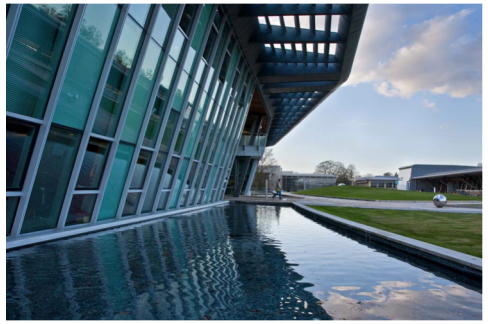
NGS read alignment

Vivek Iyer
Adams Group (Exp Cancer Genetics)
WTSI
vvi@sanger.ac.uk

In the footsteps of Anthony Doran



wellcome trust sanger institute EMBL-EBI

NGS read alignment

The most important first step in understanding next-generation sequencing data is the initial alignment or assembly that determines whether an experiment has succeeded and provides a first glimpse into the results.

Flicek & Birney, 2009. *Nature Methods*

Sequence alignment in NGS is:

- *Process of determining the most likely source of the observed DNA sequencing read within the reference genome sequence.*

Principles and approaches to sequence alignment have not changed much since 80's

Overview

Intro

Methods / Aligners

Alignment Outputs

Alignment Viewers

NGS Workflows, QC and BAM Improvement

The *reference* is *key* in all that follows

Sequence alignment in NGS is:

- *Process of determining the most likely source of the observed DNA sequencing read within the* ***reference genome sequence***

The *reference* is *key* in all that follows

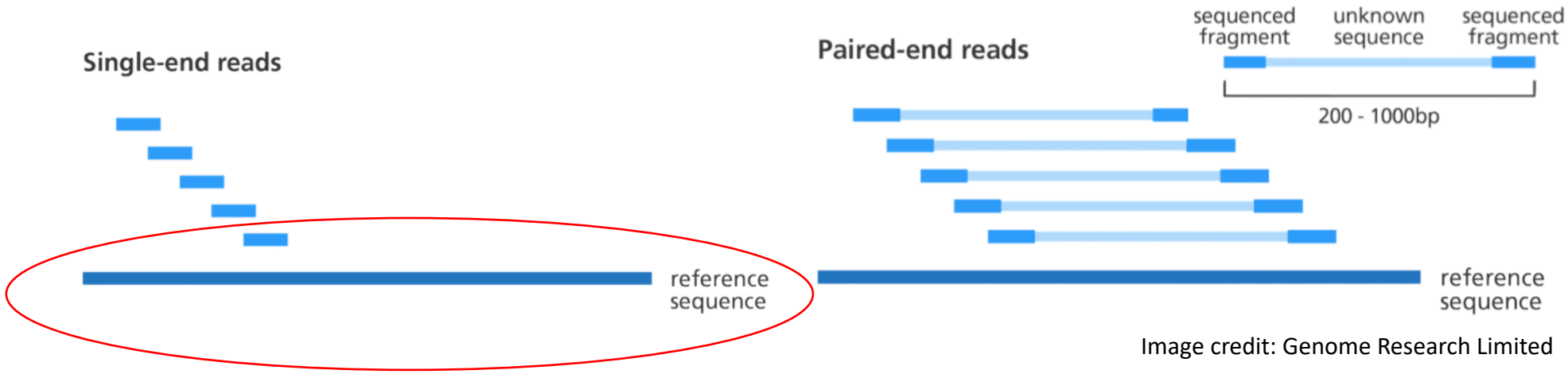


Image credit: Genome Research Limited

The view of alignment in *this* lecture is *wildly* asymmetric:

- short read sequences
- very long reference sequences. Genomes (~Gbp) or Transcriptomes (~100Mbp)

Where do references come from?

Organism => Consortium => Money => Sequencing => Assembly => Curation=> Dissemination => Curation, Dissemination => …

Human, Mouse, Zebrafish, Chicken:

Lots of curation / dissemination was done at various centres and distributed by NCBI. NCBI still distributes!

Now the umbrella for curation / patching is: **Genome Reference Consortium**

The *reference* is *key* in all that follows

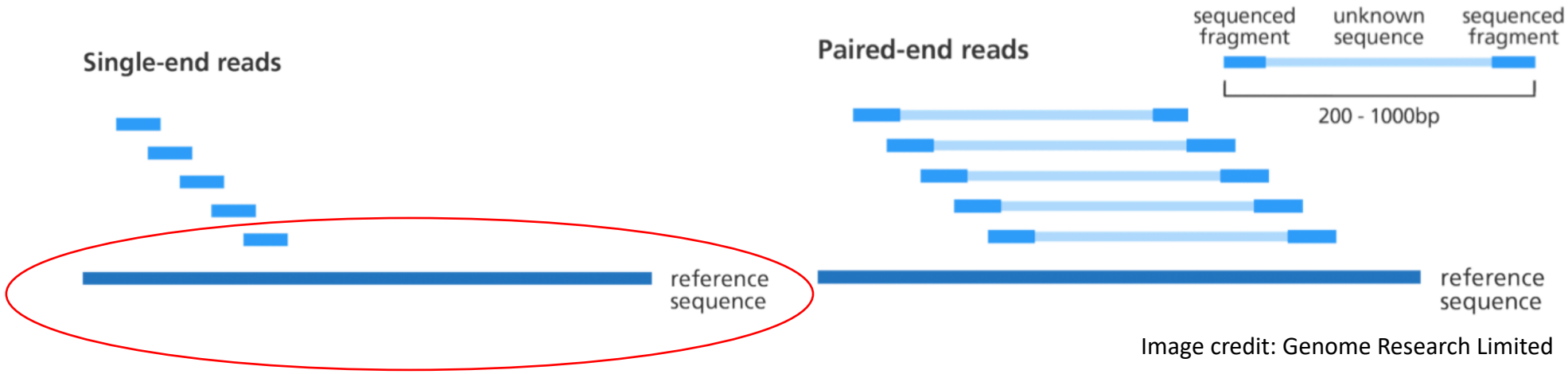


Image credit: Genome Research Limited

HUMAN reference sequences

Release name	Date of release	Equivalent UCSC version
GRCh38	Dec 2013	hg38
GRCh37	Feb 2009	hg19
NCBI Build 36.1	Mar 2006	hg18
NCBI Build 35	May 2004	hg17
NCBI Build 34	Jul 2003	hg16

MOUSE reference sequences

Release name	Date of release	Equivalent UCSC version
GRCm38	Dec 2011	mm10
NCBI Build 37	Jul 2007	mm9
NCBI Build 36	Feb 2006	mm8
NCBI Build 35	Aug 2005	mm7
NCBI Build 34	Mar 2005	mm6

The actual reference is a just a (big) sequence (fasta) file: $ ls –h GRCh38.fa
> 3.1G GRCh38.fa

Why align?

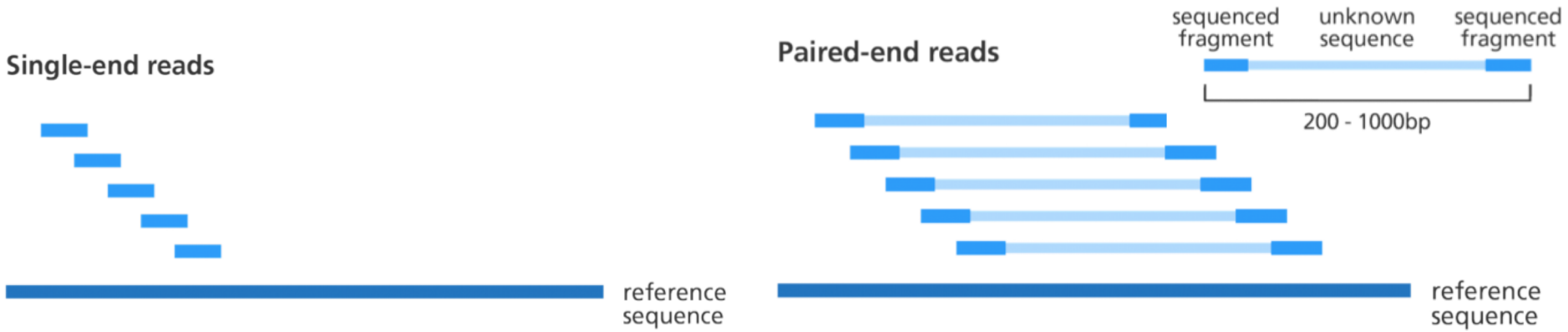
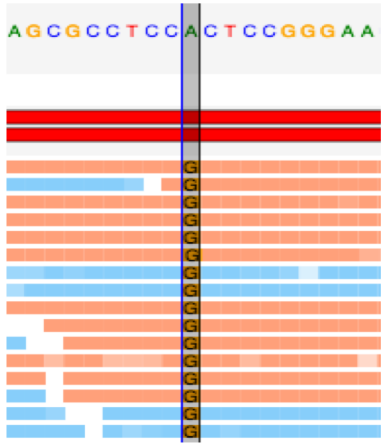
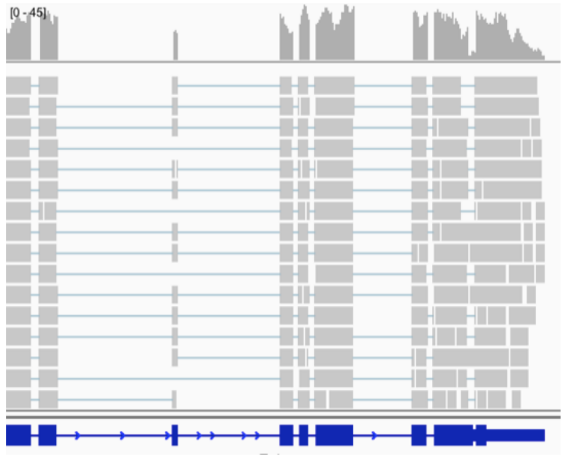


Image credit: Genome Research Limited

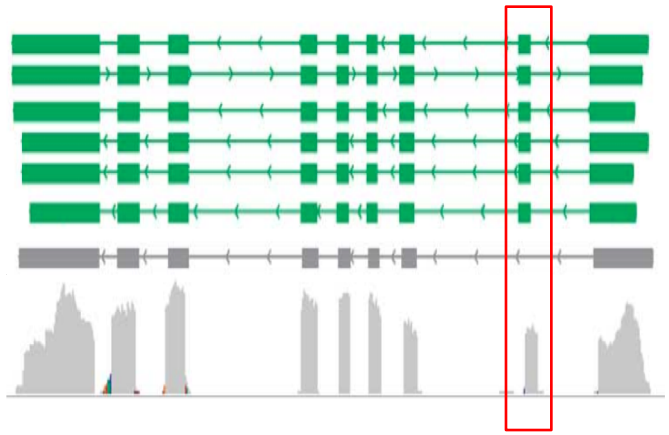
Identify variation



Transcript abundance



Discover transcribed sequence



Overview

Intro

Methods / Aligners

Alignment Outputs

Alignment Viewers

NGS Workflows, QC and BAM Improvement

Local alignment: Smith-Waterman algorithm (1981)

Algorithm for generating the optimal pairwise alignment between two sequences

Optimal alignment: the alignment which exhibits the most correspondences and the least differences, i.e. the alignment with the highest mapping score

Time consuming to carry out for every read - impractical

- Aligner will use it to refine a quick approximate placement
- Variant caller might use it to correctly re-align reads with insertions/deletions

We will now get a feel for this

Local alignment: Smith-Waterman algorithm (1981)

We get a feel for this by aligning things by eye, then doing it using S-W:

QUERY SEQUENCES:
ATG

REFERENCE SEQUENCE:
ATCG

ALIGNMENT:
```
AT-G
||  |
ATCG
```

QUERY SEQUENCES:
ACCG

REFERENCE SEQUENCE:
ATCG

ALIGNMENT:
```
ACTG
|| ||
ATCG
```

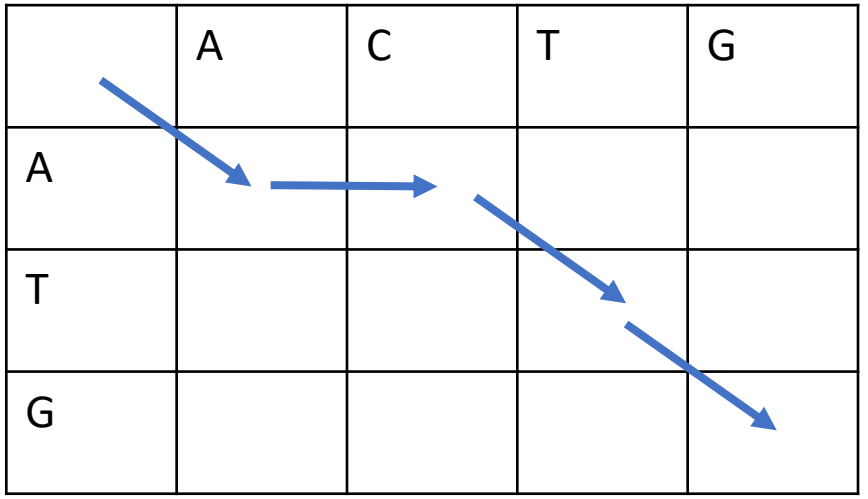
Local alignment: Smith-Waterman algorithm (1981)

We get a feel for this by aligning things by eye, then doing it using S-W:

QUERY SEQUENCES:
ATG

REFERENCE SEQUENCE:
ATCG

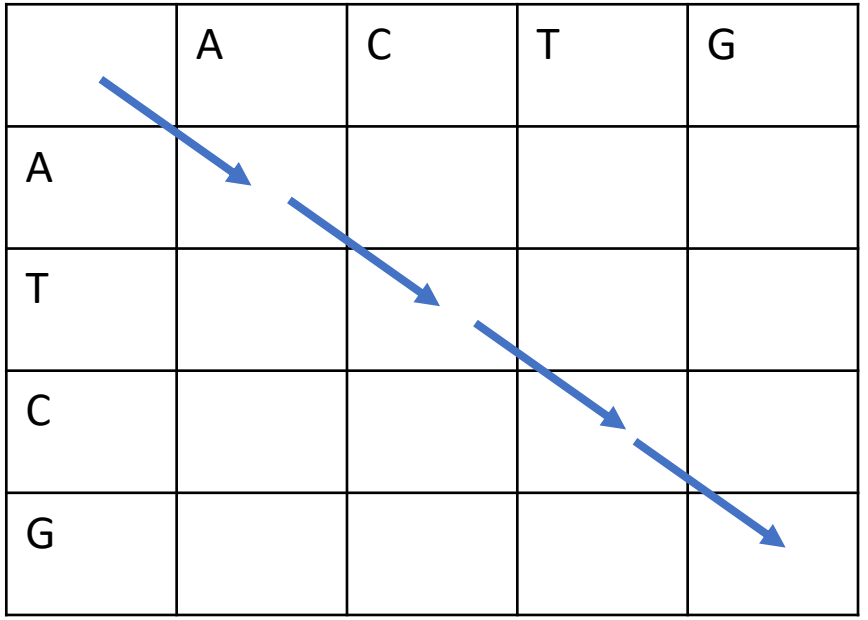
ALIGNMENT:
```
A-TG
|  ||
ATCG
```



QUERY SEQUENCES:
ACCG

REFERENCE SEQUENCE:
ATCG

ALIGNMENT:
```
ACTG
|||||
ATCG
```



Local alignment: Smith-Waterman algorithm (1981)

Algorithm for generating the optimal pairwise alignment between two sequences:

Score every cell in matrix from top to bottom.
Scoring is “cumulative” from previous cell:

Each cell has three possible scores:
Entry from diagonal (match or mismatch)
Entry from left (gap open)
Entry from above (gap open)

Accumulate this score to previous cell, based on match/mismatch/gap
Match: +1
Mismatch: -1
Gap Open: -1
- 3 possible scores based on 3 previous cells
Put “top” cumulative score in current cell.
If top score is negative, replace with 0
Record the *source* of top score (which cell you entered from)

		A	C	T	G
	0	0	0	0	0
A	0	-1,-1,1 = 1	0,-1,-1 = 0	-1,-1,-1 = -1 = 0	-2,-1,-1 = -1 = 0
C	0	-1,0,-1 = 0	-1,-1,2 =2	1,-1,-1 = 1	0, -1, -1 = 0
G	0	-1,-1,-1 = -1 = 0	-1,1,-1 = 1	0, 0 ,1 = 1	-1, -1, 2 = **2**

wellcome trust sanger institute

Local alignment: Smith-Waterman algorithm (1981)

Algorithm for generating the optimal pairwise alignment between two sequences:

TRACEBACK:

Start at highest score (2) and trace back to source until you get to 0 score.

WRITE OUT ALIGNMENT:

```
S1    A    C    T    G
S2    A    C    -    G
```

At each point of S1/S2: either a character or ”-”
Entry from sides = gap
Entry from diagonal = match or mismatch

		A	C	T	G
	0	0	0	0	0
A	0	-1,-1,1 = 1	0,-1,-1 = 0	-1,-1,-1 = -1 = 0	-2,-1,-1 = -1 = 0
C	0	-1,0,-1 = 0	-1,-1,2 =2	1,-2,-1 = 1	0, -1, -2 = 0
G	0	-1,-1,-1 = -1	-2,1,-1 = 1	0, 0 ,1 = 1	-1, -1, 2 = 2

Local alignment: Smith-Waterman algorithm (1981)

Algorithm for generating the optimal pairwise alignment between two sequences

Optimal alignment: the alignment which exhibits the most correspondences and the least differences, i.e. the alignment with the highest mapping score

Time consuming to carry out for every read

- Only applied to a small subset of the reads that don't have an exact match
- Important for correctly aligning reads with insertions/deletions

We will now get a feel for this (done!)

Further concepts:

Local alignment: Any segment of one sequence can align to an arbitrary position in the other sequence. **Segments** of query align to **segments** of target.

Global alignment: Alignment of two complete sequences. *If sequences are very dissimilar in size then there will be lots of gaps*.

NGS read alignment

NGS: Nucleotide based alignment (also known as read mapping)

Number of 150bp reads in an 40x coverage Illumina X10 genome sequence?
800x10^6

These all have to be aligned against a mammalian genome (3Gbp)

Two primary approaches:

- Hash-based alignment
- Suffix/prefix trees

Hash table alignment

Note: K-mer is a short fixed sequence of nucleotides

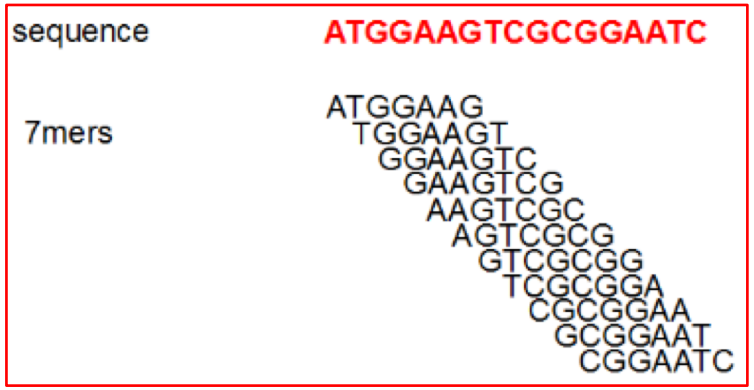
First thing: “wrap up” the reference genome to bring it to the reads – build a kmer hash

Scan the reference genome:

Build a profile (index) of all possible k-mers of length *n*

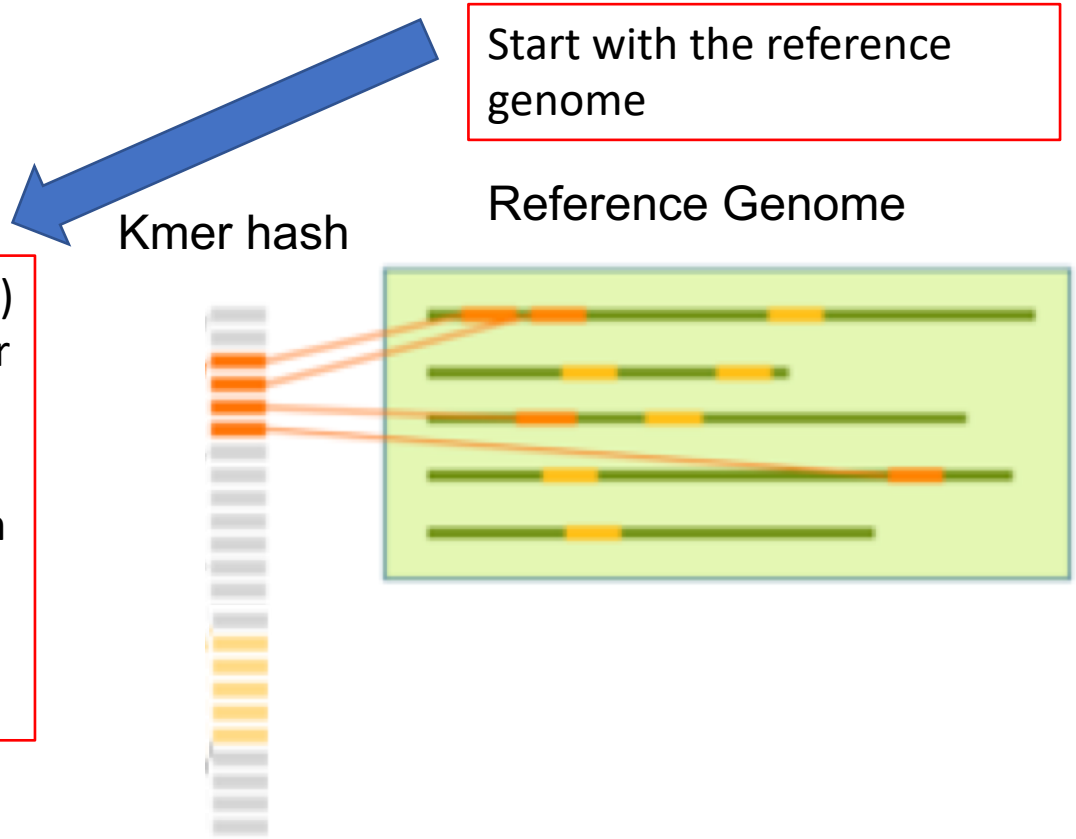
and the locations in the reference genome they occur:

Hash table will be several Gbytes in size for human genome



Hash table alignment

Start with the reference genome



Build a data structure (hash) allows rapid lookup by kmer sequence.

