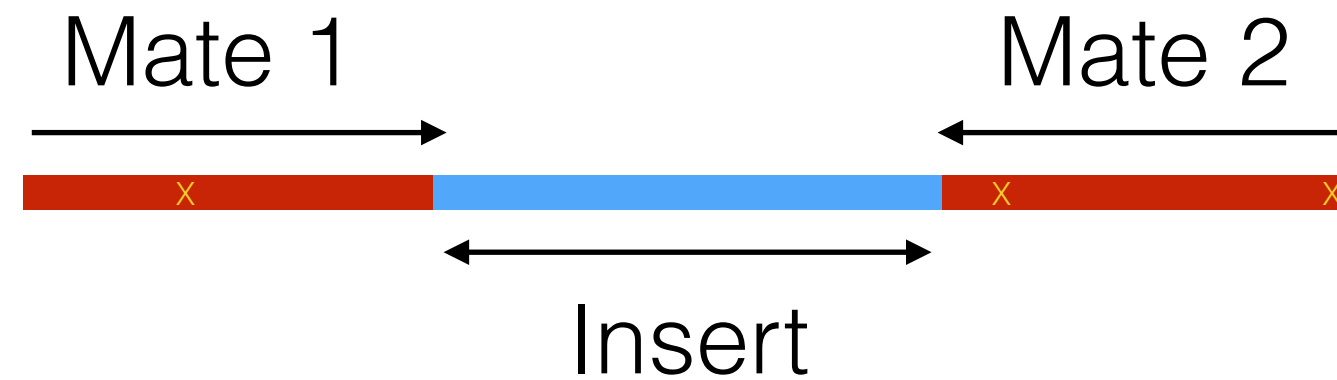


---

	mapped uniquely
	mapped to multiple locations
	unmapped



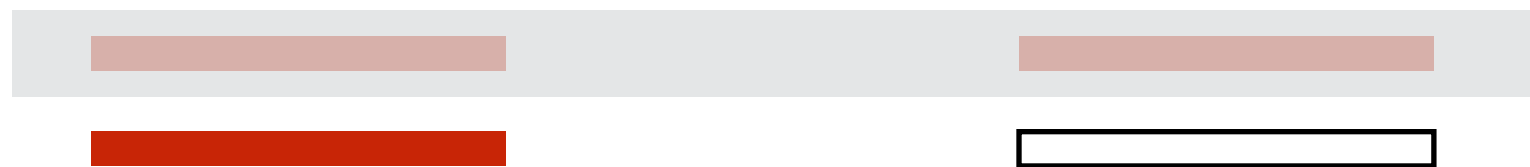
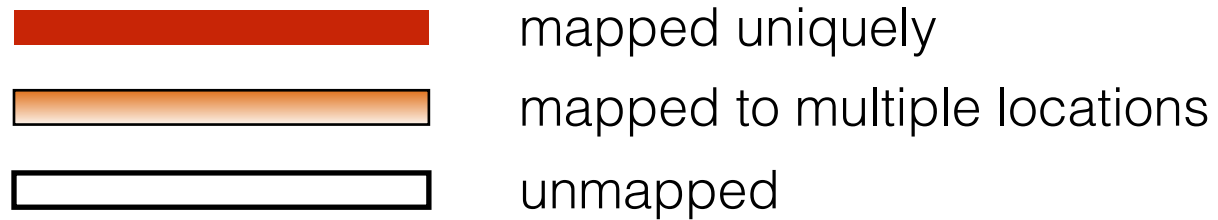
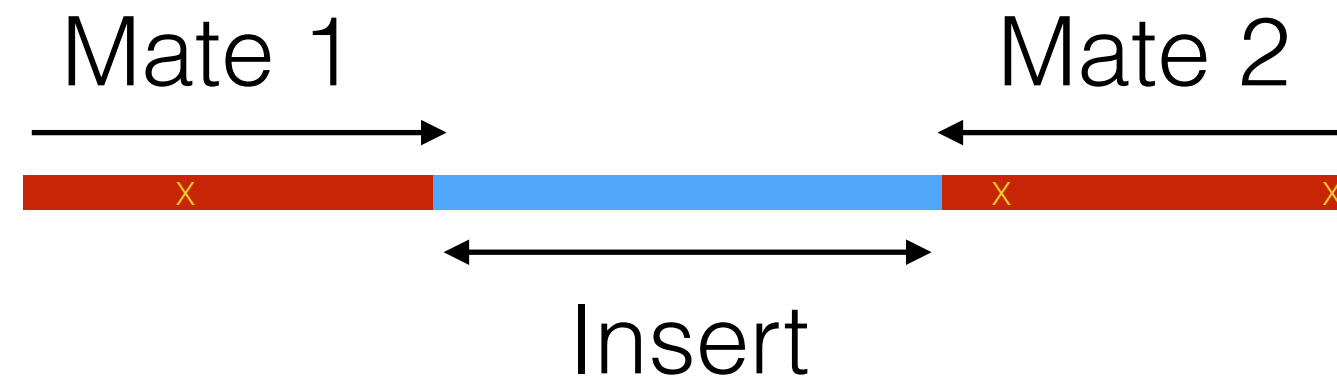
# Paired end reads