Each kmer entry in the hash stores the genome location(s) the kmer came from.

Hash table alignment

Note: K-mer is a short fixed sequence of nucleotides

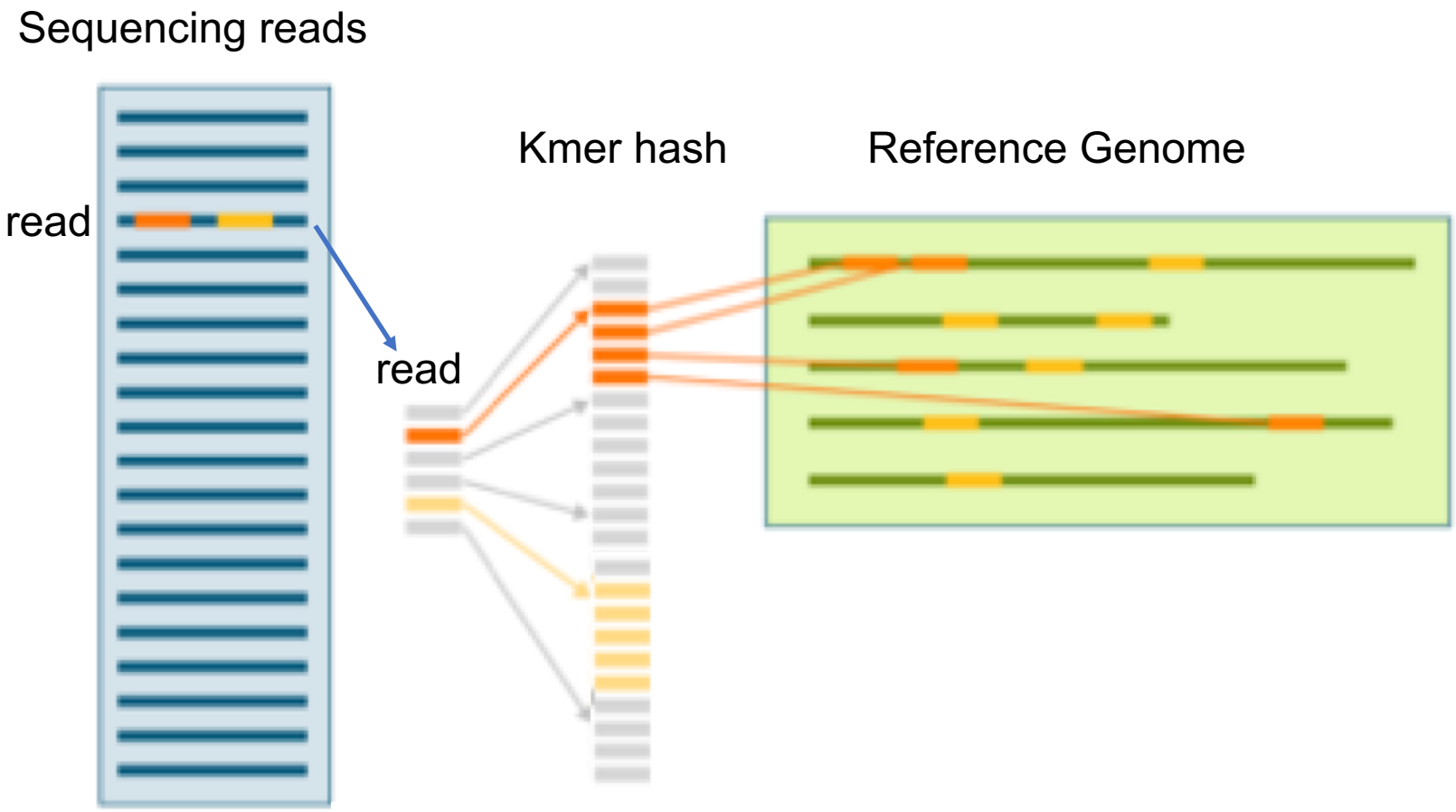
First thing: "wrap up" the reference genome to bring it to the reads – build a kmer hash
Scan the reference genome:
Build a profile (index) of all possible k-mers of length *n*
and the locations in the reference genome they occur:
Hash table will be several Gbytes in size for human genome

Next: For each sequence read:
Split the read into k-mers of length *n*
Lookup the locations in the reference via the index hash table (you made above)
Your hash table stores genomic locations => Pick region on the genome with most k-mer hits
Perform Smith-Waterman alignment to fully align the read to the region
Output the alignment of each read onto the reference in BAM (or equivalent) format.

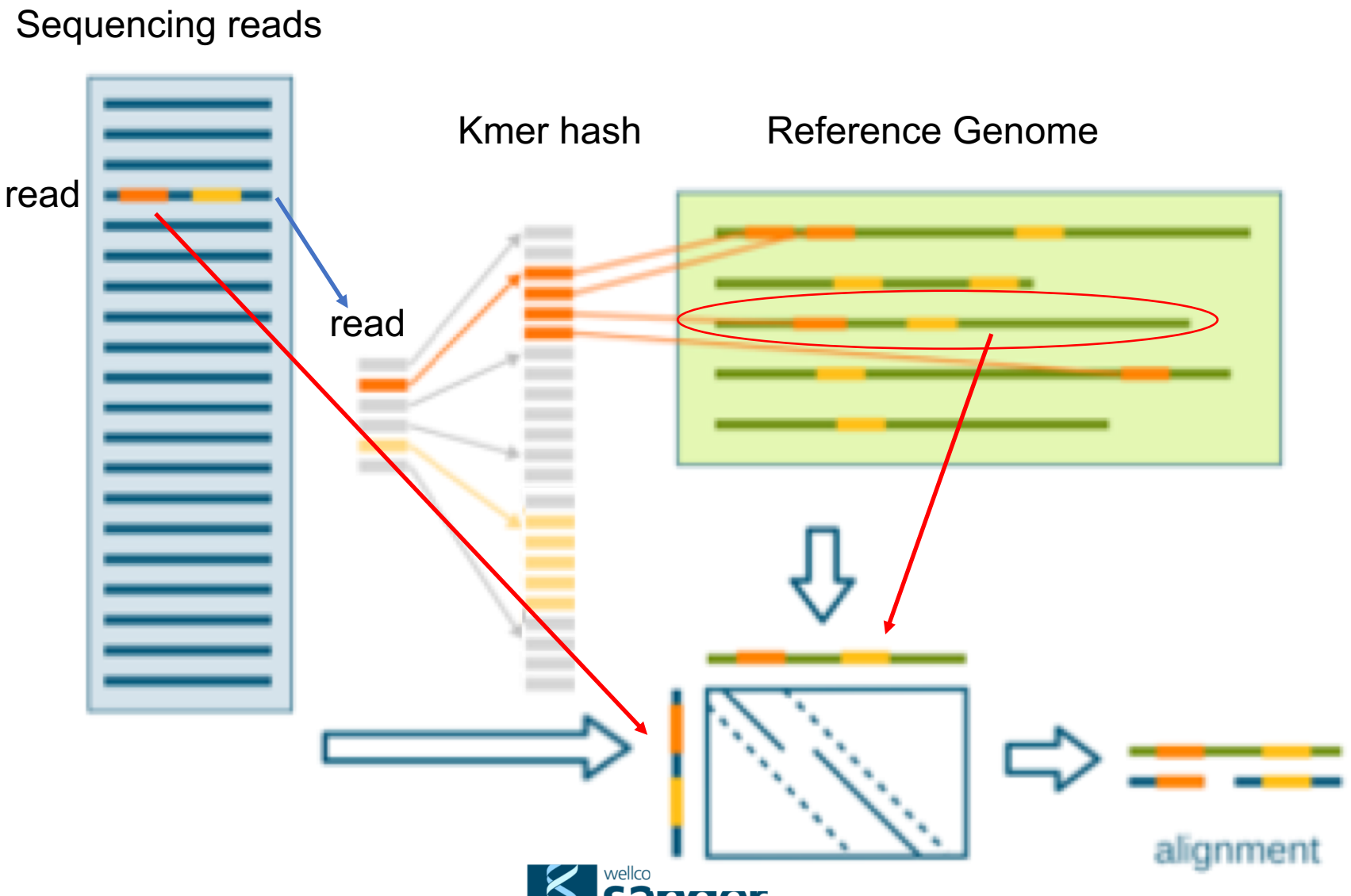
Hash of the reads: MAQ, ELAND, ZOOM and SHRiMP
Smaller but more variable memory requirements

Hash the reference: SOAP, BFAST and MOSAIK
Advantage: constant memory cost

Hash table alignment



Hash table alignment



Hash table alignment

Note: K-mer is a short fixed sequence of nucleotides

First thing: "wrap up" the reference genome to bring it to the reads – build a kmer hash
Scan the reference genome:
Build a profile (index) of all possible k-mers of length *n*
and the locations in the reference genome they occur:
Hash table will be several Gbytes in size for human genome

Next: For each sequence read:
Split the read into k-mers of length *n*
Lookup the locations in the reference via the index (**seed phase**)
Pick region on the genome with most k-mer hits
Perform Smith-Waterman alignment to fully align the read to the region
Output the alignment of each read onto the reference in BAM (or equivalent) format.

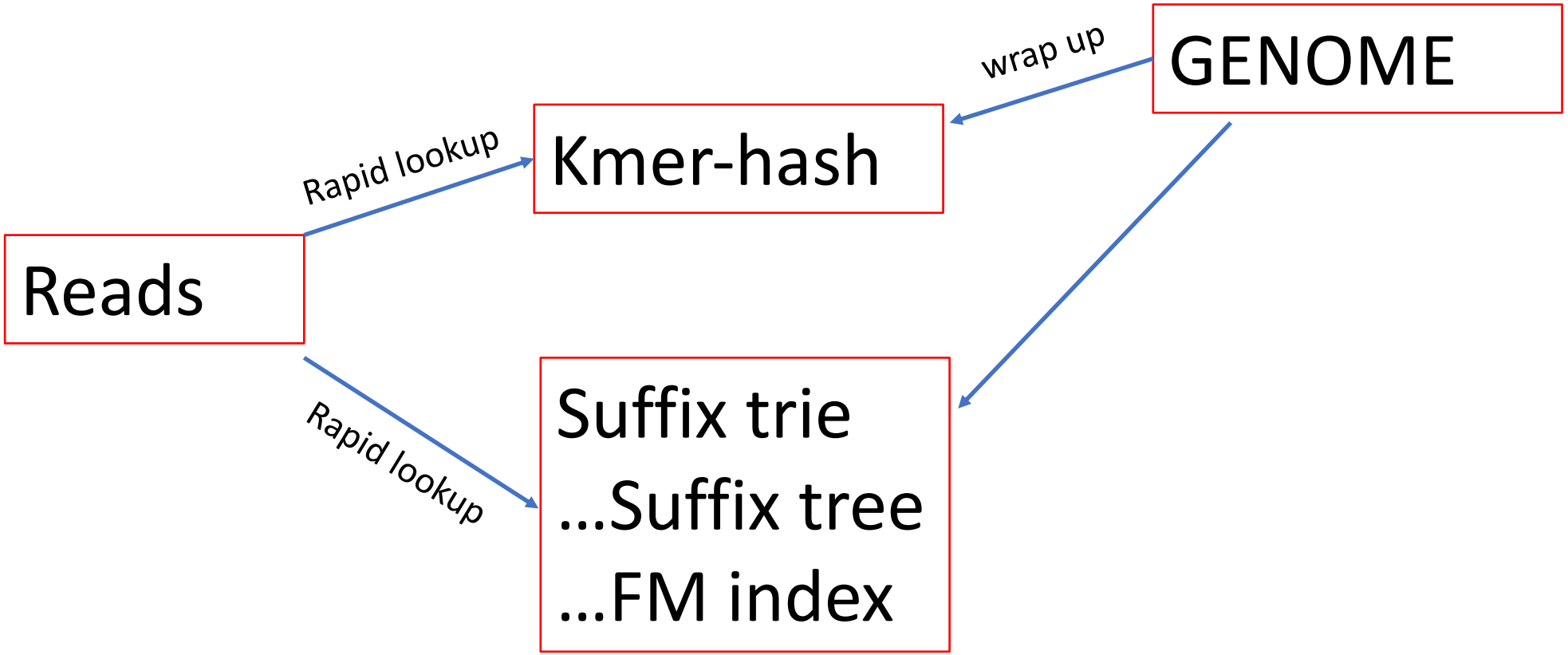
Hash of the reads: MAQ, ELAND, ZOOM and SHRiMP
Smaller but more variable memory requirements

Hash the reference: SOAP, BFAST and MOSAIK
Advantage: constant memory cost

Wrapping up the genome and bringing it to your reads

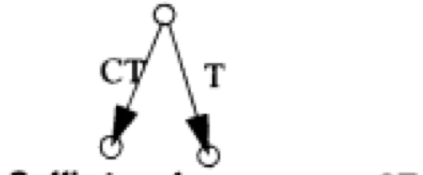
Kmer hash:

Summarises (folds up) content of genome and brings it close to the reads



Suffix/Prefix tree based aligners

For fast string matching: A suffix trie, or simply a trie, is a data structure that stores all the suffixes of a string, enabling fast string matching. To establish the link between a trie and an FM-index, a data structure based on Burrows-Wheeler Transform (BWT)



Suffix tree for sequence CT

Genome

Trie based on genome (slightly compacted)

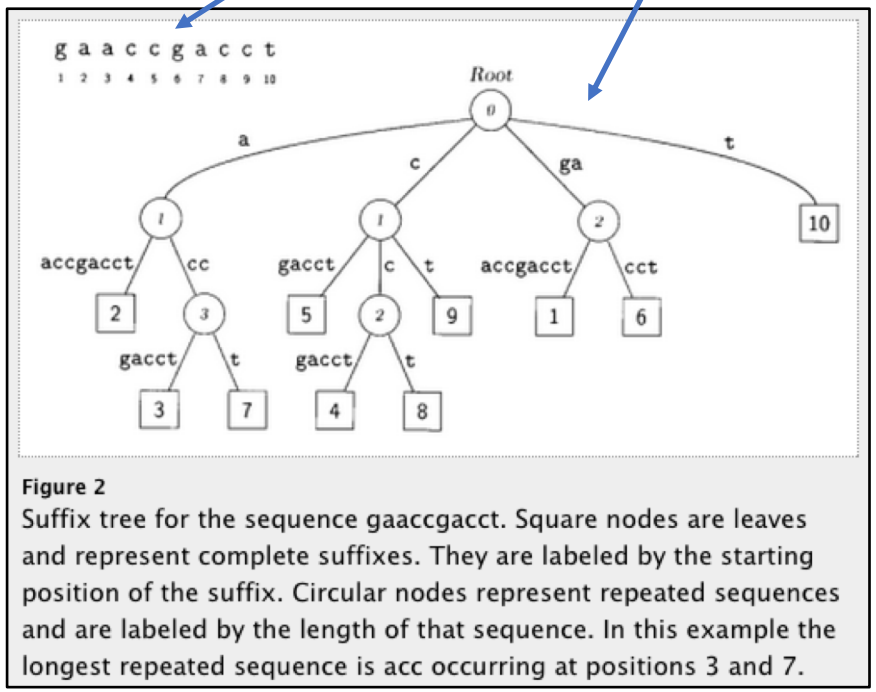


Figure 2
Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. In this example the longest repeated sequence is acc occurring at positions 3 and 7.

Delcher et al (1999) *NAR*

Build a suffix-trie for the genome:

- a different structure to bring the genome to the query
- But the memory requirements are huge

More terms: BW transform, FM-index …

- Methods of reducing memory & time footprint

Examples: **BWA**, bowtie2, MUMMer

wellcome trust sanger institute

Mapping qualities

What if there are several possible places in the genome to align your sequencing read?

Genomes contain many different types of repeated sequences

- Transposable elements (40-50% of vertebrate genomes)
- Low complexity sequence
- Reference errors and gaps
- Plenty of closely paralogous genes!

The aligner will issue a MAPPING QUALITY to each alignment

Mapping quality is a measure of how confident the aligner is that the read is corresponds to this location in the reference genome

Typically represented as a phred score (log scale)

Q0 = read placed with identical score elsewhere

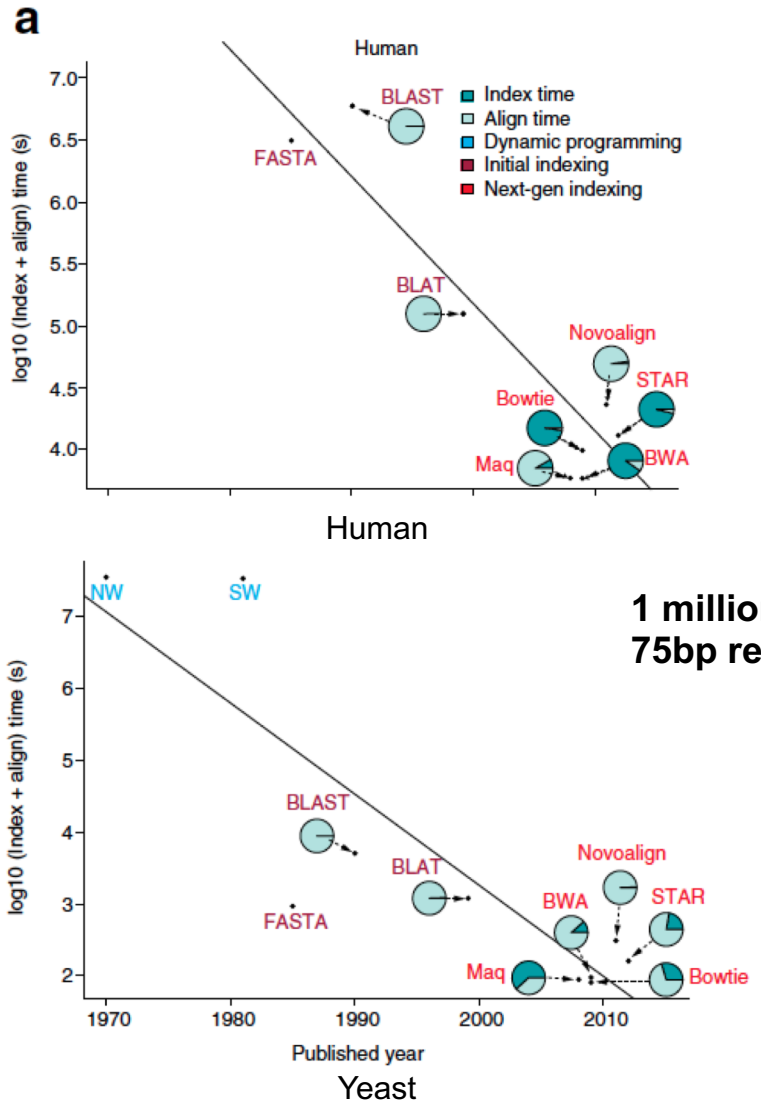
Q10 = 1 in 10 incorrect

Q20 = 1 in 100 incorrect

Paired-end sequencing is useful - if one end maps inside a repetitive elements and one outside in unique sequence: the the aligner will use this

- Hence prefer paired-end sequencing

Aligner choice – some considerations



- Align times are dropping exponentially by year (roughly Moore's law)
- Yeast is quicker than human.
- Dynamic programming (NW) is slow cf hashing (BLAT) is slow compared to suffix trie (BWA)

wellcome trust sanger institute

Muir et al. Genome Biology (2016)
DOI 10.1186/s13059-016-0917-0

Aligner choice – some considerations

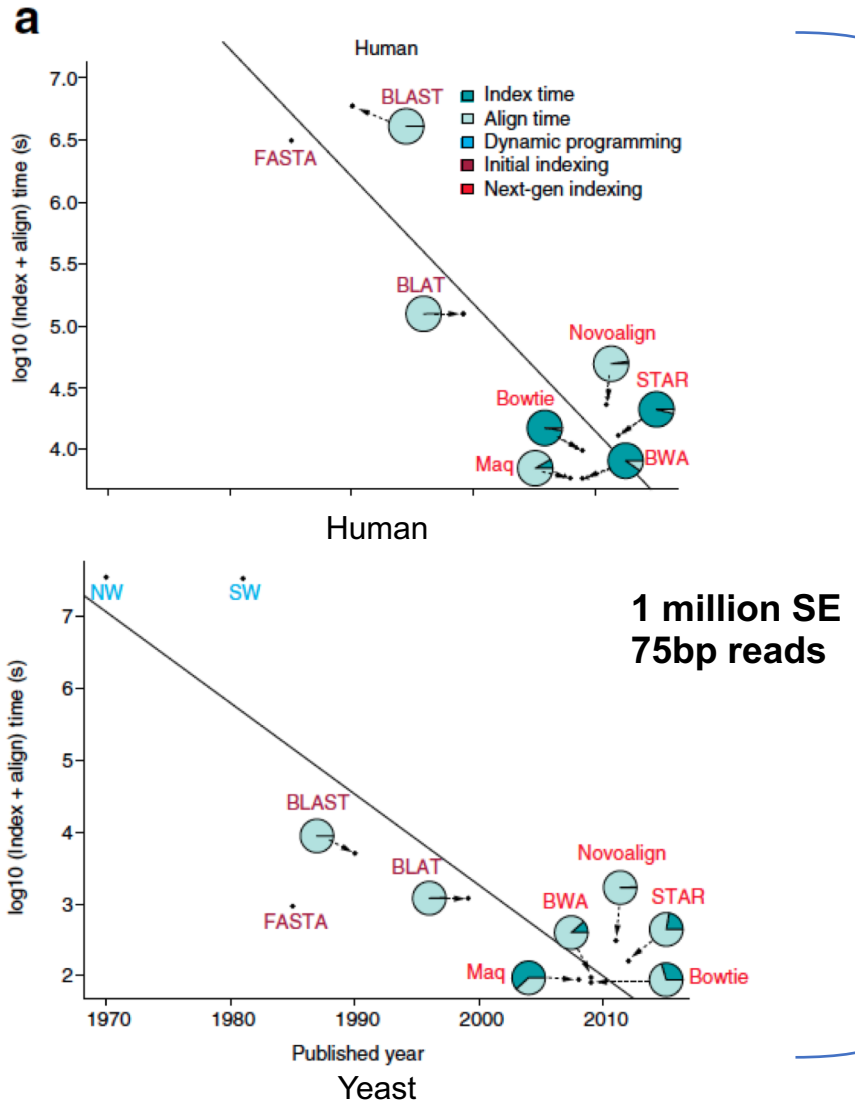
In the event that you have a choice of aligner to use