Both mates map uniquely



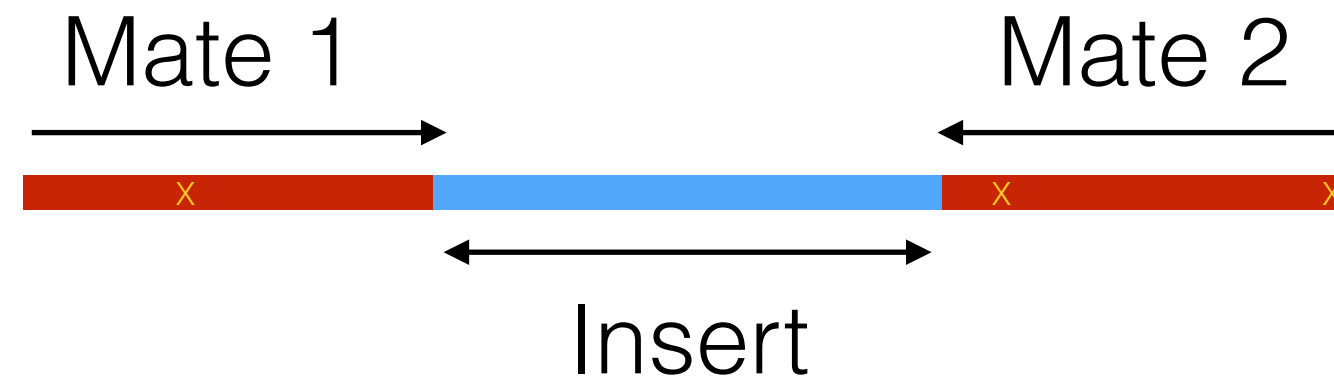
# Paired end reads






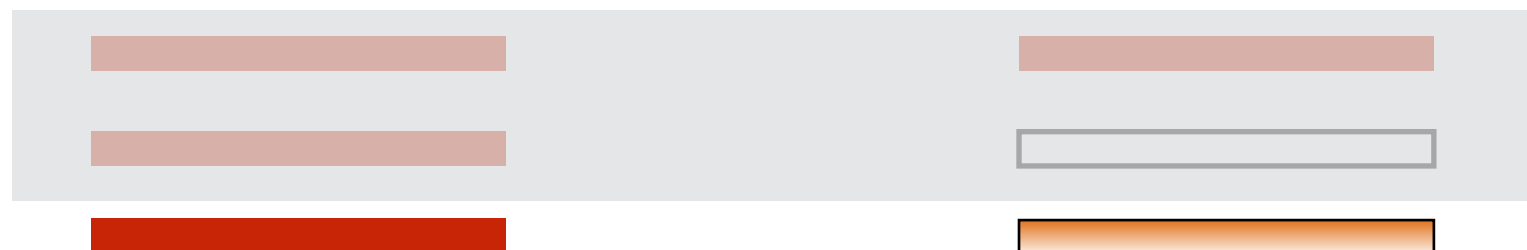
One mate maps uniquely, the other is unmapped