Read manual, publication, vignette, etc

Sometimes, defaults may not be appropriate for your study

Does an aligner/mapper support:

- my platform (Illumina, PacBio etc)?
- my sequence type (DNA, RNA, etc)?
- paired end reads?
- my read lengths (e.g. PACBio are long 10's of thousands)
- Consider speed, and memory requirements and your resources
- Does it allow for mismatches (SNPs)? (*Configurable)*
- Support for gapped alignments? *(Yes)*
- How does it handle multi-mapping reads? (*Configurable)*
- Does it allow spliced alignment (RNA seq)? *(Purpose build aligners for this)*



- Align times are dropping exponentially by year (roughly Moore's law)
- Yeast is quicker than human.
- Dynamic programming (NW) is slow cf hashing (BLAT) is slow compared to suffix trie (BWA)

wellcome trust sanger institute

Muir et al. Genome Biology (2016)
DOI 10.1186/s13059-016-0917-0

Some alignment options (there are lots more)

careful

Unspliced: No intron-sized gaps

Spliced: Allow for big gaps inside reads, **are aware of gene-structure**

Aligner	Gapped alignment	SE and/or PE	Input format	Output format	Trimming?	BS-seq	Note
BWA-**MEM**	Yes	SE, PE	FASTQ FASTA	SAM	No	No	Seed + extend: Local alignment
~~Bowtie~~	~~No~~	~~SE, PE~~	~~FASTQ FASTA~~	~~SAM~~	~~Yes~~	~~No~~	~~Mismatches <=3~~
Bowtie2	Yes	SE, PE	FASTQ FASTA qseq	SAM	Yes	No	Seed + extend: Local alignment
TopHat	Yes	SE, PE	FASTQ FASTA qseq	SAM	Yes	No	Uses bowtie or bowtie2 as base aligner
STAR	Yes	SE, PE	FASTQ FASTA	SAM	Yes	No	Fastest and most accurate for RNAseq

Don't use old tech

Spliced aligns (RNASeq)

Independently performs spliced aligns (no preprocessing)

Alignments – scaling up

Question: IF: 50Mbp bwa mem = 5 CPU minutes.
AND: 150 Gbp per HiSeqX10 lane – 1 human genome sequenced at 45x

THEN: How long will it take to align your X10 human genome sequencing?

Alignments – scaling up

Question: IF: 50Mbp bwa mem = 5 CPU minutes.
AND: 150 Gbp per HiSeqX10 lane – 1 human genome sequenced at 45x

THEN: How long will it take to align your X10 human genome sequencing?

How many lots of 50Mb in 150Gb ?

150Gb / 50Mb = (150 x 10^9) / (50 x 10^6) = 3 x 10^3 lots

Each lot takes 5 minutes => total time is 5 x 3 x 10^3 CPU minutes

15000 CPU minutes = 250 hours ~ 10 days ***ouch.***

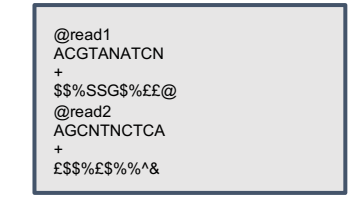
Alignments – scaling up

This is why we like compute clusters!

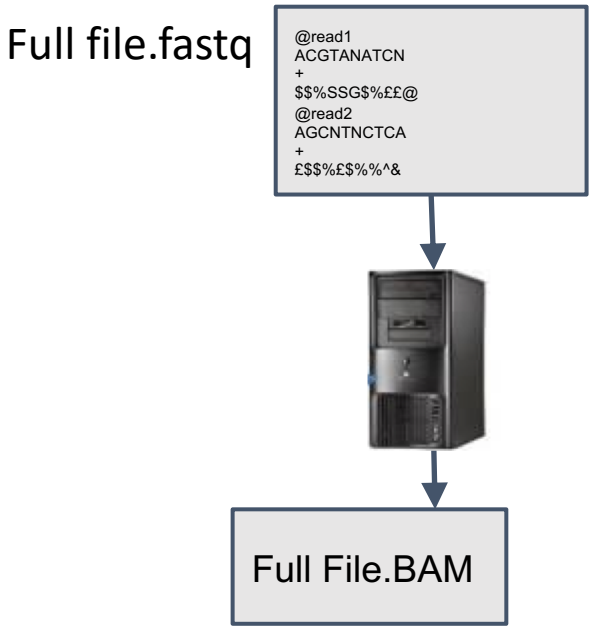
AS ALWAYS: The strategy is to:

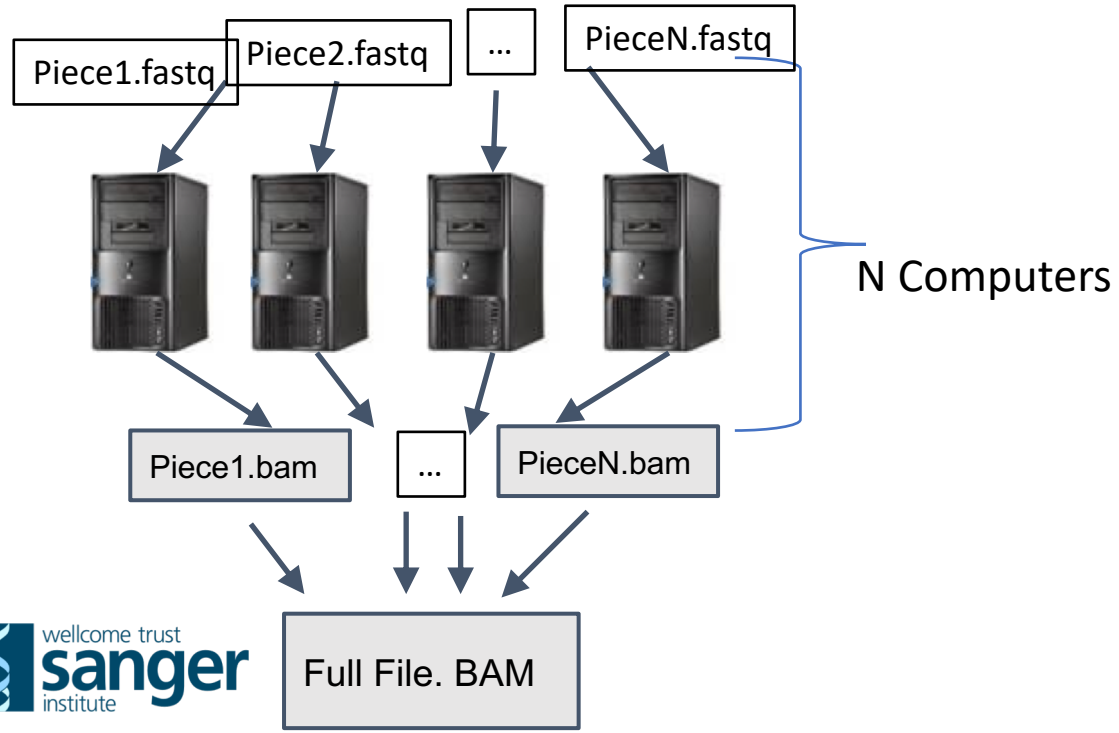
- Chop up the problem into small pieces and
- Solve each piece in parallel, with N computers working **at same time**
- Put the pieces back together after.

RESULT: Time of problem is reduced by N.



Full file.fastq





wellcome trust sanger institute

Alignments – scaling up

Question: IF: 50Mb bwa mem = 5 CPU minutes.
AND: 150 Gbp per HiSeqX10 lane – 1 human genome sequenced at 45x
THEN: How long will it take to align your human genome sequencing?

How many lots of 50Mb in 100Gb ?
150Gb / 50Mb = (150 x 10^9) / (50 x 10^6) = 3 x 10^3

Each lot takes 5 minutes => total time is 5 x 3 x 10^3 CPU minutes
15000 CPU minutes = 250 hours ~ 10 days ***ouch.***

Compute Cluster with N nodes: Time of problem is reduced by N.

E.g. if you have dedicated 200 notes, the time for alignment drops to ***~ 1.25 hrs***
- Which is ok for a single sample.
- If you have 1000 samples, you need more nodes + better optimisation.
- At WTSI, alignment is more much more efficient (dedicated resources)

wellcome trust sanger institute

Overview

Intro

Methods / Aligners

Alignment Outputs

Alignment Viewers

NGS Workflows, QC and BAM Improvement

Pre-alignment file formats

Data generation

FASTQ

Read ID

Read sequence

Read ID

Qualities

```
@IL16_4408:3:5:17860:13258
CTGGCTCACATACAGGCCAGTATAAAGCGTCTCCTTTAAA
+
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHE
@IL16_4408:3:14:13276:1210
GTAGAAAACAATATAGAAGCCTTAGGACAGAAAACCCTAA
+
HHHHHHHHHHHHHHHHHDHHHHHHHHBGHHHHHCDCCF>
@IL16_4408:3:83:5210:21157
AGGCAGCAGGAACTCCAGTTAGGCCATAGTTCATCCTTGC
```

The same 4 line format for paired reads (PE sequencing)

```
@IL16_4408:3:5:17860:13258
CAGTTTTTTCAGAGCAGTAGCCATTAGGCACAATGTGATT
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAFFBFFFF
@IL16_4408:3:14:13276:1210
AGTCAACAGATGTCCTTTGAGCTTAAGAATTCAGCAGAAG
+
FFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFDCF
@IL16_4408:3:83:5210:21157
GGGCCAGTGTGTCCCCTCCTGCCACTGAAGACCATGCTAT
```

wellcome trust sanger institute

Post-alignment file formats

SAM (Sequence Alignment/Map) format

- Single unified format for storing read alignments to a reference genome
- Developed by the 1000 Genomes group in 2009

BAM (Binary Alignment/Map) format

- Binary equivalent of SAM
- Developed for fast processing/indexing
  - Block GZIP compression for random access of regions

Key features

- Can store alignments from most aligners
- Supports multiple sequencing technologies
- Supports indexing for quick retrieval/viewing
- Compact size (e.g. 112Gbp Illumina = 116Gbytes disk space)
- Reads can be grouped into logical groups e.g. lanes, libraries, samples
- Widely support by variant calling software packages

Specification maintained by the Global Alliance for Genomics and Health

Global Alliance
for Genomics & Health

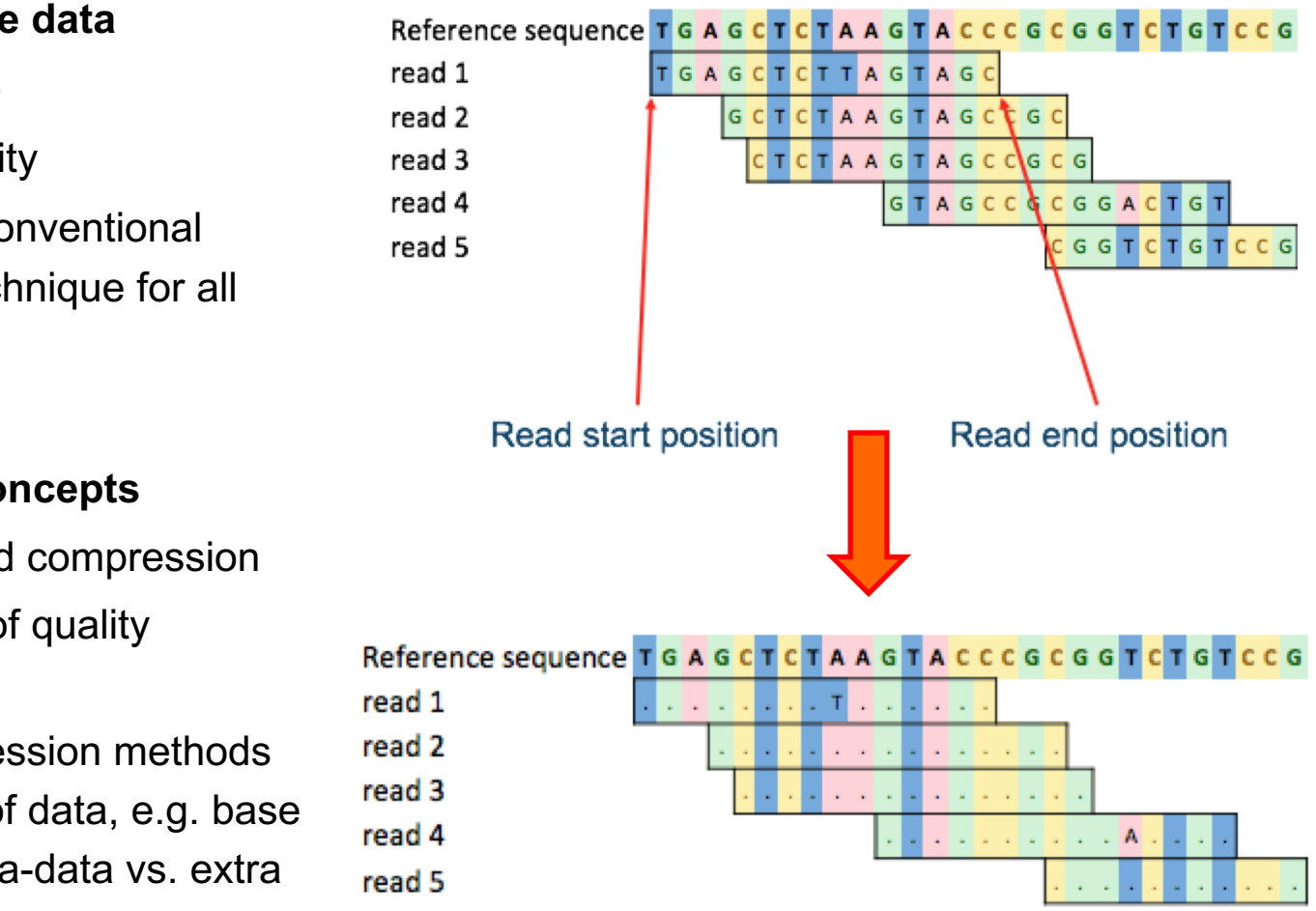
Reference based compression

BAM stores all of the data

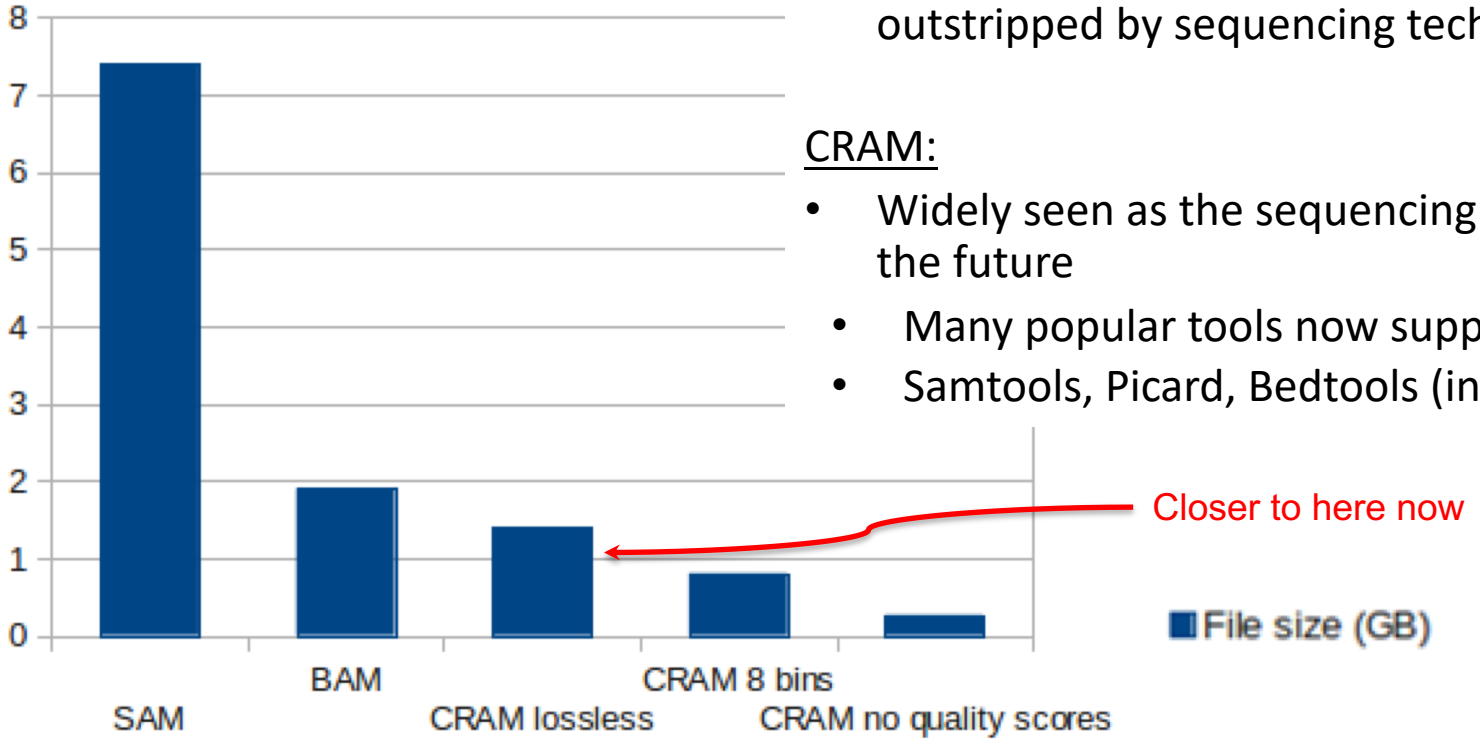
- Every read base
- Every base quality
- Using a single conventional compression technique for all types of data

CRAM: Important concepts

- Reference based compression
- Controlled loss of quality information
- Different compression methods to suit the type of data, e.g. base qualities vs. meta-data vs. extra tags



CRAM format



- BAM files are too large
  - ~1.5-2 bytes per base pair
- Increases in disk capacity are being far outstripped by sequencing technologies

CRAM:

- Widely seen as the sequencing format of the future
  - Many popular tools now support CRAM
  - Samtools, Picard, Bedtools (in the future)

wellcome trust sanger institute

SAM/BAM format

SAM fields

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,254}	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

CIGAR string

Cigar has been traditionally used as a compact way to represent a sequence alignment: to go “FROM one string TO another you perform these operations”.