# Paired end reads

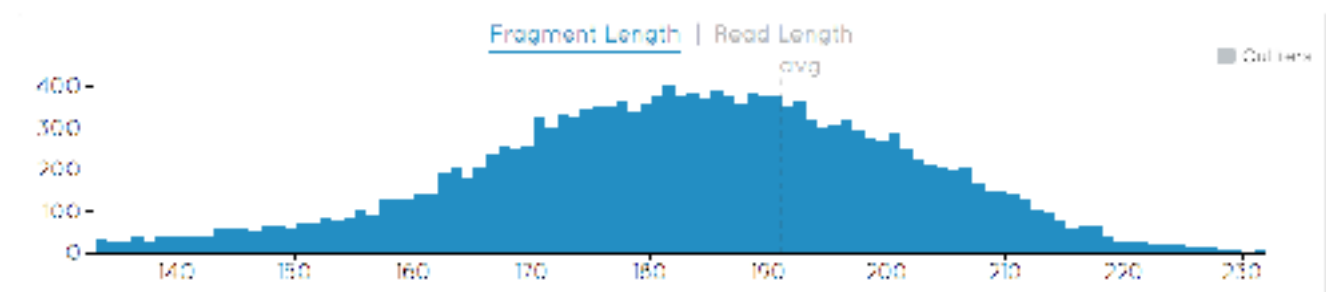


-  mapped uniquely
-  mapped to multiple locations
-  unmapped

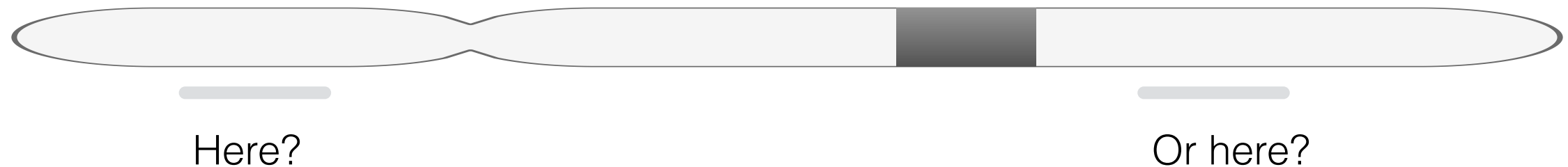


One mate maps uniquely, the other maps to multiple locations

Use fragment length distribution to determine most likely location



# What needs to be stored?



Where did the read map?

How confident are we that we are correct?

Which strand does the read come from?

Are there any differences with the reference?

What is the DNA sequence?

What are the quality scores for each base in the read?

What do we know about the mate?

Which read group does the read belong to?



# What needs to be stored?



Where did the read map?

How confident are we that we are correct?

Which strand does the read come from?

Are there any differences with the reference?

What is the DNA sequence?

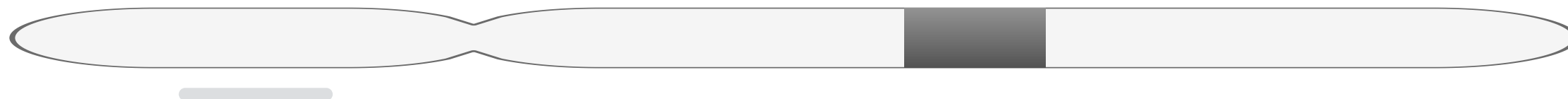
What are the quality scores for each base in the read?

What do we know about the mate?

Which read group does the read belong to?



# What needs to be stored?



Where did the read map?

How confident are we that we are correct?

Which strand does the read come from?

Are there any differences with the reference?

What is the DNA sequence?

What are the quality scores for each base in the read?