Operations include

M - match or mismatch

I - insertion

D - deletion

S - soft clip (ignore these bases)

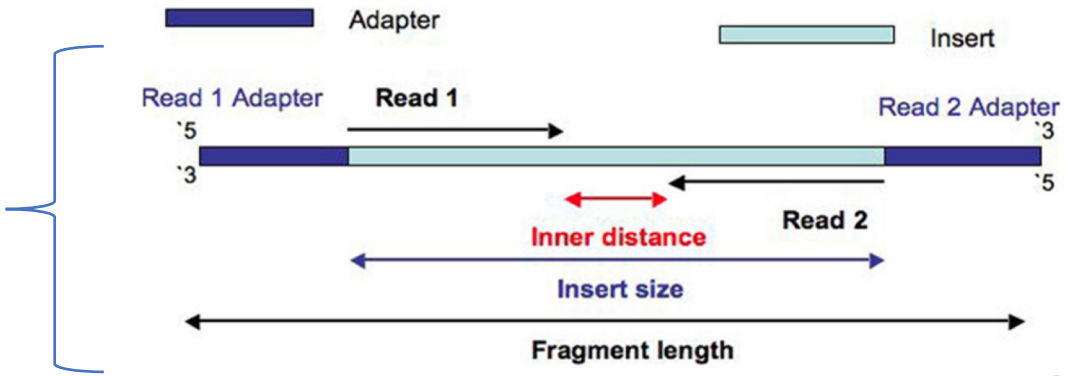
H - hard clip (ignore and remove these bases)

```
E.g.Read:     ACGCA-TGCAGTtagacgt
     Ref:     ACTCAGTG--GT
     Cigar:   5M 1D 2M 2I 2M 7S
```

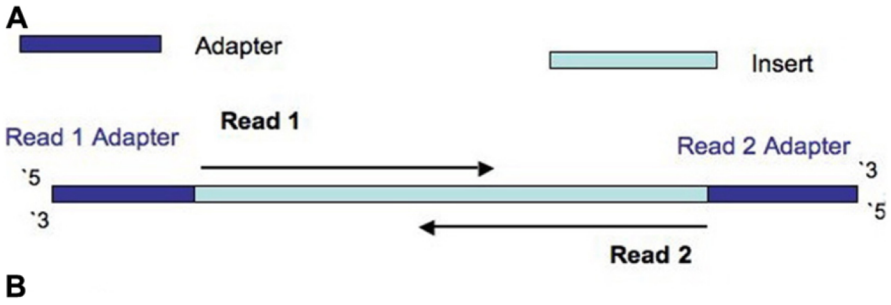
INTERLUDE – PAIRED END SEQUENCING PARAMETERS

Fragment length, insert size and inner mate distance

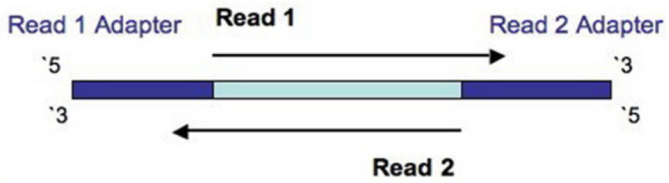
Insert size > length(read 1 + read 2)



Insert size < length(read 1 + read 2)



Insert size < length(read 1);
Insert size < length(read 2)



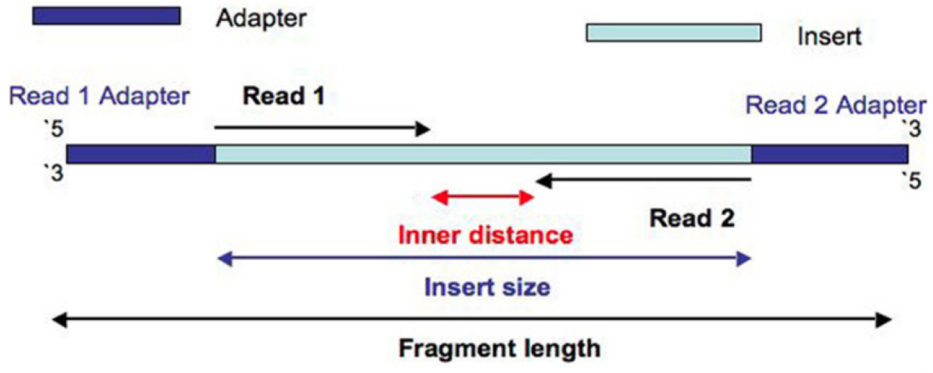
Turner, 2014. PMID:24523726

wellcome trust sanger institute

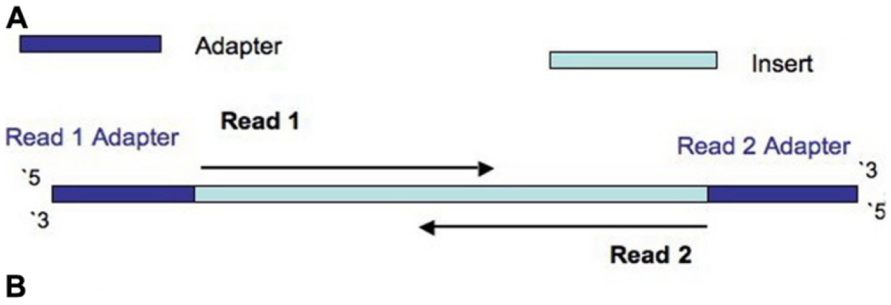
Fragment length, insert size and inner mate distance

Insert size > length(read 1 + read 2)

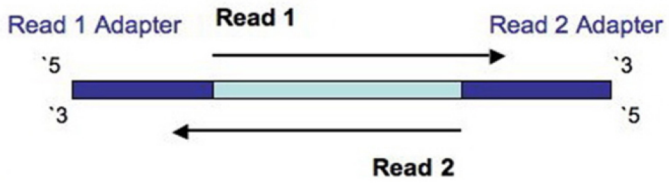
Both reads get the same "name"



Insert size < length(read 1 + read 2)



Insert size < length(read 1);
Insert size < length(read 2)



Turner, 2014. PMID:24523726

wellcome trust sanger institute

NOW WE RESUME OUR NORMAL PROGRAMMING

wellcome trust sanger institute

SAM/BAM/CRAM format in practice

```
HX8_24050:5:2103:20770:9449  83  7 4189207 0 151M   = 4188954 -404 GTTG <JJJ MC:Z:151M MD:Z:151 NM:i:0 AS:i:151 XS:i:151
…
HX8_24050:5:2103:20770:9449  163 7 4188954 0 151M   = 4189207 404  TTGG AAFF MC:Z:151M MD:Z:151 NM:i:0 AS:i:151 XS:i:151

HX8_24050:3:2212:32319:38772 99  7 4188947 0 151M   = 4189300 436  ATGG AAAF MC:Z:68S83M MD:Z:151 NM:i:0 AS:i:151 XS:i:151
…
HX8_24050:3:2212:32319:38772 147 7 4189300 0 68S83M = 4188947 -436 ACAG --7- MC:Z:151M MD:Z:5C10T4T61 NM:i:3 AS:i:68 XS:i:75
```

QNAME: read id
- matches fastq

FLAG:
> samtools flags 83: **PAIRED, PROPER_PAIR, REVERSE, READ1**
> samtools flags 163: **PAIRED, PROPER_PAIR, MREVERSE, READ2**

RNAME, POS – chromosome & base pair position

MAPQ mapping quality
CIGAR M? S? I ? D ?

RNEXT, PNEXT - paired read chr & position
RNEXT "=" means same as first read
PNEXT: read2 aligns 5' of read1.
TLEN: insert size

SEQuence
QUALity (alignment doesn't use them, but variant callers will)

ALIGNMENT TAGS:
http://www.samformat.info/sam-format-alignment-tags

```
MC:mate cigar
MD:string for mismatching positions
NM:Edit distance to the reference (no clipping)
AS:Aligner's Alignment score
XS:suboptimal alignment score (aha!!)
```

SAM/BAM/CRAM format in practice

```
HX8_24050:5:1217:22455 83 7 87483848 60 118M1I32M = 87483669 -329 TG.. FJ.. MC:Z:151M MD:Z:150 NM:i:1 AS:i:143 XS:i:0
```

QNAME:
FLAG
RNAME
POS
MAPQ
CIGAR
RNEXT, PNEXT
RNEXT
PNEXT
TLEN: can we work this out from the POS and PNEXT?
SEQ:
QUAL:
MC: how does the mate match?
MD: Does the mate have a mismatch?
NM: This read: what is the edit distance?
AS: what the current alignment score?
XS: what is the align score of the next-best-hit?

Overview

Intro

Methods / Aligners

Alignment Outputs

Alignment Viewers

NGS Workflows, QC and BAM Improvement

Three viewers, in order of sophistication

Context:

You have a mouse.
You extracted DNA from mouse kidney and performed paired-end Whole-Genome Sequencing.
Core sequencing facility (and/or an informatician and / or you) produces the alignments and summary stats.

They do variant calling! You find some variants you like, or some differentially expressed genes.
YOU HAVE TO LOOK AT YOUR ALIGNMENTS.

You will need:
Genome file (ie the reference)
Genome file: GRCm38.fa. Remember, there's always a reference back there …
Gene annotations.
Alignment file: **MySample.bam**

Moral

… You need to have a look at it. You cannot just trust a spreadsheet of variants given to you by someone else.

Viewer 1: samtools

You could just use samtools **view** at the command line.

- This shows you the bam header: chromosomes/contigs, read-groups, programs which have acted
- This shows you each read id and CIGAR string, as well as mate structure (as we saw before)

$ samtools view –H MySample.bam

```
@HD    VN:1.5  SO:coordinate
@SQ    SN:1    LN:195471971    AS:GRCm38
@SQ    SN:10   LN:130694993   AS:GRCm38
…
@SQ    SN:X    LN:171031299    AS:GRCm38
@SQ    SN:Y    LN:91744698     AS:GRCm38
…
@RG    ID:332872      PL:ILLUMINA    SM:MD5638a  DS:WGS_ILLUMINA_short   PU:24050_1
@RG    ID:332873      PL:ILLUMINA    SM:MD5638a  DS:WGS_ILLUMINA_short   PU:24050_2
…
@PG    ID:basecalling_0      PN:Unknown      PP:SCS_0      DS:Basecalling Package  VN:Unknown
@PG    ID:bambi_0     PN:bambi       CL:/software/solexa/pkg/bambi/0.9.11/bin/bambi i2b ….
@PG    ID:bamadapterfind_0   …
@PG    ID:bwa_0       PN:bwa  CL:/software/solexa/pkg/bwa/ ...

HX8_24050:4:2105:31456:16147   99     1      3000000 40     20M1I104M26S    =      3000301 45 …
```

Viewer 1: samtools

You could just use samtools at the command line.

- This shows you the bam header.
- This shows you each read id and CIGAR string, as well as mate structure.

- But otherwise it's a pretty poor way of visualizing anything.

- No feel for relationship of pairs or how multiple reads show the same signal

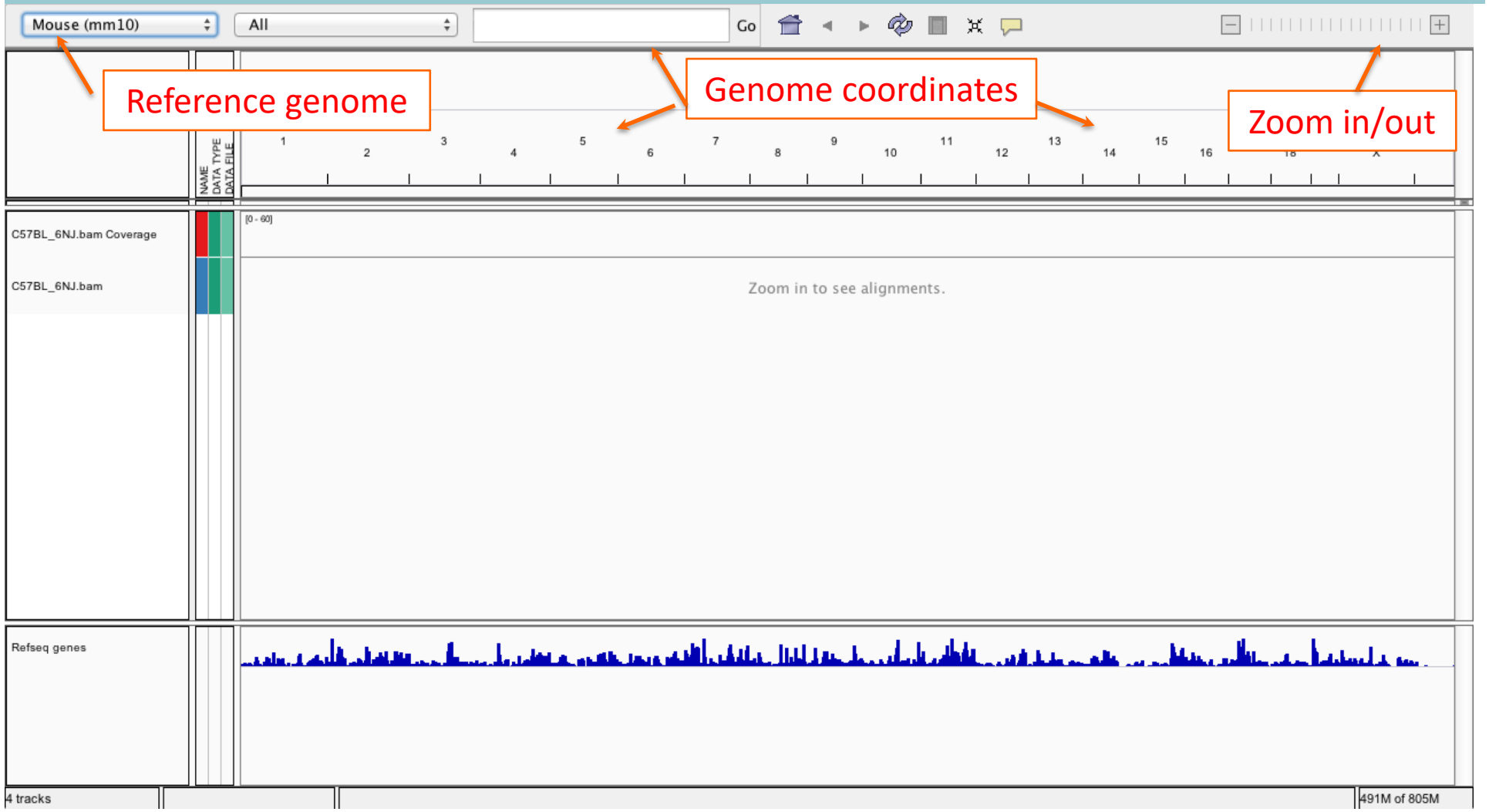
The whole point of short-read sequencing is to infer variation etc from PILEUP – the cooperative action of lots of reads together, all having the same variation at the same point
Samtools doesn't cut it.

```
HX8_24050:5:2103:20770:9449  83  7 4189207 0 151M    = 4188954 -404 GTTG <JJJ MC:Z:151M MD:Z:151 NM:i:0 AS:i:151 XS:i:151
…
HX8_24050:5:2103:20770:9449  163 7 4188954 0 151M    = 4189207 404  TTGG AAFF MC:Z:151M MD:Z:151 NM:i:0 AS:i:151 XS:i:151

HX8_24050:3:2212:32319:38772 99  7 4188947 0 151M    = 4189300 436  ATGG AAAF MC:Z:68S83M MD:Z:151 NM:i:0 AS:i:151 XS:i:151
…
HX8_24050:3:2212:32319:38772 147 7 4189300 0 68S83M = 4188947 -436 ACAG --7- MC:Z:151M MD:Z:5C10T4T61 NM:i:3 AS:i:68 XS:i:75
```

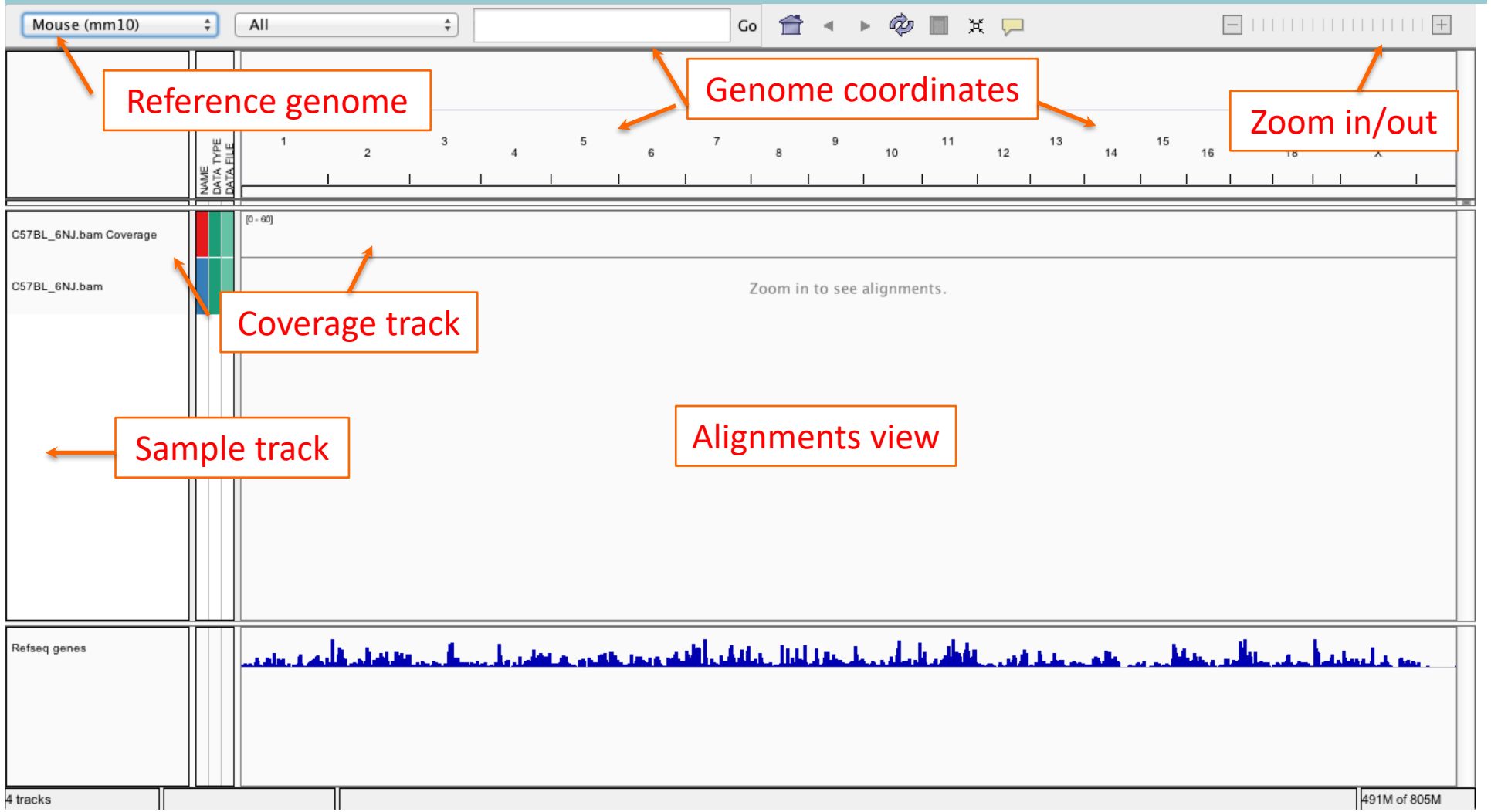
wellcome trust sanger institute

Option 3: Integrative Genomics Viewer (IGV)



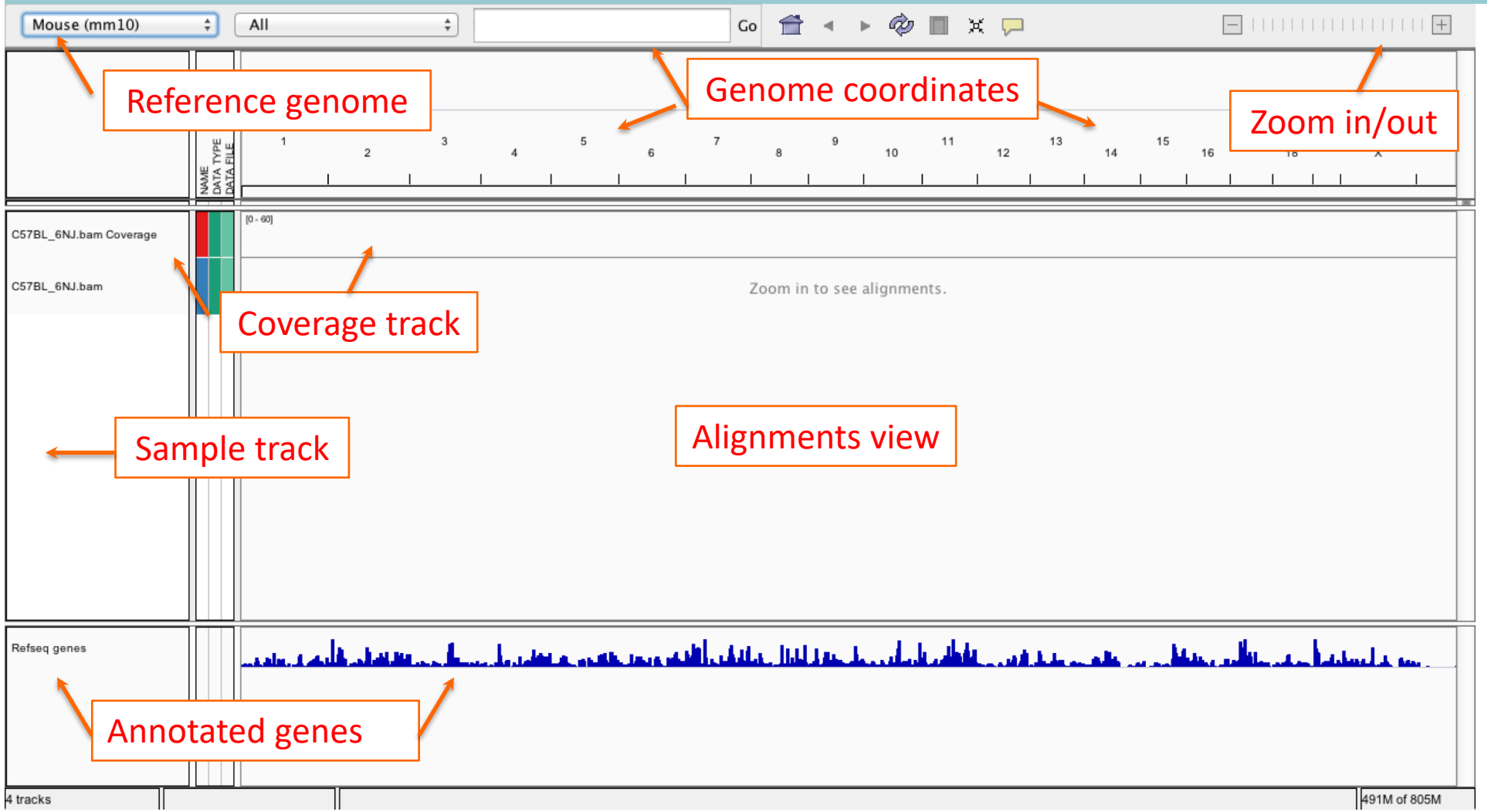
wellcome trust sanger institute

Integrative Genomics Viewer (IGV)