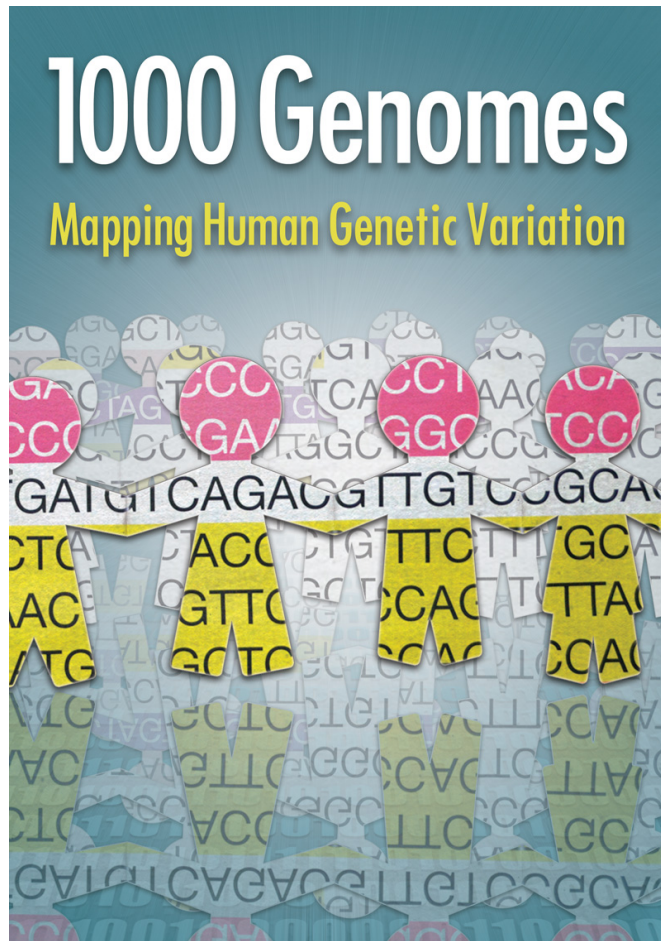
What do we know about the mate?

Which read group does the read belong to?





# Store the alignment



Standardise alignment formats

SAM - Sequence Alignment/Map

- Compressed (BAM) saves space
- Can be indexed allowing fast access of regions
- Simple format
- Can represent single and paired end reads
- Many toolkits now available to process data



# SAM format

Version (VN) and sort order (SO) - Important!

Reference sequence (SQ) and sequence length (LN)

```
@HD VN:1.3 SO:coordinate
@SQ SN:20 LN:63025520
@RG ID:HG00096 SM:HG00096
@PG ID:HG00096 PN:bwa CL:/Users/AlistairNWard/Work/gkno/gkno_launcher/tools/bwa/bwa mem -t 4
```

Read group (RG) and sample (SM)

Programs (PG) that have been run on the data



# SAM format

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

1	2	3	4	5	6	7	8	9	10
SRR062634.14576120	163	20	899919	60	100M	=	900037	218	TTCCCCAGTAGCTGGGATTACAGGCATACGCCACCATC

?

??



# Flag



Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate
0x800	supplementary alignment

Secondary alignment - an alternative location for the read (sequence and quality strings are '\*')

Supplementary alignments - read is split into a set of alignments. All but one are represented as supplementary



# Mapping quality

Quality scores

$$Q = -10 \log_{10}(P_{\text{err}})$$

1 in a 100 chance that we mismapped:  $Q = 20$

1 in a 1,000 chance that we mismapped:  $Q = 30$

If  $Q = 0$ , what is the interpretation?