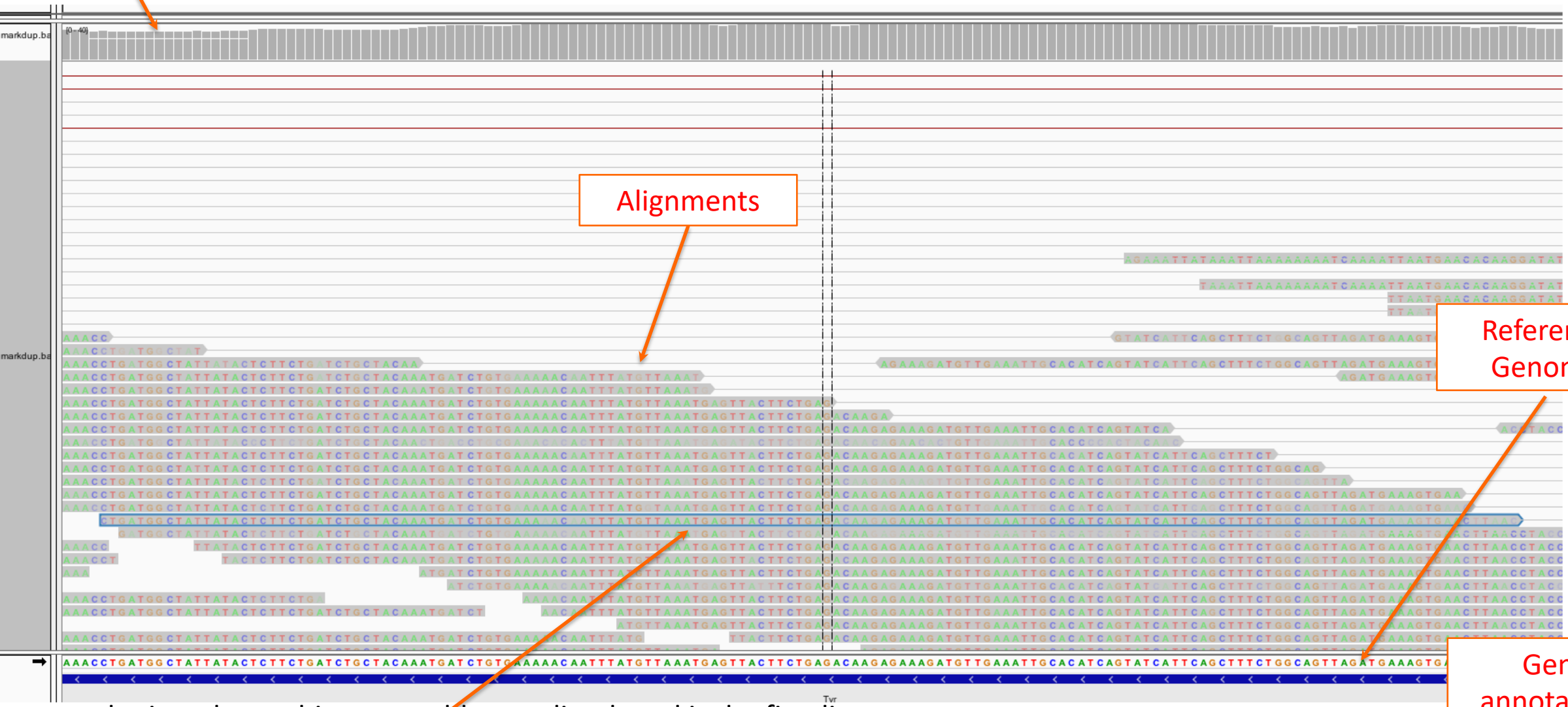
wellcome trust sanger institute

Integrative Genomics Viewer (IGV)



wellcome trust sanger institute

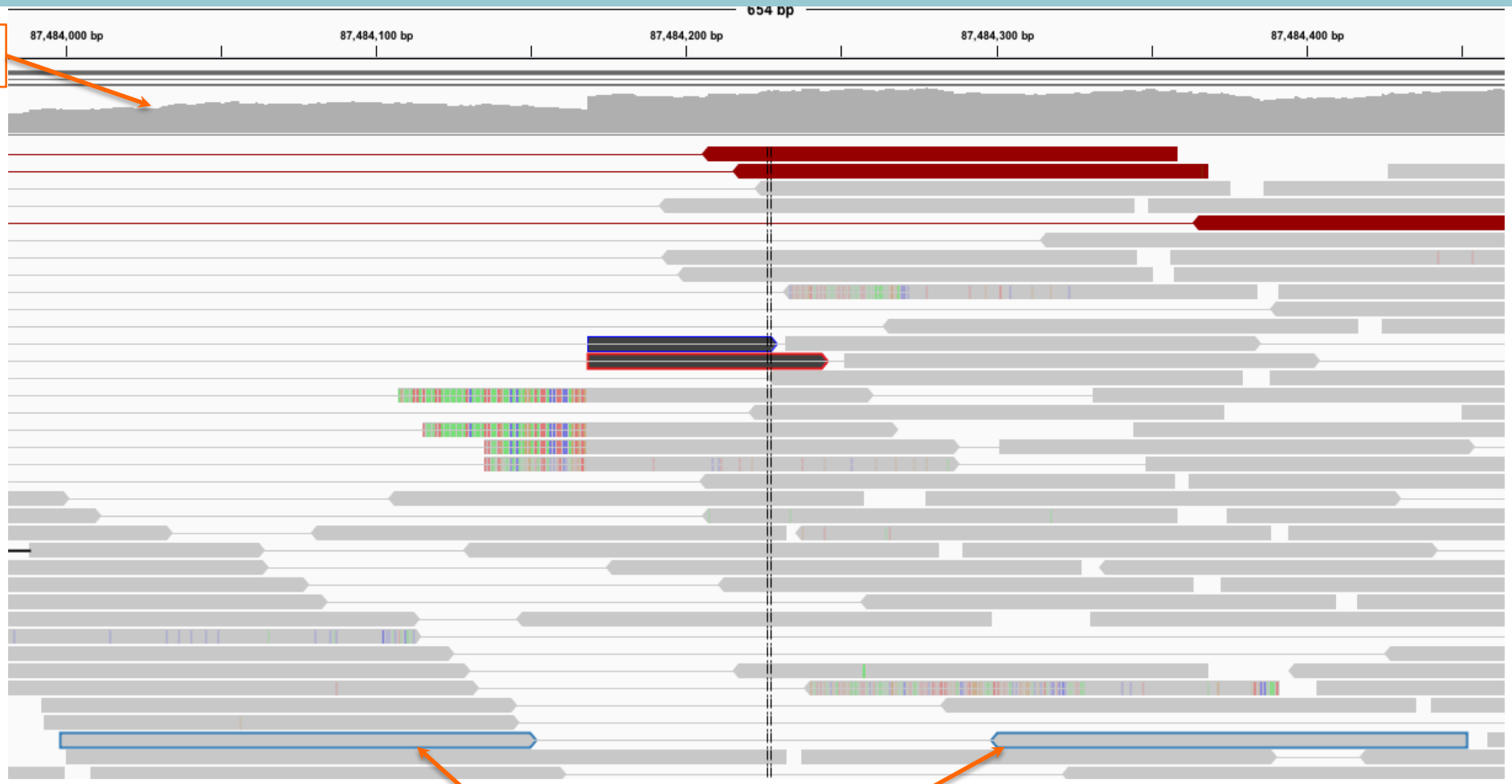
Integrative Genomics Viewer (IGV)



samtools view shows this output: blue-outlined read is the first line

```
HX8_24050:3:2122:30411:65986    163    7    87483999    60    151M    =    87484301    453    CTGATGGCTA
HX8_24050:3:2122:30411:65986    83     7    87484301    60    151M    =    87483999    -453   TGTCACTAGT
```

Integrative Genomics Viewer (IGV)

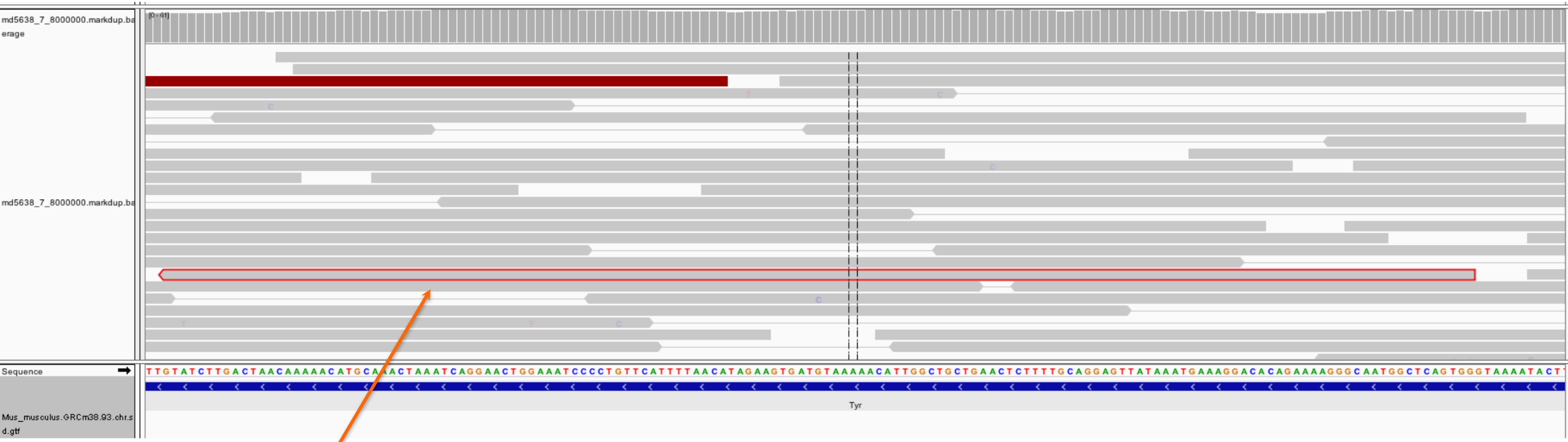


Coverage track

samtools view shows this output: blue-outlined reads are a pair

```
HX8_24050:3:2122:30411:65986    163    7    87483999    60    151M    =    87484301    453     CTGATGGCTA
HX8_24050:3:2122:30411:65986    83     7    87484301    60    151M    =    87483999    -453    TGTCACTAGT
```

Integrative Genomics Viewer (IGV)

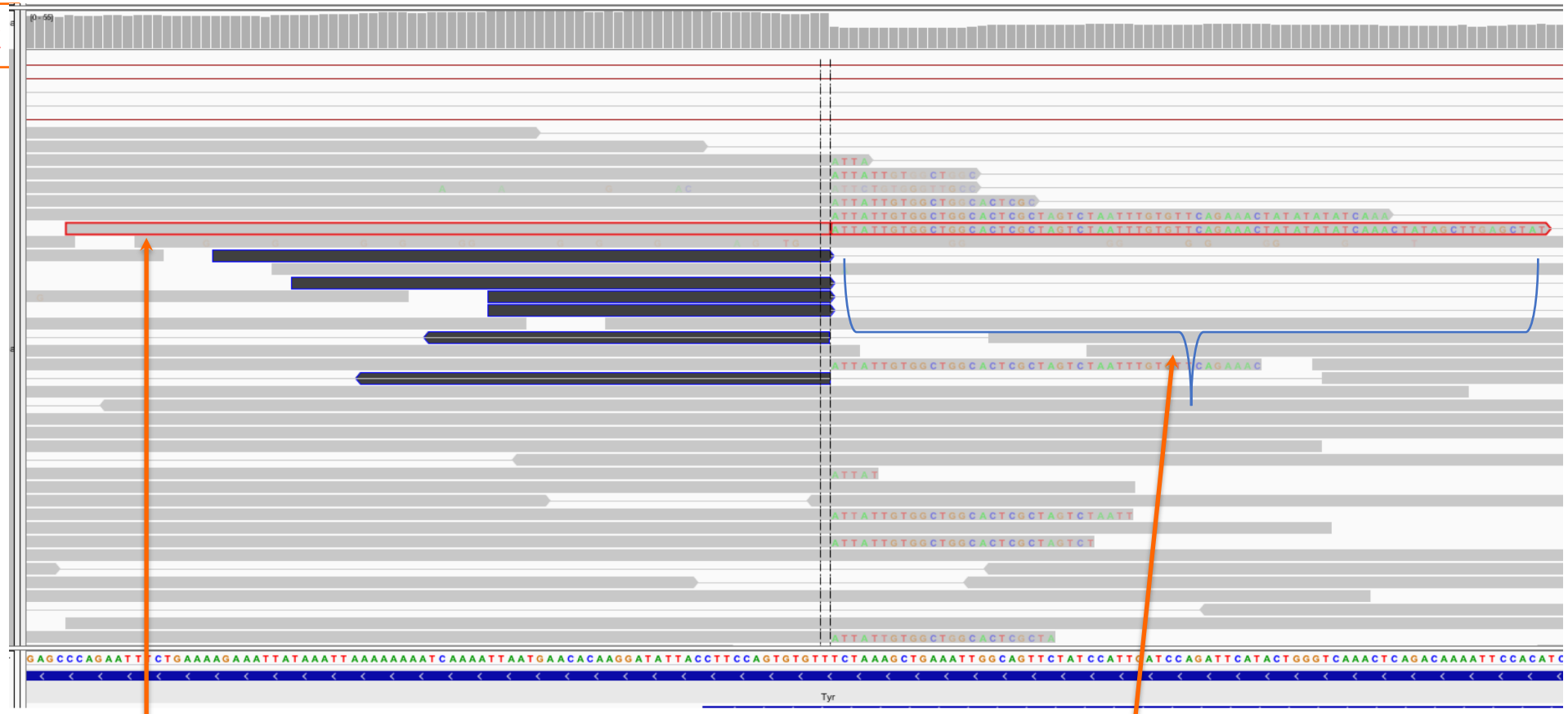


samtools view shows this output: red-outlined read is the 3’ read

```
HX8_24050:1:2112:29315:29173    147    7    87484452    60    151M    =    87483756    -847    GTAT
```

Integrative Genomics Viewer (IGV)

Coverage track

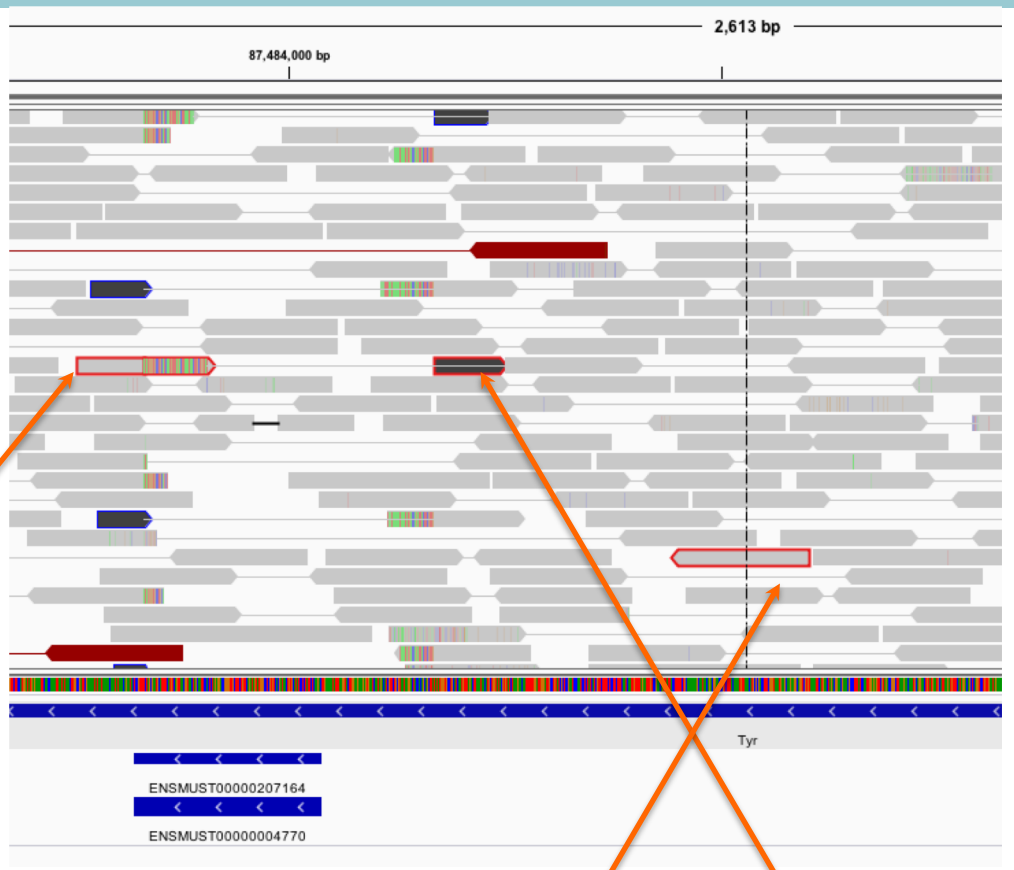


samtools view shows this output: red-outlined read is the 5’ read

```
HX8_24050:1:2112:29315:29173    99    7    87483756    60    78M73S    =    87484452    847    CCAGA
```

Notice the CIGAR string is 78M 73S – softclipped means that the alignment STOPs at bp 78, but the rest of the read is NOT trimmed. Result is that EVERY BASE PAIR of the read will mismatch ref => bases are coloured!

Integrative Genomics Viewer (IGV)



But wait! There's more

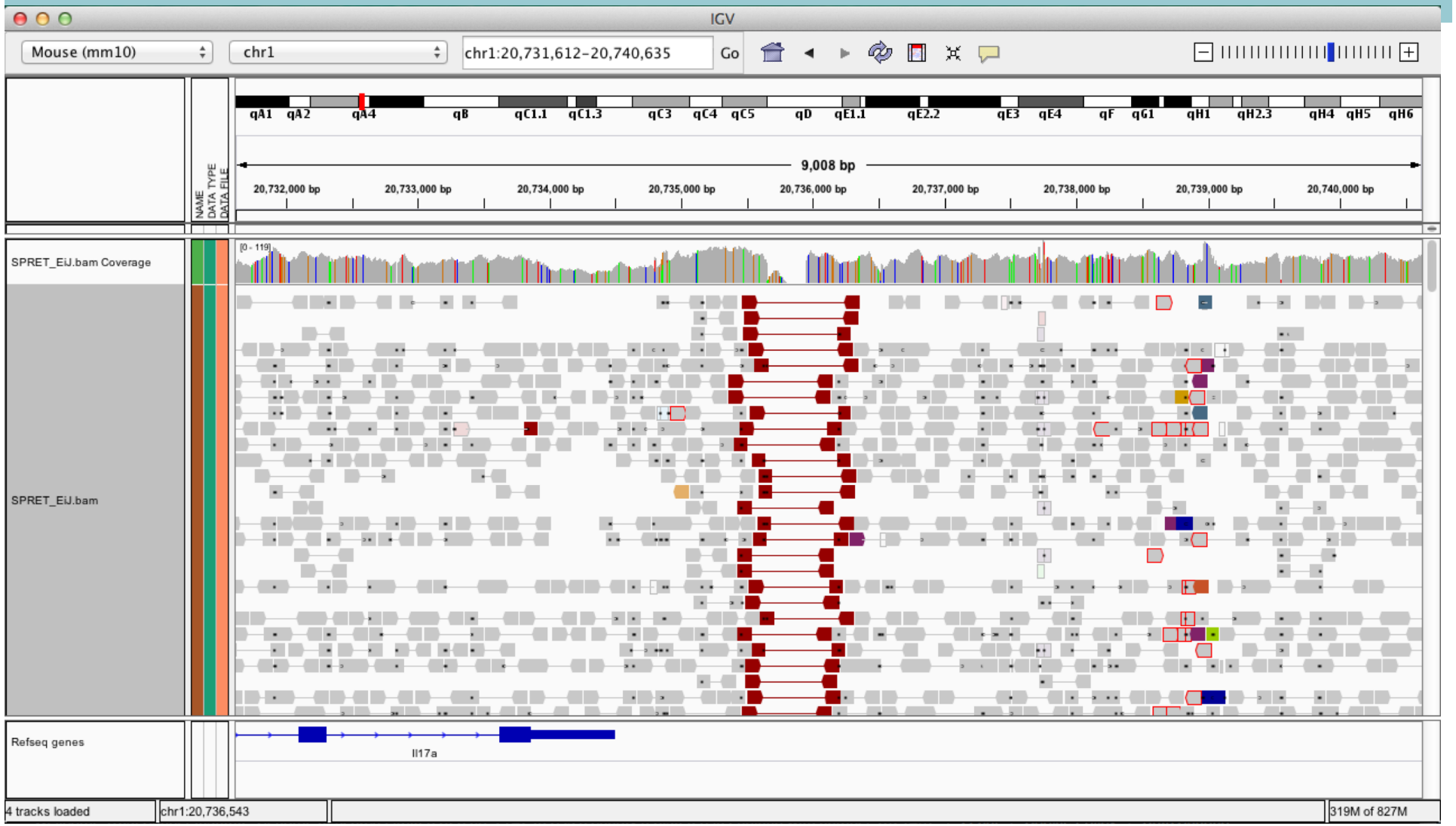
samtools view will actually show three lines in the file with the same read name.

We have examined the left read (softclipped) and the right read (perfectly aligned).

The middle read alignment is the 'right half' of the left read. It is hard-clipped and marked as 'secondary' – see flags

```
HX8_24050:1:2112:29315:29173    99     7    87483756    60    78M73S    =    87484452    847     CCAGA
HX8_24050:1:2112:29315:29173    2147   7    87484169    60    76H75M    =    87484452    434     TTATT
HX8_24050:1:2112:29315:29173    147    7    87484452    60    151M      =    87483756    -847    GTAT
```

Viewing reads as pairs



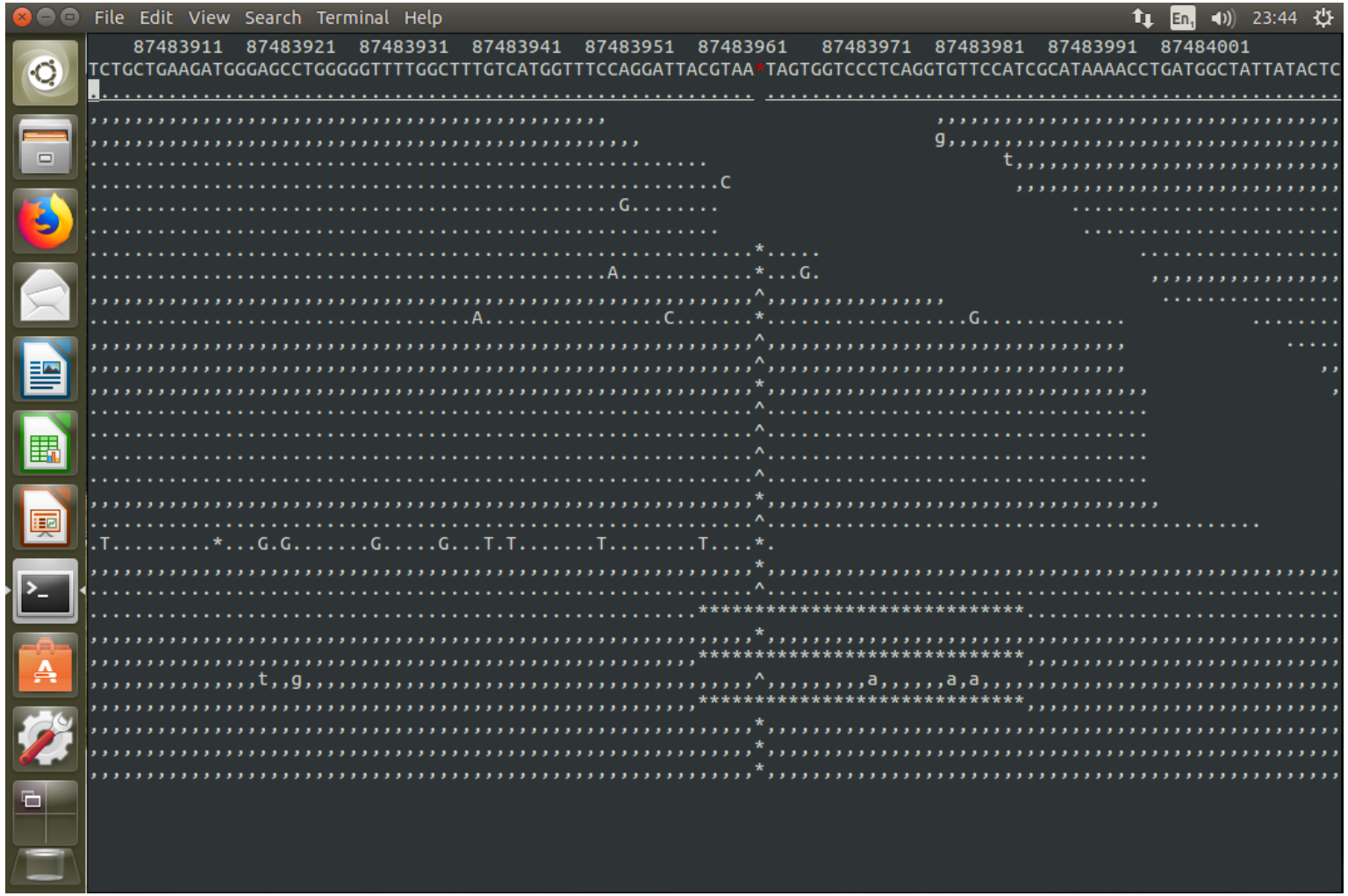
wellcome trust sanger institute

Two samples

wellcome trust sanger institute

Option 2: samtools tview

If you were stuck on a desert island, this would do the job: and you can try it for fun during the practical!
samtools tview md5638.markdup.bam ../../ref/GRCm38.68.dna.toplevel.fa



wellcome trust sanger institute

Overview

Intro

Methods / Aligners

Alignment Outputs

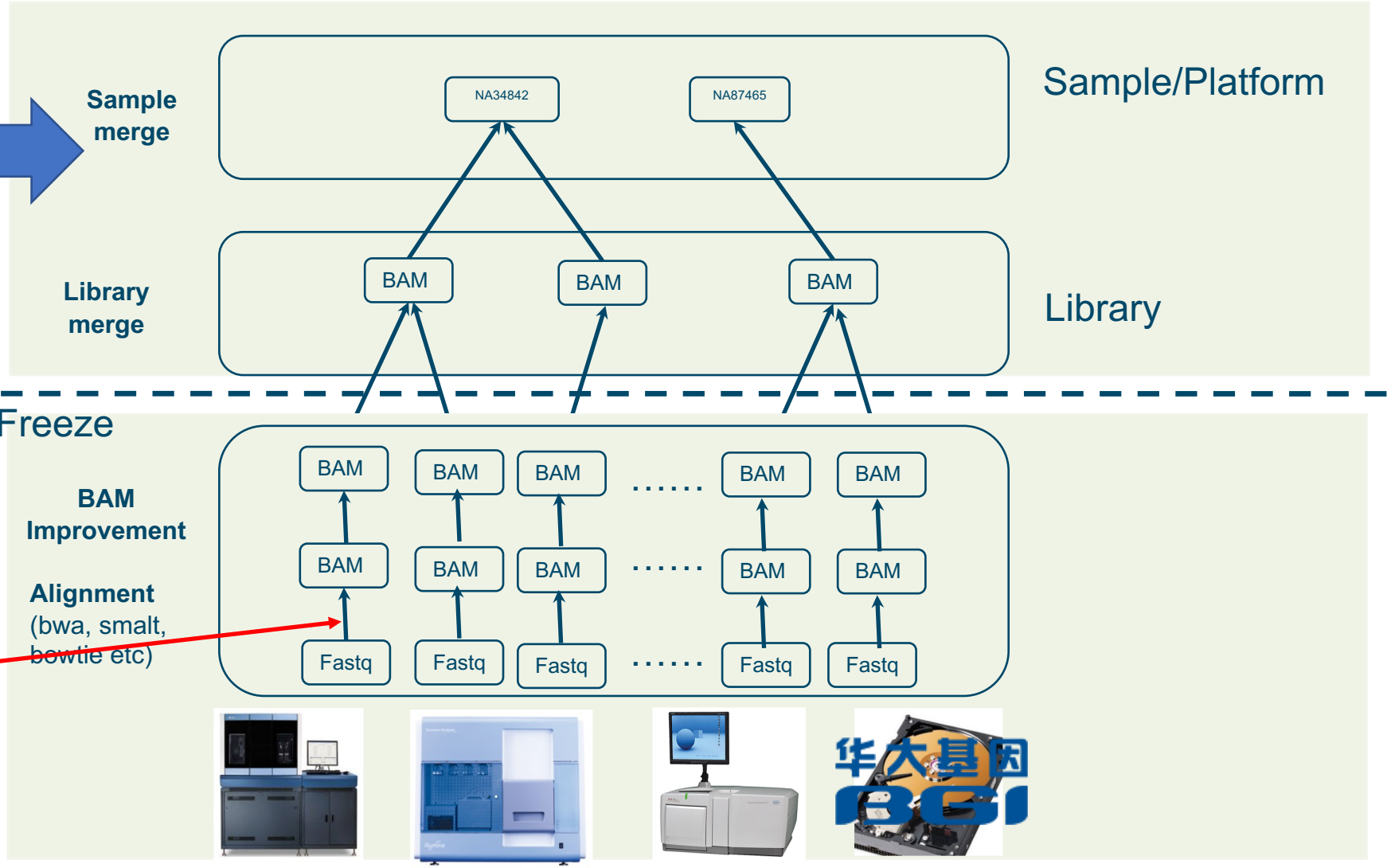
Alignment Viewers

NGS Workflows, QC and BAM Improvement

A typical NGS workflow

There might be 10's, 100's or thousands of samples.

This needs a coordinated workflow:



Alignment is just one part

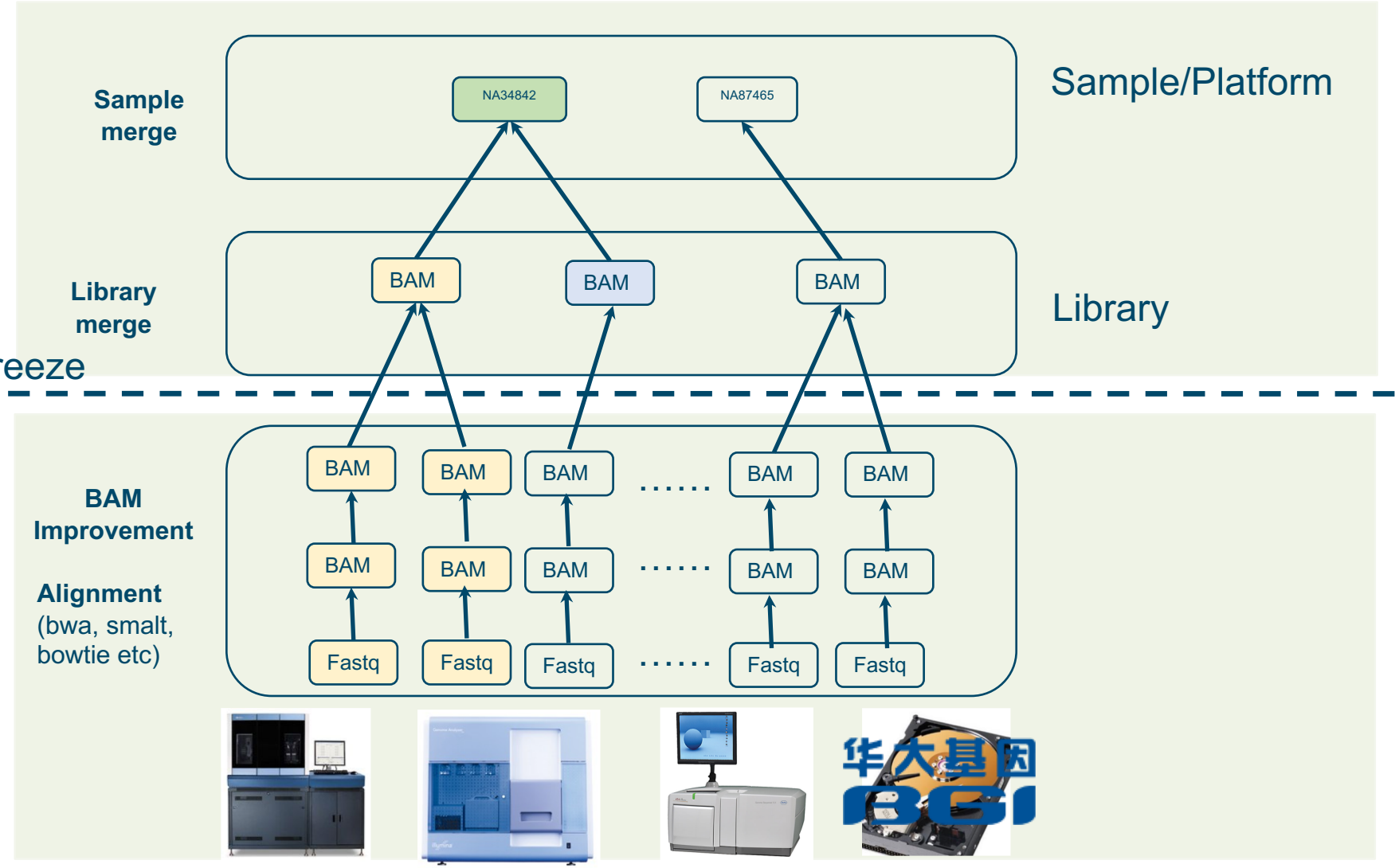
wellcome trust sanger institute

A typical NGS workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



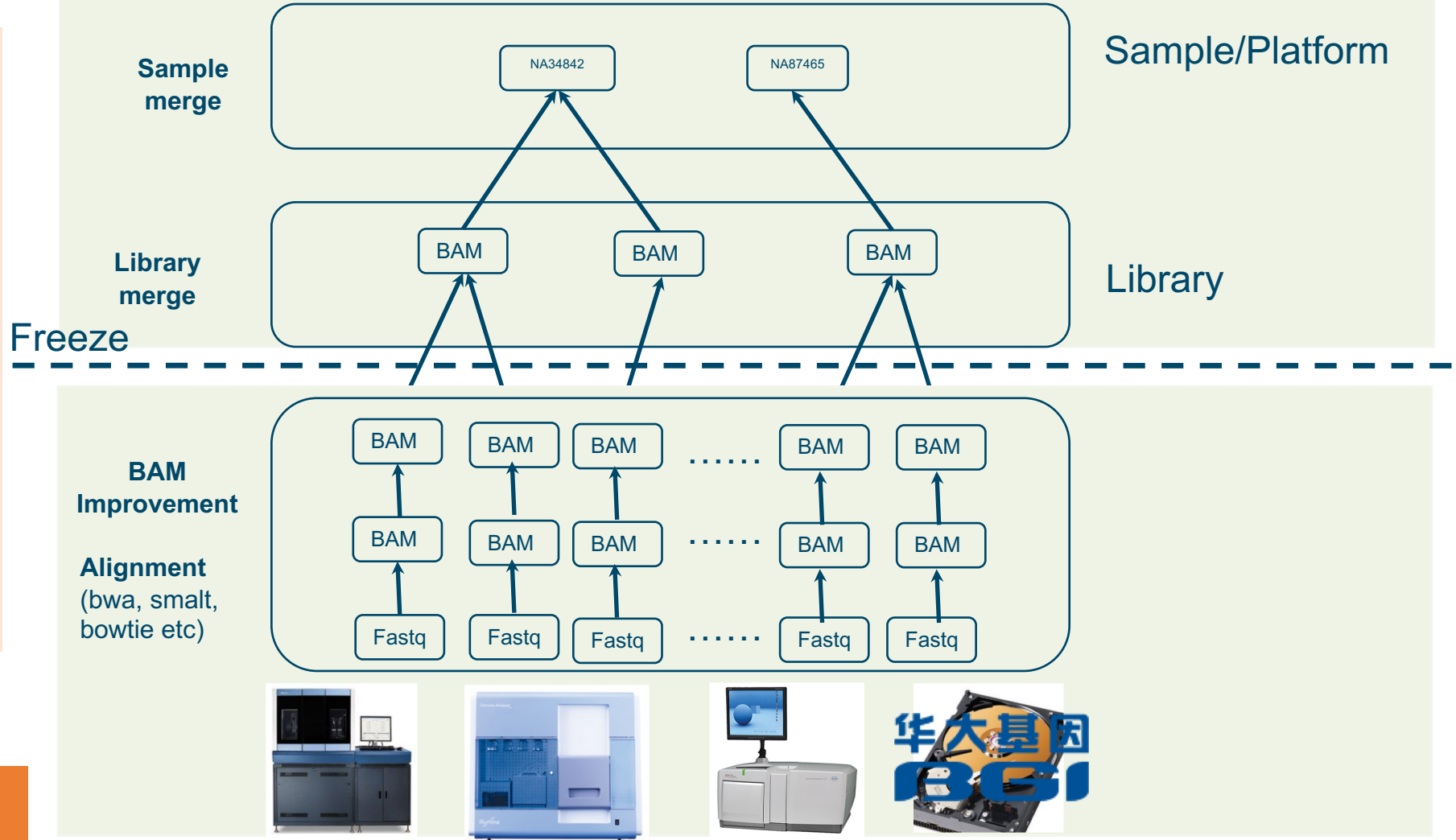
wellcome trust sanger institute

Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



WHY?

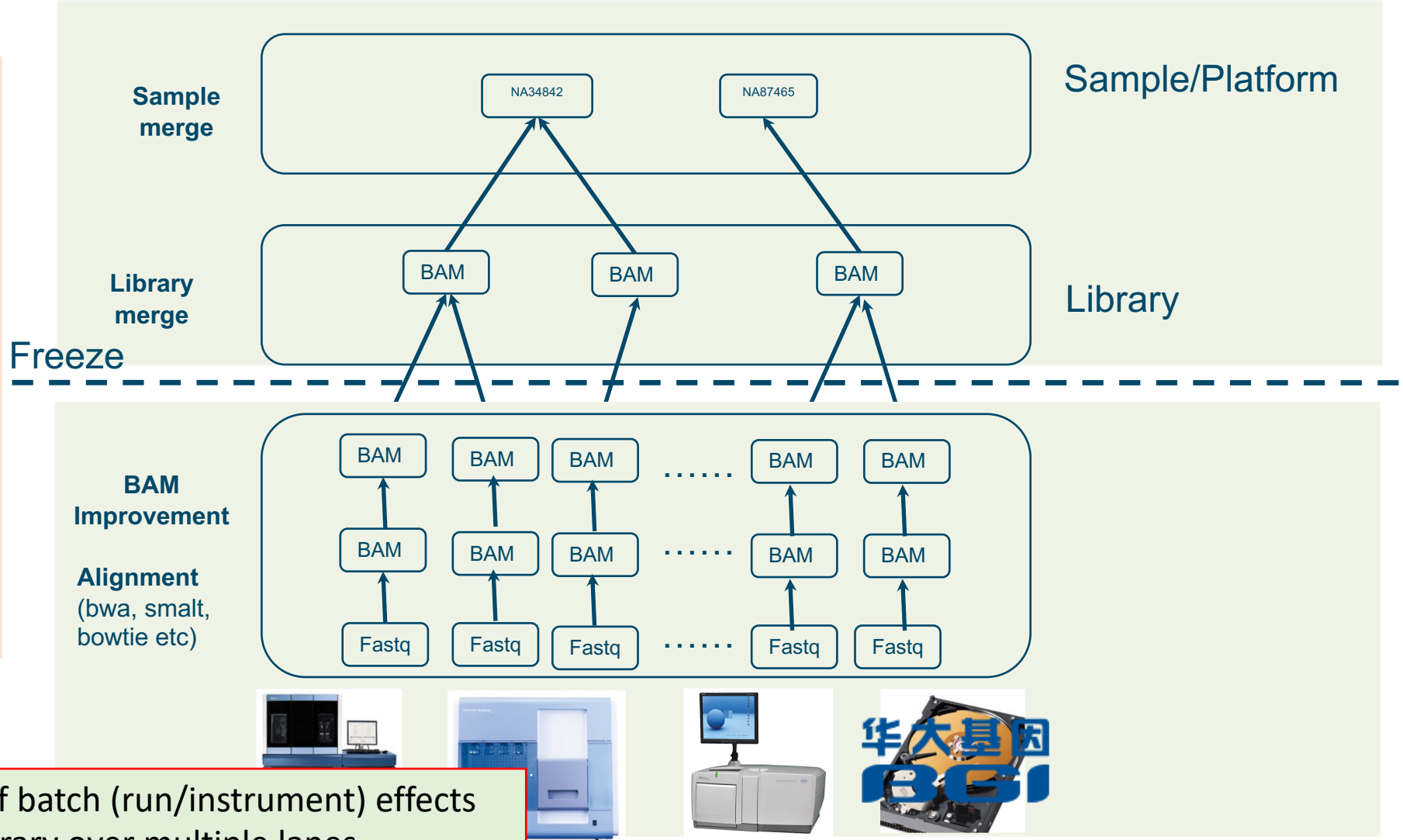
wellcome trust sanger institute

Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



- Mitigation of batch (run/instrument) effects
  - Split library over multiple lanes
- Increasing sequencing depth (more libraries)

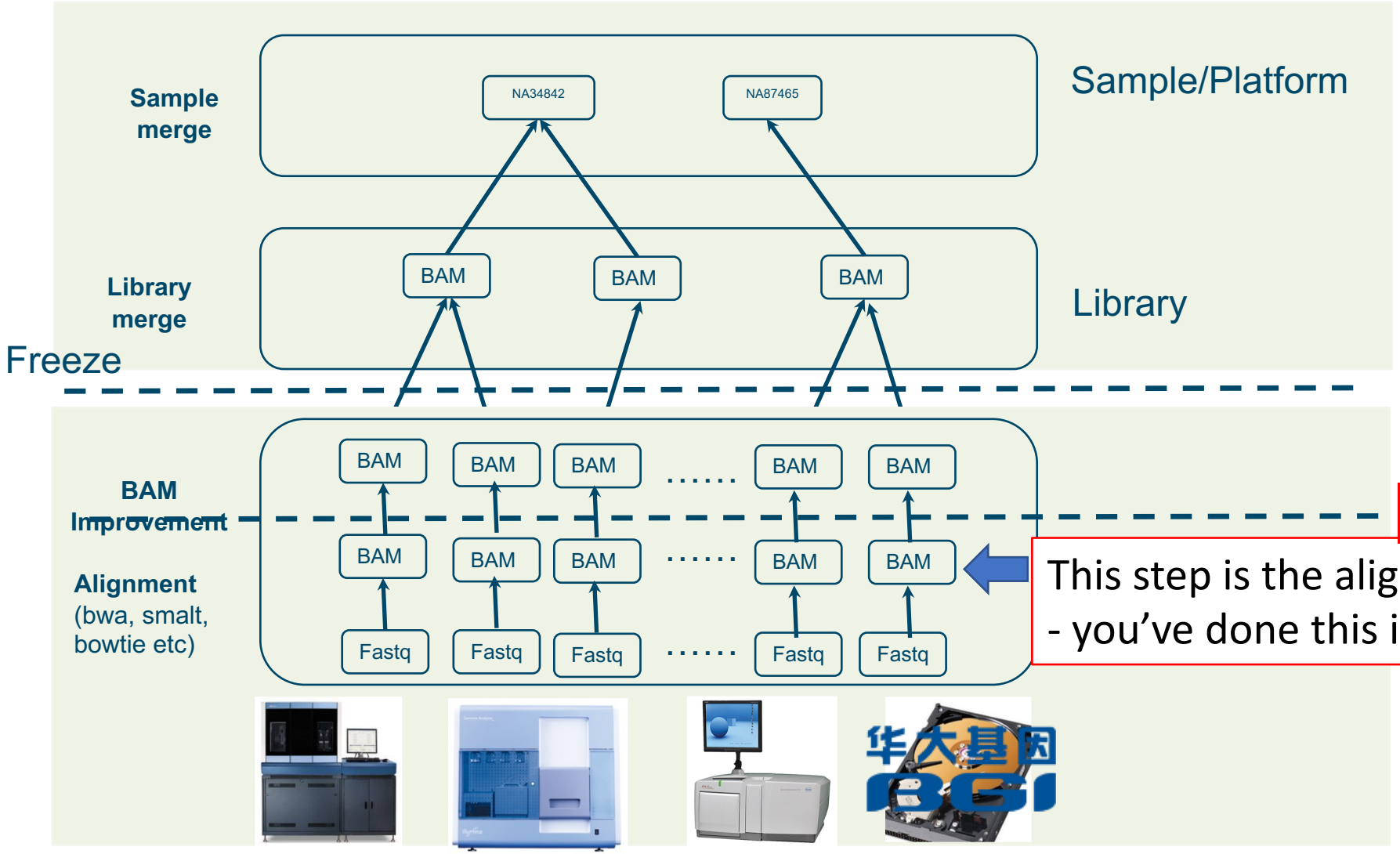
wellcome trust sanger institute

Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



Data QC from alignments – a brief revision

samtools stats my_aligned_reads.bam > my_aligned_reads.stats.txt
plot-bamstats my_aligned_reads.stats.txt