# Mapping quality

Quality scores

$$Q = -10 \log_{10}(P_{\text{err}})$$

1 in a 100 chance that we mismapped:  $Q = 20$

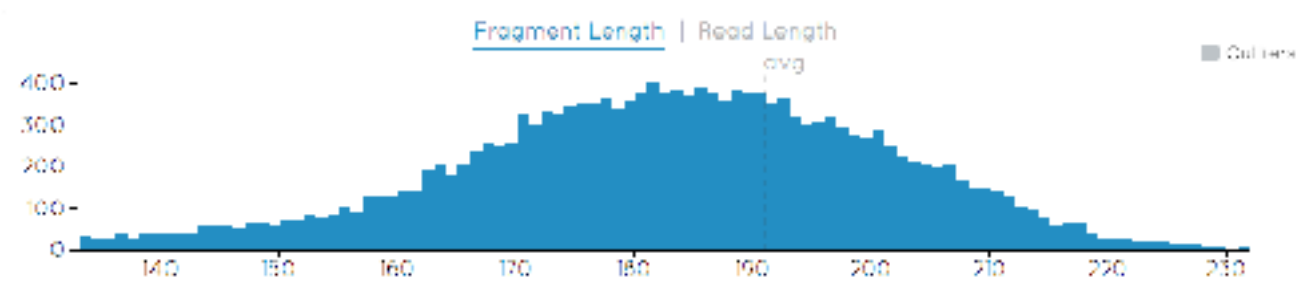
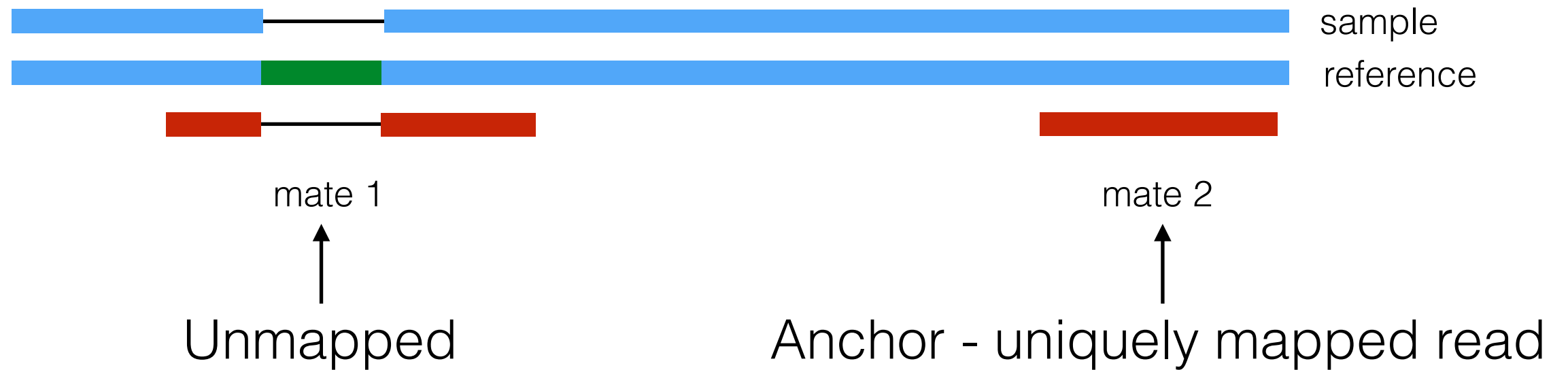
1 in a 1,000 chance that we mismapped:  $Q = 30$

If  $Q = 0$ , what is the interpretation?

The read is multiply mapped. We are not confident of where the read goes, e.g. mobile elements