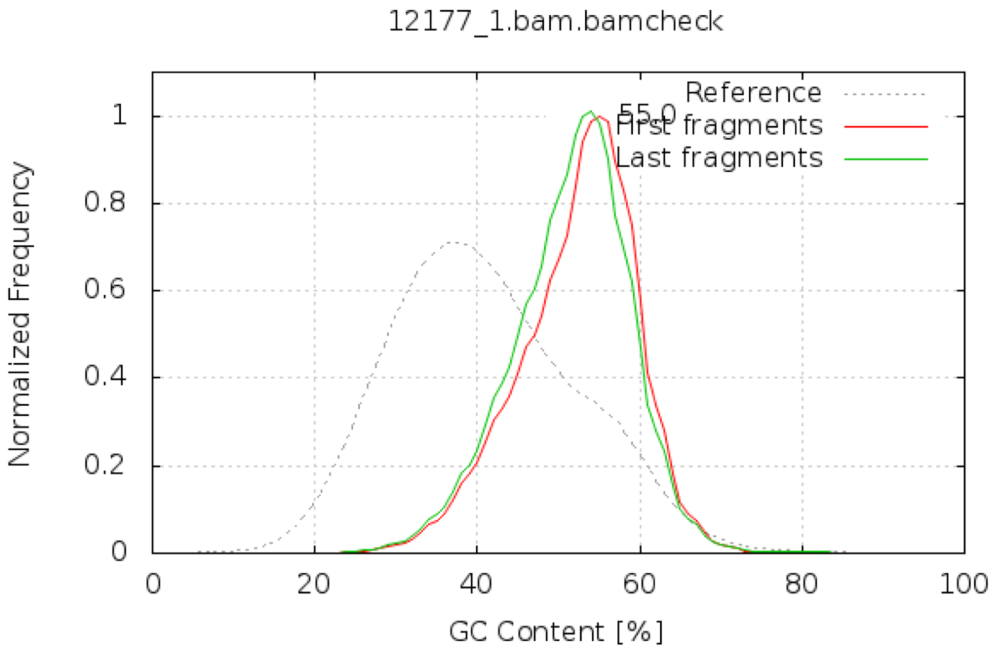
Already covered by the QC lecture: several useful metrics to assess the quality of your data and alignments produced:

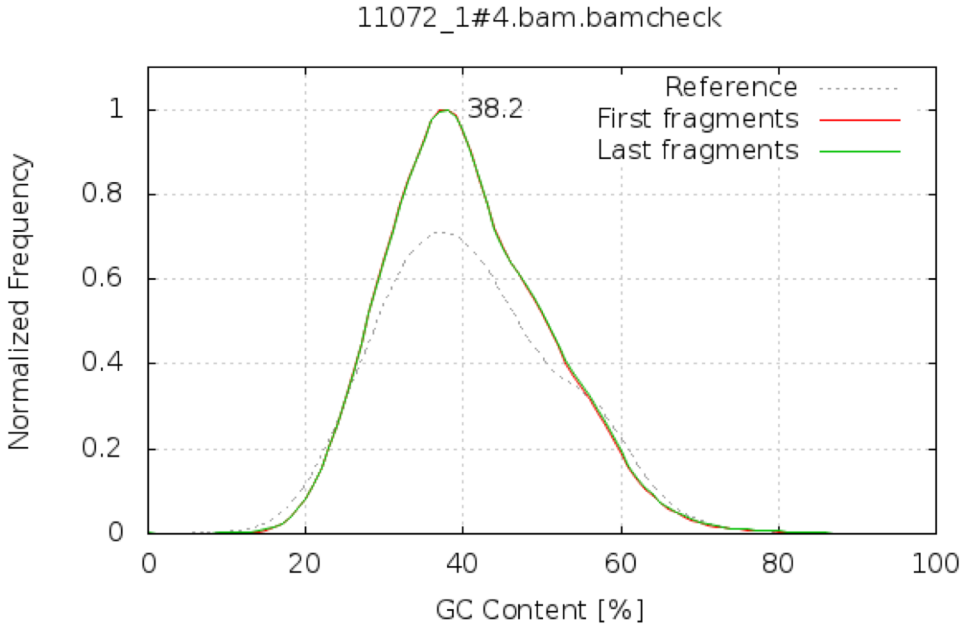
- Mapping rates – should be > 90%
- Duplicate marked rates – some want 5%, *I see anectdotally around 15%*
- GC content vs read depth – which does need mapping (why?). ~ *Flat.*
- Fragment size distribution. *Depends on read length and insert size.*
- Indels by cycle. *No weird spikes.*

Data QC from alignments – a brief revision

And now, a quiz!

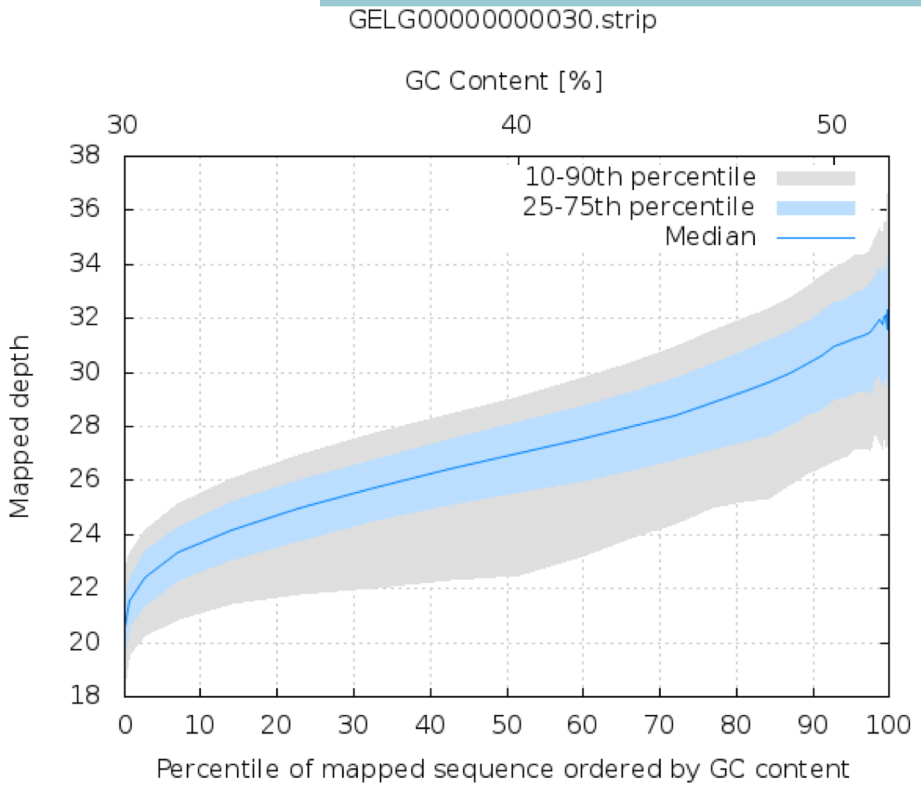
GC of reads

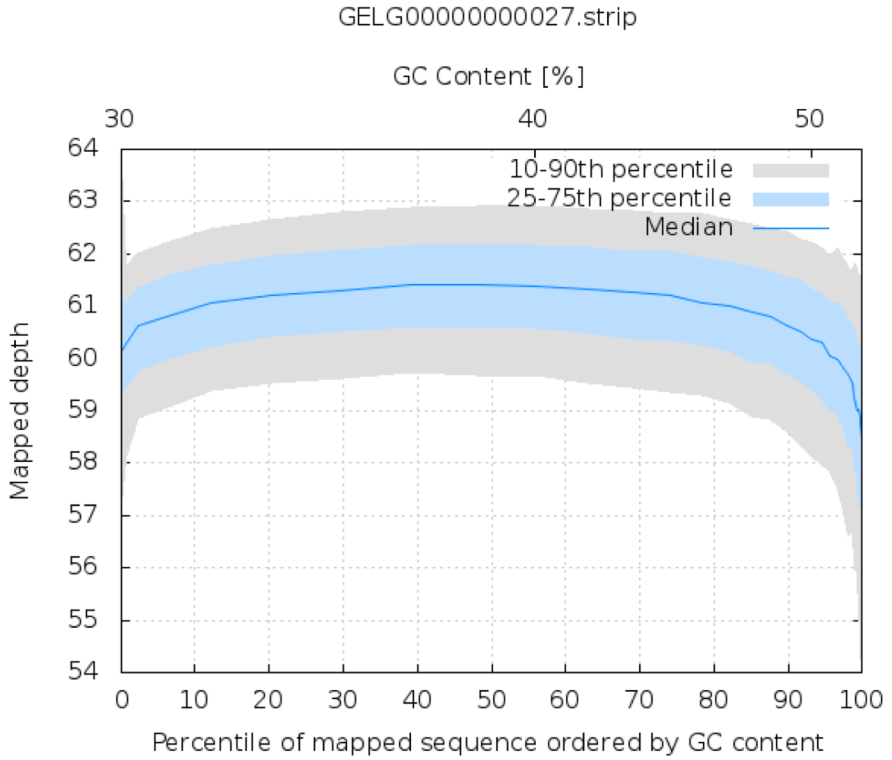




wellcome trust sanger institute

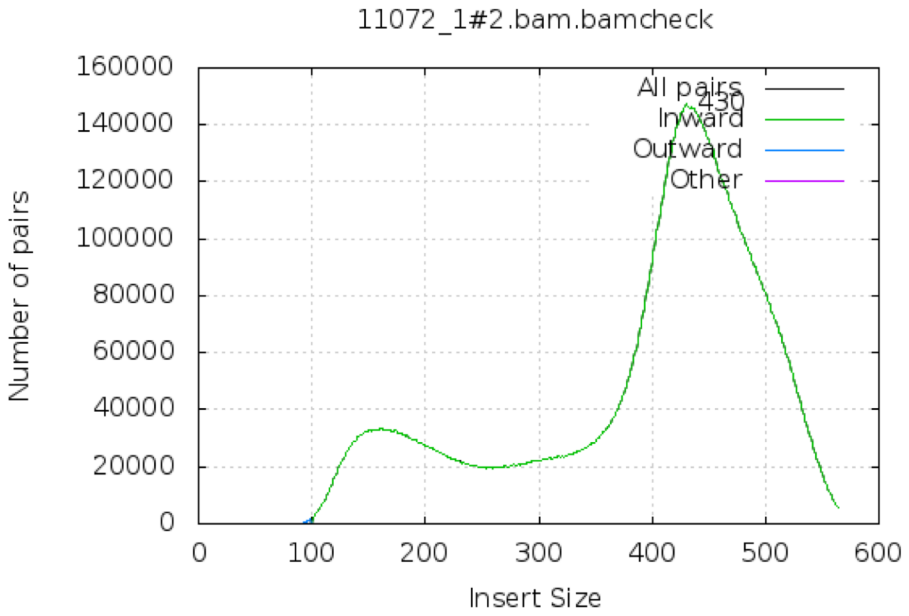
GC vs depth

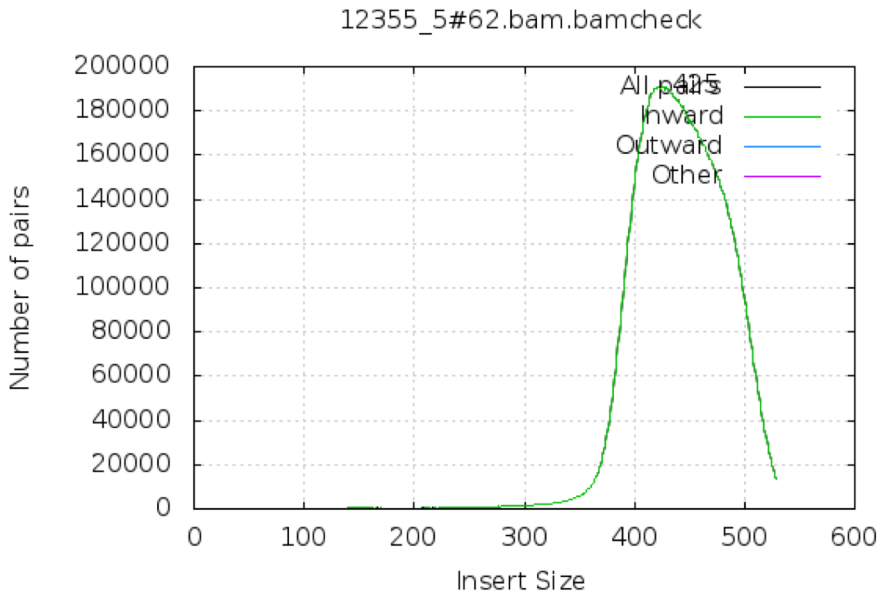




wellcome trust sanger institute

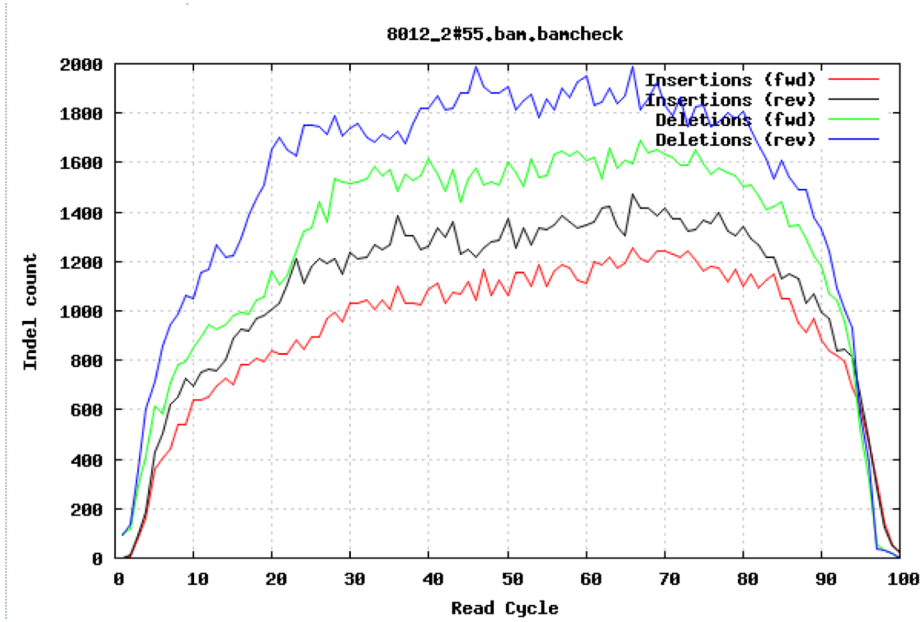
Fragment size

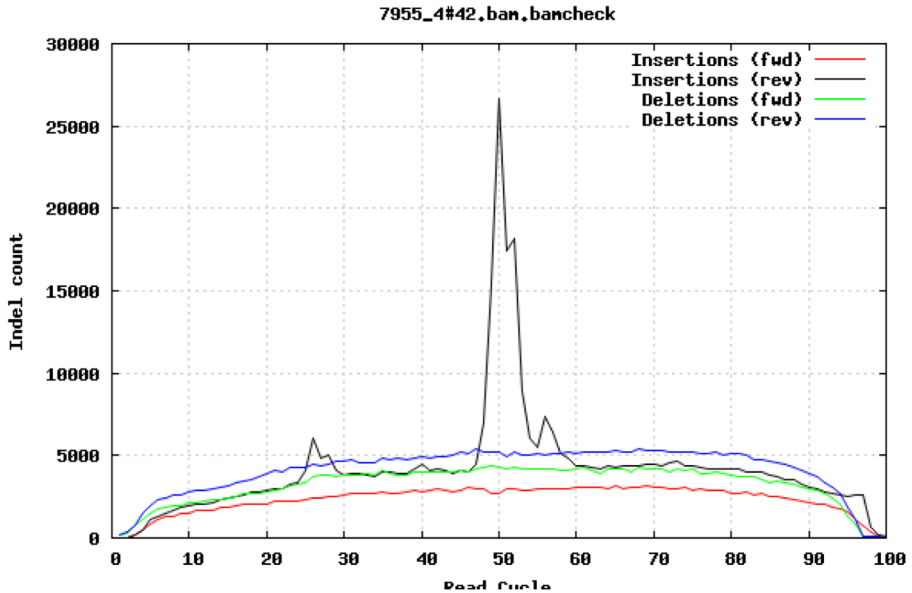




wellcome trust sanger institute

Indels per cycle





wellcome trust sanger institute

Data QC from alignments – a brief revision

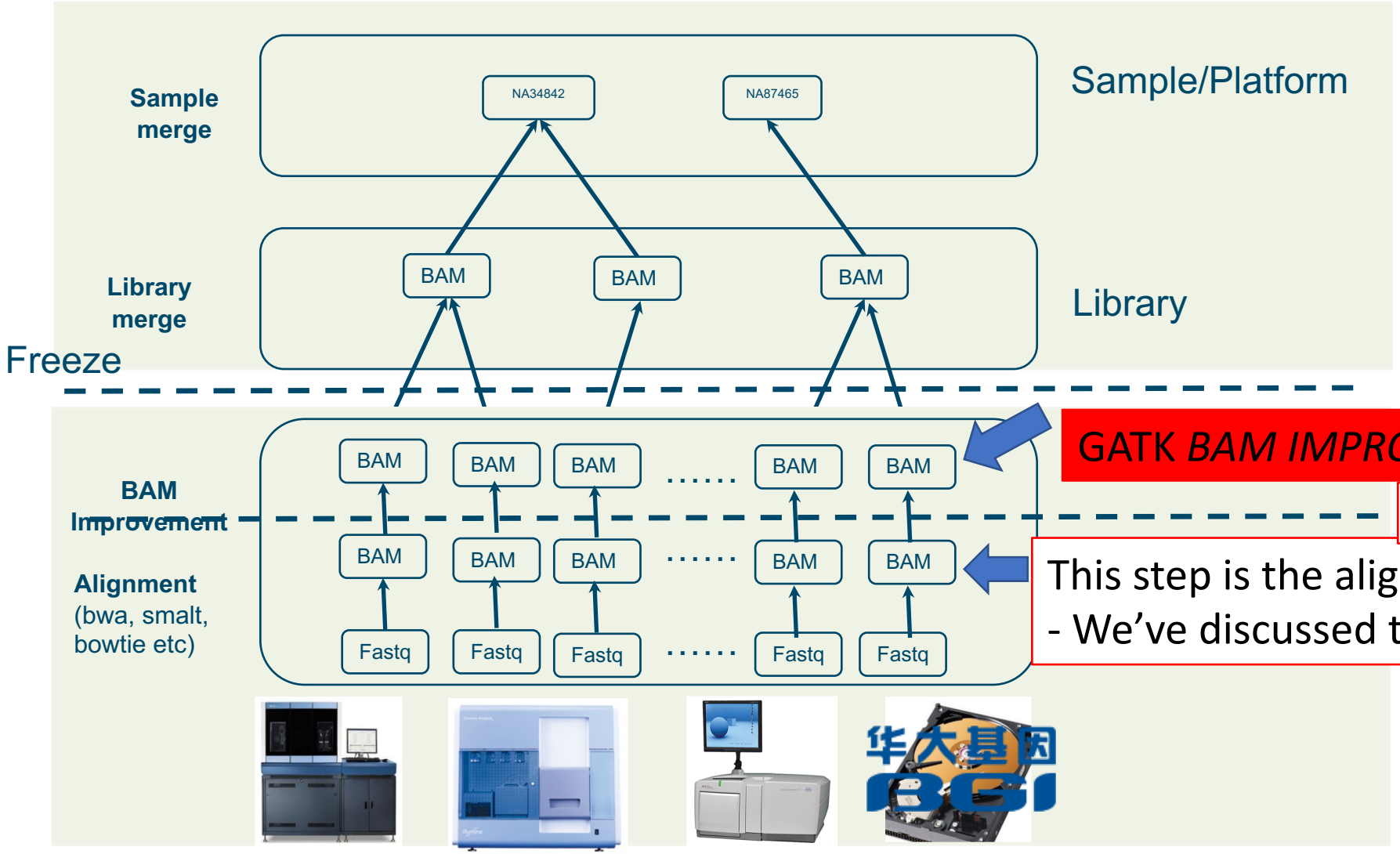
QUIZ IS OVER

Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



wellcome trust sanger institute

BAM improvement

FIRST - align each separate FASTQ: library x lane

Input: BAM

~~**Process 1: INDEL (local) realignment**~~

Process 2: Base quality recalibration

Output: BAM

BAM improvement
Part of …
“GATK Best Practice”

Merge independent lanes BAMS in same library together

Process 3: library - level mark-duplication

BAM Improvement - Indel Realignment

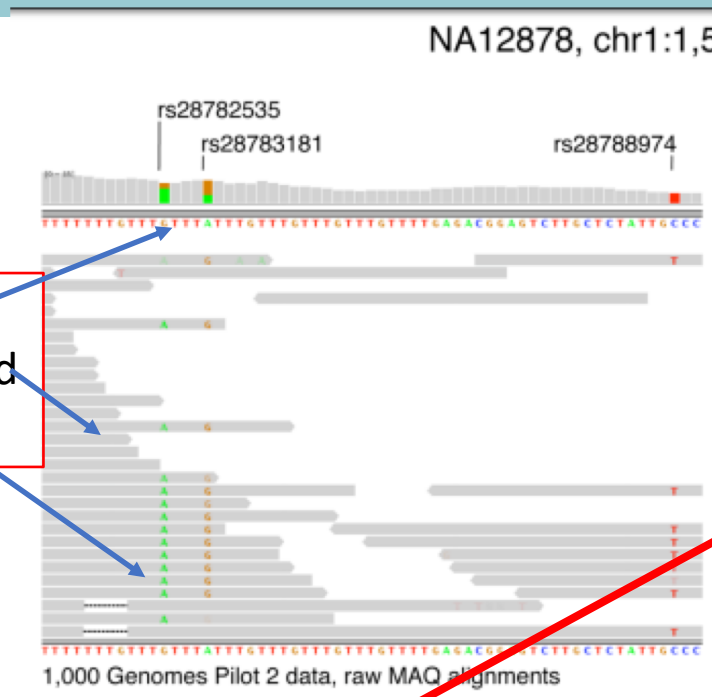
THIS IS NOW DEPRECATED (even by GATK!).
I will mention it briefly because you may see it in different contexts.