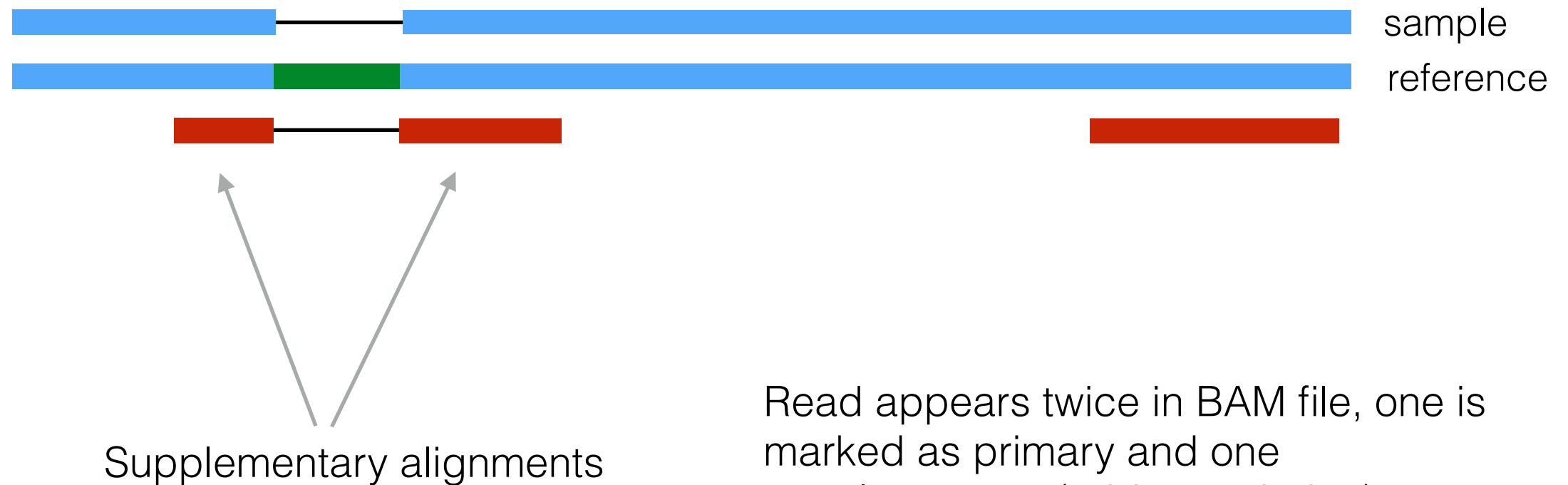


# Split read mapping



Estimate of fragment length -  
we have an idea of where to look

# Split read mapping



Read appears twice in BAM file, one is marked as primary and one supplementary (arbitrary choice)