PROBLEM: read aligners act on each read individually.

IF the read had an indel,
THEN the aligner might get it right.
OR – if the indel happened near the edge of a read – it may get alignment wrong and miss the indel.
THEN the read would have lots of MISMATCHES instead of a single indel.

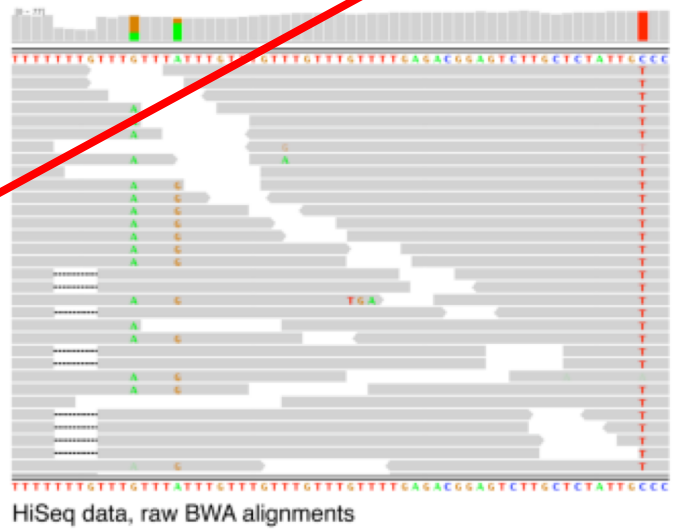
Indel Realignment

M.A. DePristo et al., Nature Genetics 2011



1. "…GTTTGTTTATTT…"
2. These reads mis-aligned
3. These SNPs are false

Before Indel Realign



INDEL Realignment algorithm

FOR indel sites and a BAM file

At each site, model the **indel haplotype** vs **the reference haplotype**

Which alignment minimises ALL the observed mismatches?

Indel or no indel?

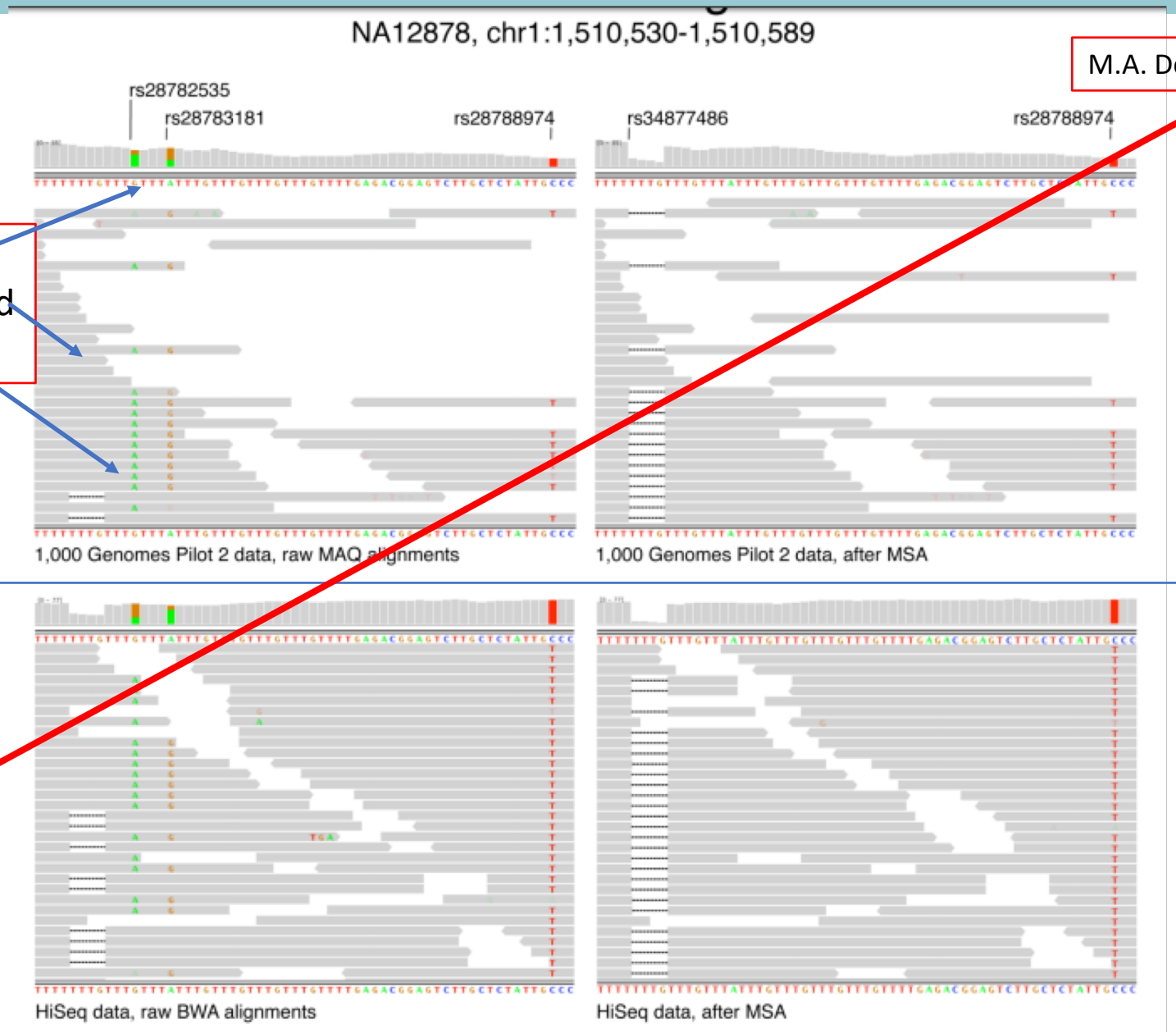
Result: Group alignment for *all* the reads (not one-by-one)

Result: New BAM file produced with read cigar lines modified where indels have been introduced by the realignment process

Indel Realignment

M.A. DePristo et al., Nature Genetics 2011

1. "…GTTTGTTTATTT…"
2. These reads mis-aligned
3. These SNPs are false



Before Indel Realign

After Indel Realign

BAM Improvement - Indel Realignment

THIS IS NOW DEPRECATED (even by GATK!).

Reads are getting longer and aligners are getting better – the problem is going away.

Local realignment now automatically performed by variant callers (downstream) ANYWAY.

So GATK have removed it from their ”best practice” docs.

BAM Improvement - Base Quality Score Recalibration

Each base call has an associated base call quality

- What is the chance that the base call is incorrect?
  - Illumina evidence: intensity values + cycle
- Phred values (log scale)
  - Q10 = 1 in 10 chance of base call incorrect
  - Q20 = 1 in 100 chance of base call incorrect
- Accurate base qualities essential measure in variant calling

Rule of thumb: Anything less than Q20 is not useful data

BUT:

The base quality scores produced by a sequencer can be influenced by *systematic technical error*:

They can vary with sequence context, position in read etc.

Therefore the quality score can be off

Therefore variant calling can be influenced.

BQSR: adjusts the quality of each base to adjust for these systematic errors

wellcome trust sanger institute

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3409179/

Base quality recalibration

The idea of recalibration:

- Use the alignment of your reads to a human reference
- **Remove** all known variants dbSNP+1000G etc SNP sites
- **Assume** all other mismatches in your data **are sequencing errors**
- Now you can infer the mean observed ("empirical") error rate for these bins:
  - Position of base in read (1 => length of read)
  - Dinucleotide Sequence Context: AA, AT …, CA, CT …
  - Empirical quality = number of mismatches / total number in bin.
- Compare empirical rate to *reported* sequencer base quality in each of these categories.
- Derive an adjustment for each category, to be applied to each reported base quality value

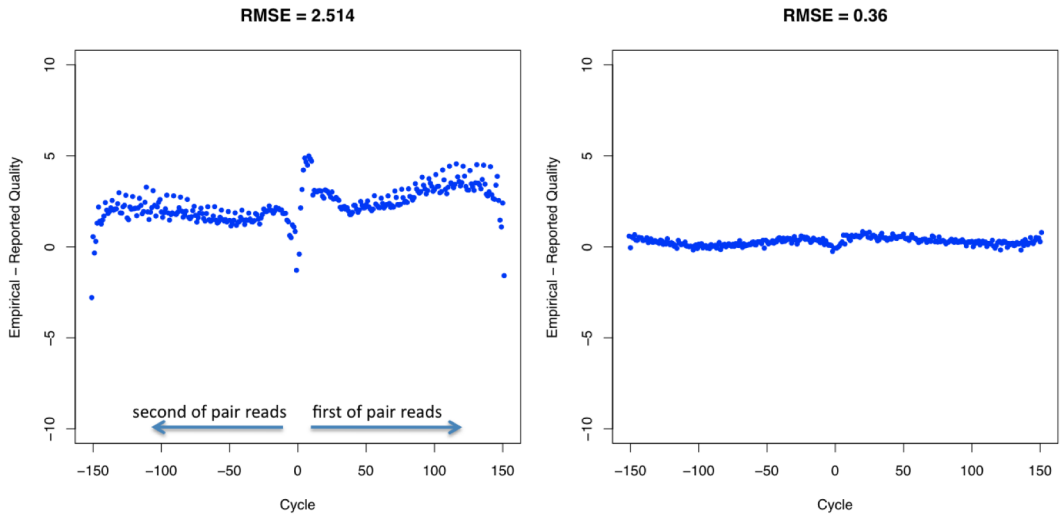
e.g. Sequencer reports Q25 base calls for a "T" in context "AT"

- After alignment - it may be that "T" in context "AT" actually mismatches the reference at a 1 in 100 rate, so are actually Q20
- Adjustment: for every T in "AT" – subtract 5 from BQ. (30=>25, 21=> 16 etc)

NOTE: requires a reference genome and a catalog of variable sites

Base quality recalibration effects

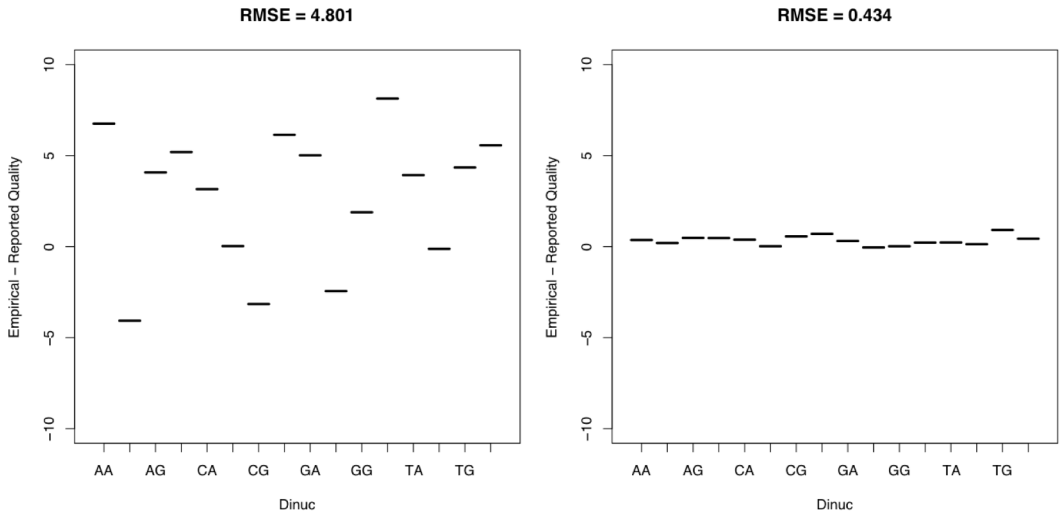
Residual Error by Machine Cycle



Original Data

After GATK Recalibration

Residual Error by Dinucleotide



Original Data

After GATK Recalibration

wellcome trust sanger institute

Data production workflow

One sample

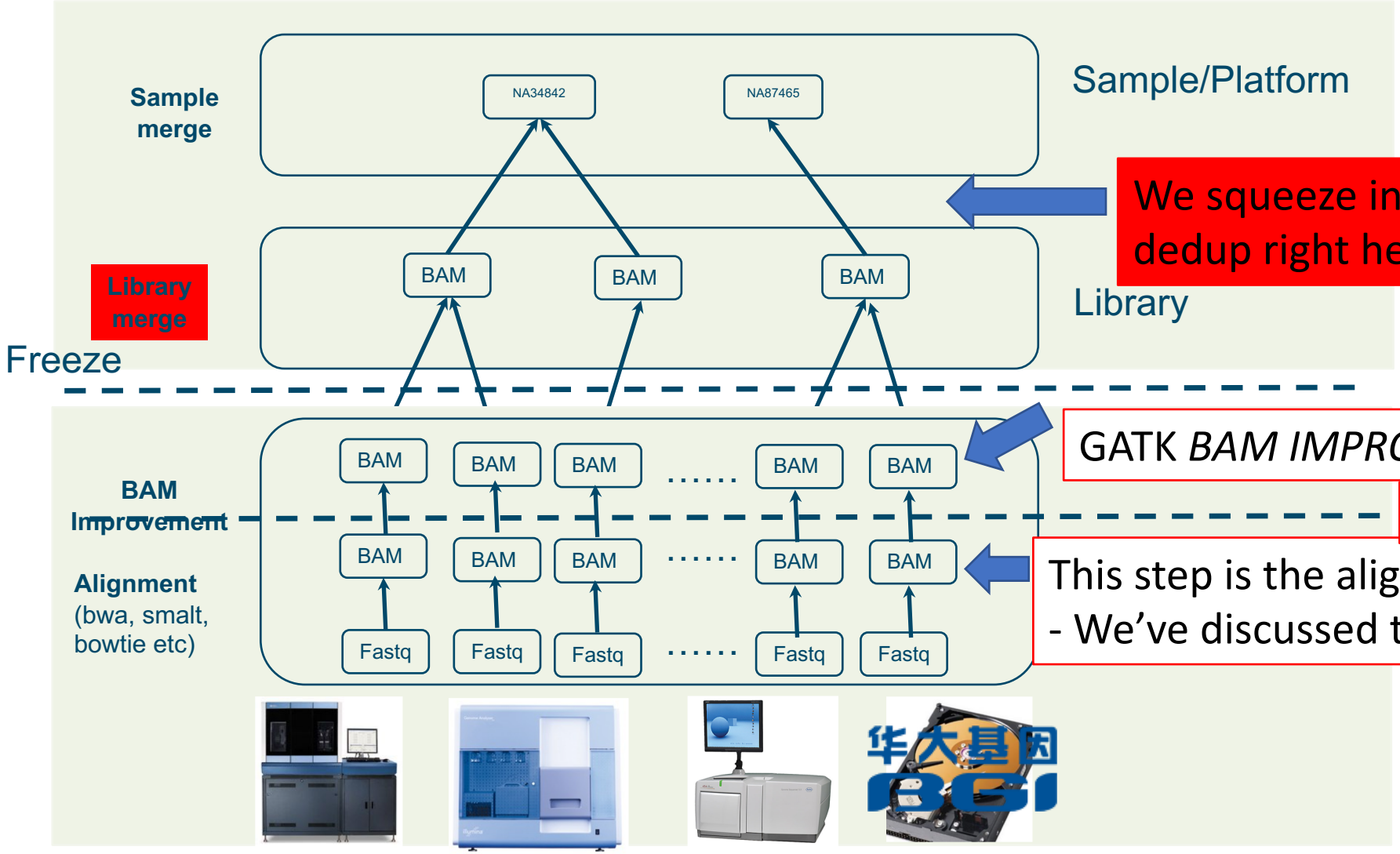
…spread over…

Multiple libraries

One library

…spread over…

Multiple runs (lanes)



wellcome trust sanger institute

Library duplicates

All second-gen sequencing platforms are NOT single molecule sequencing

- PCR amplification step in library preparation
- Can result in duplicate DNA fragments in the final library prep.
- PCR-free protocols do exist – require larger volumes of input DNA

Problem: can result in false SNP calls

- Duplicates manifest themselves as high read depth support

Solution:

- Align reads to the reference genome
- Identify read-pairs where the ***outer ends*** map to the same position on the genome and remove all but 1 copy
  - ➢ Samtools: samtools rmdup
  - ➢ Picard/GATK: MarkDuplicates

Generally low number of duplicates in good libraries (<5%). *But I see ~15%*

Data production workflow

One sample

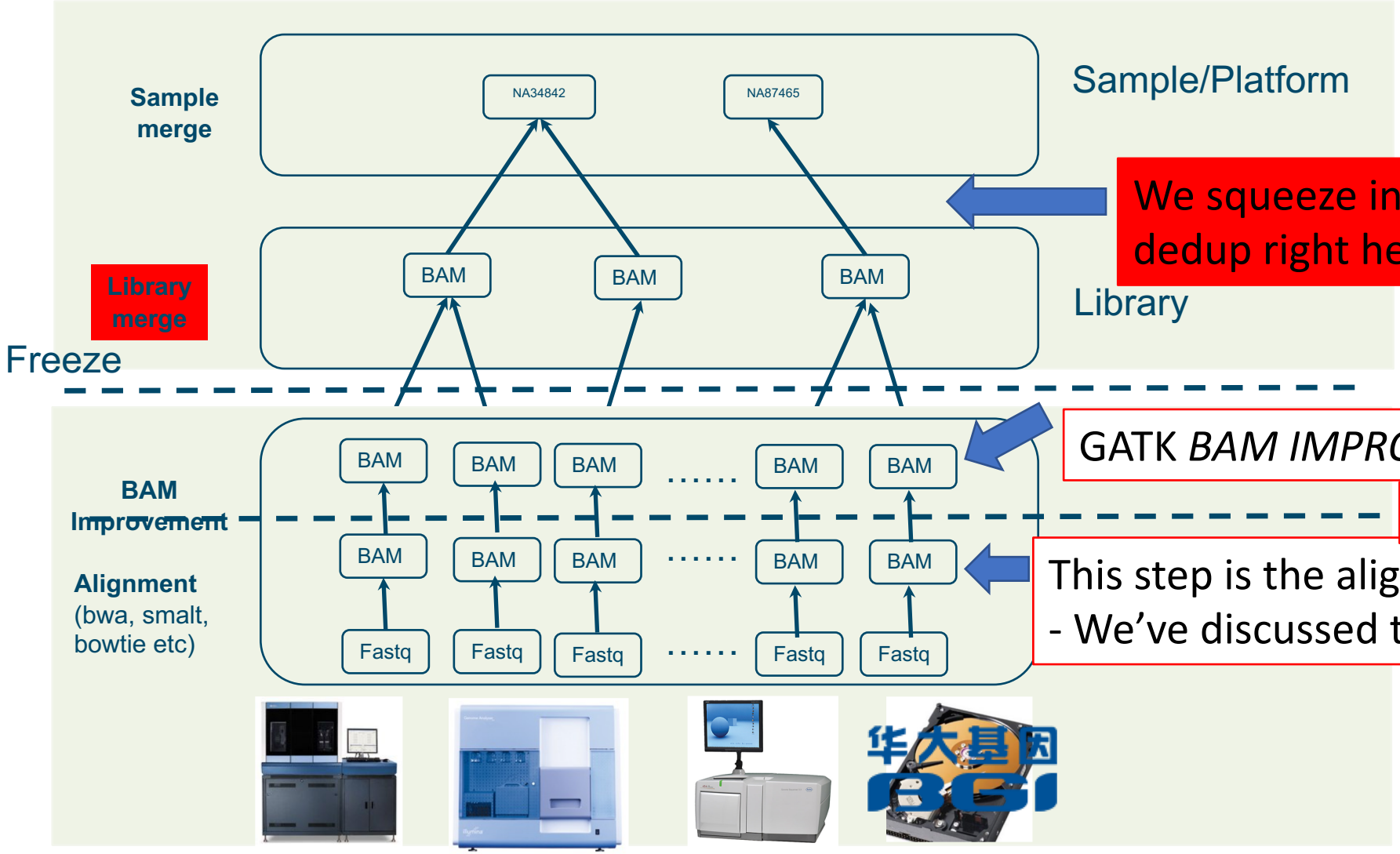
…spread over…

Multiple libraries

One library

…spread over…

Multiple runs (lanes)



wellcome trust sanger institute

Overview

Intro

Methods / Aligners

Alignment Outputs

Alignment Viewers

Alignment summary statistics and QC – BRIEF REVISION

After alignment: BAM file improvement before variant calling

If you are an consumer then you should have a grasp of this

An informatics group should have control of this, but you should be aware that it's being done.