Hard clipped reads

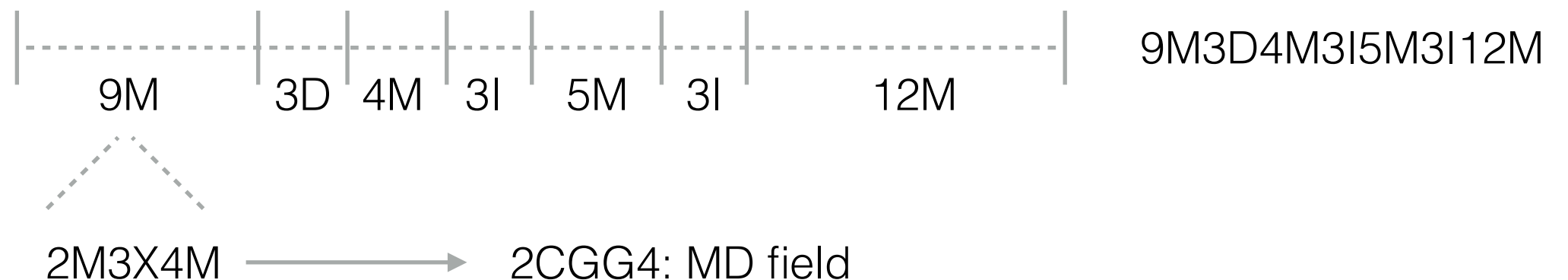
e.g. 30M70H and 30H70M



# CIGAR string

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

Reference: ACTTTTCATCCCTAAA---CAACC---CTGTGTTTCCC  
Sample: AC**CGG**TCAT---TAA**TT**CAACC**CTT**CTGTG**AAA**TCCC



# CIGAR - Clipped reads

Reduced coverage



e.g. 20S80M

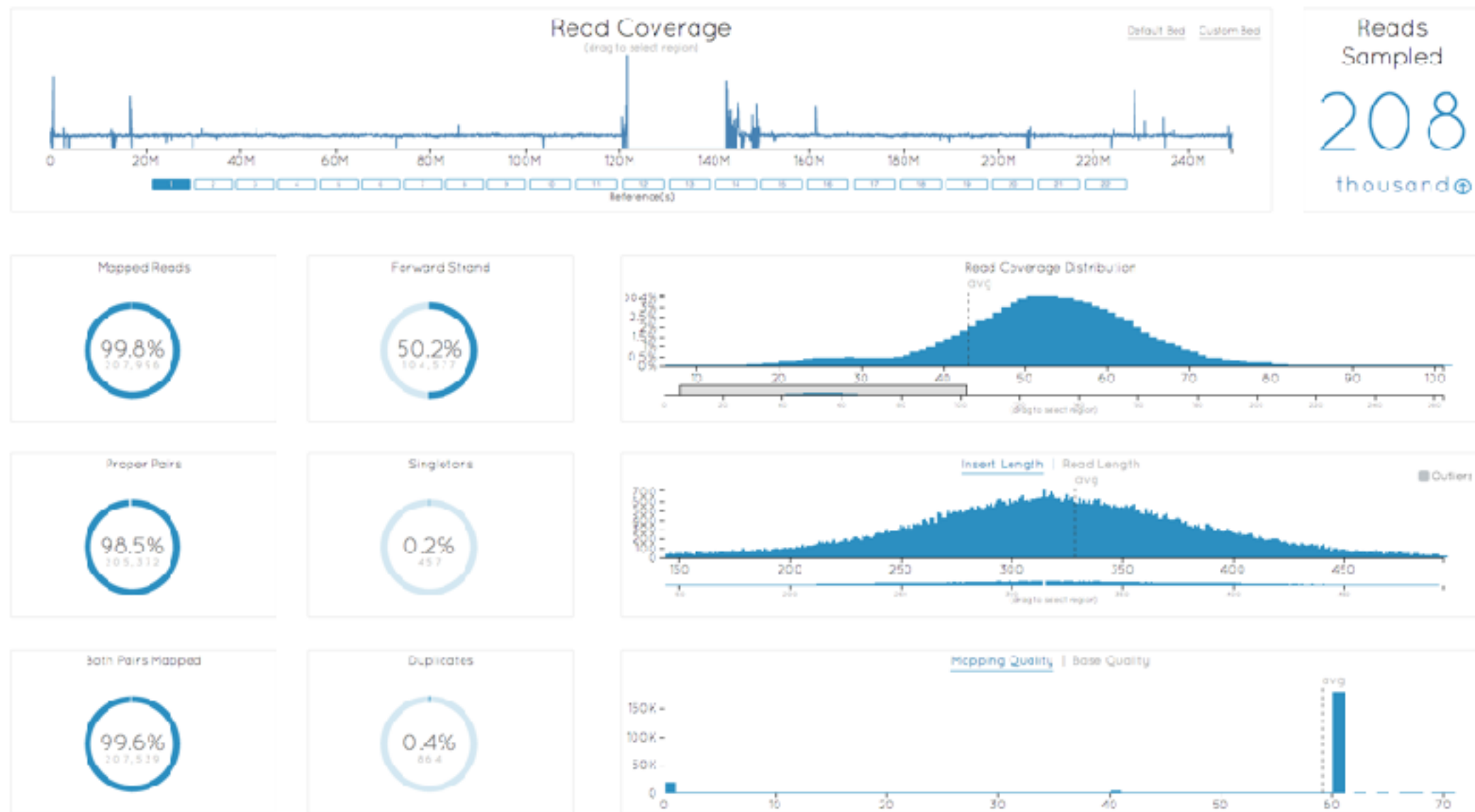
e.g. 84M16S



# Visualizing a BAM file

bam.iobio.io

an iobio project



# CRAM format

- Developed at EBI
- Significantly better lossless compression than BAM
- Fully compatible with BAM
- Tools to allow transition between BAM  $\longleftrightarrow$  CRAM
- Ability to generate *lossy* files
  - Calculable tags, eg. MD tag
  - Quality score compression



# Illumina 8 bin compression

Replace Q = [2 - 9] with 6

Table 1: Q Scores Based upon an Optimized 8-level Mapping

Old Quality Score	New Quality score
N (no cal)	N (no cal)
2-9	6
10-19	15
20-24	22
25-29	27
30-34	33
35-39	37
≥ 40	40

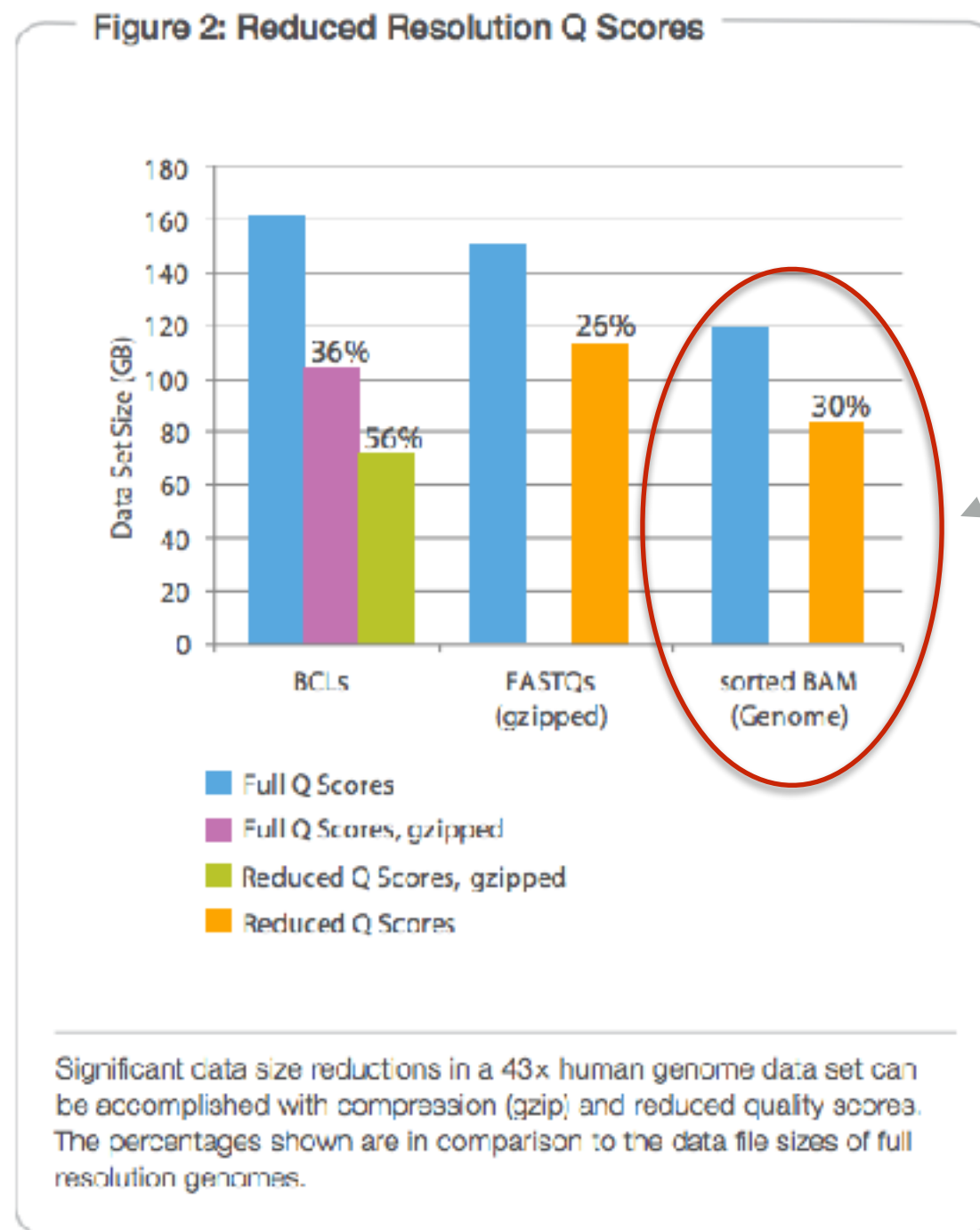
Original Q string: 2 5 8 3 3 12 15 38 42 49 56 59 58 62 ...

8-bin Q string: 6 6 6 6 6 15 15 37 40 40 40 40 40 40 ...

New string is more compressible



# Illumina 8 bin compression



43X Genome:  
30% reduction

# Illumina 8 bin compression

Table 2: Reduced Resolution Q Scores

Resolution	Sensitivity (%)	Conflicts*	Specificity (%)
Full (Elandv2e+CASAVA Variant Calling)	95.29	5,419	99.999788
Reduced (Elandv2e+CASAVA Variant Calling)	95.56	5,940	99.999792
Full (BWA + GATK)	98.40	16,766*	99.999365
Reduced (BWA + GATK)	98.40	17,400*	99.999341

\* Absolute conflict numbers cannot be directly compared, due to the different filters and thresholds used by different tools. It is the relative performance with full and reduced Q score resolution that is of interest.

Sequenced trio at 40X

Mendelian SNP conflicts

No 'significant' differences in calling quality using reduced quality scheme



# Hands-on with BAM files

The gkno website has a tutorial for playing with BAM files

<http://gkno.me/cshl.html>



[http://supportres.illumina.com/documents/myillumina/e96e90a9-698d-4a0b-9b33-9445c5ad723d/whitepaper\\_datacompression.pdf](http://supportres.illumina.com/documents/myillumina/e96e90a9-698d-4a0b-9b33-9445c5ad723d/whitepaper_datacompression.pdf)