H3ABioNet
Pan African Bioinformatics Network for H3Africa

wellcome connecting science

Next Generation Sequencing
Bioinformatics Course 2021

Read alignment

Reference sequence, alignment methods, scaling up workflows, IT costs and BAM improvement

NGS read alignment

Material prepared by: Vivek Iyer, Thomas Keane and Ant Doran

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 - REFERENCES

Methods / Aligners

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

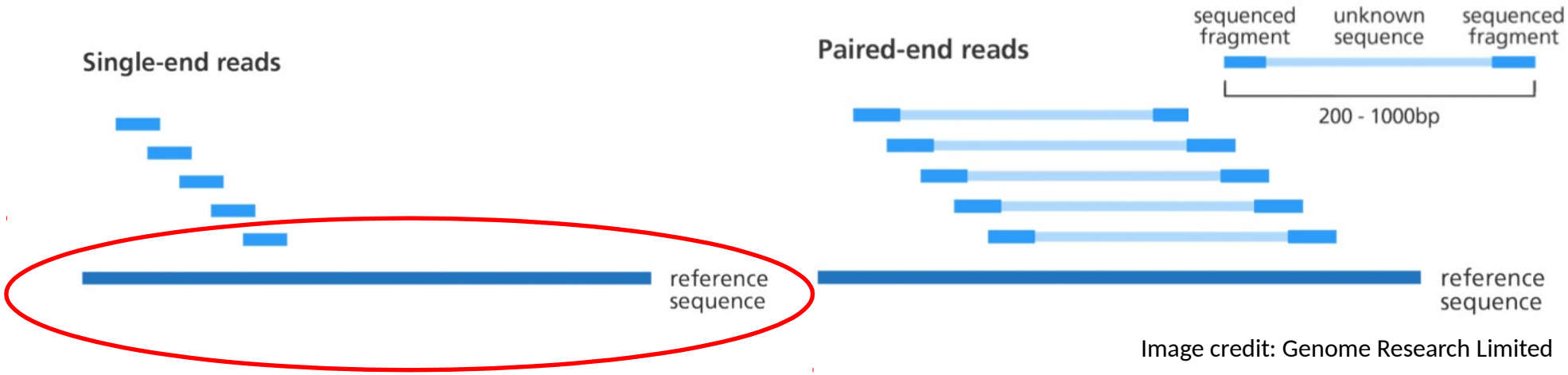


Image credit: Genome Research Limited

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

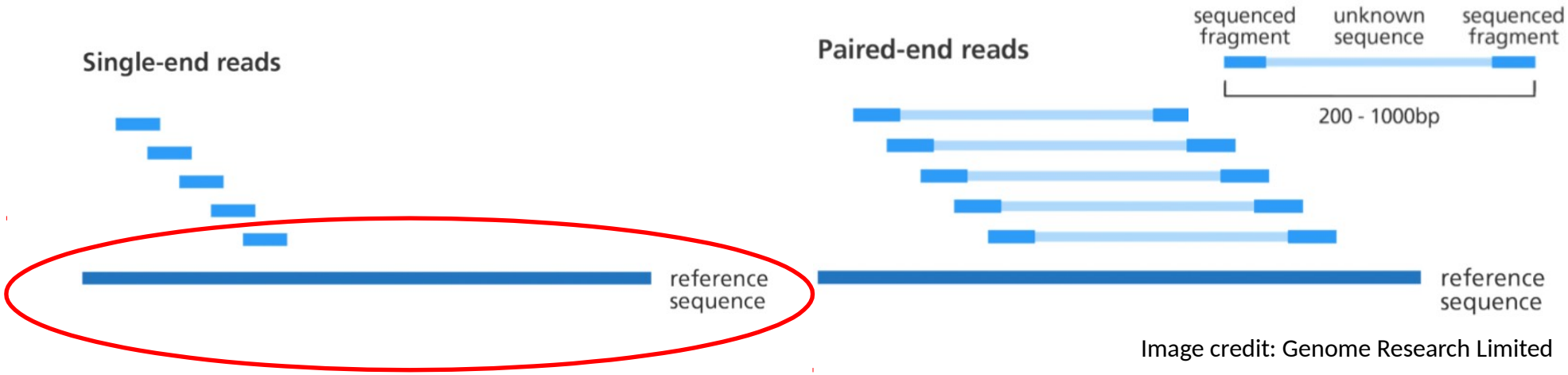


Image credit: Genome Research Limited

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**

TREE OF LIFE PROGRAM AT SANGER WILL CHANGE THIS FOREVER

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

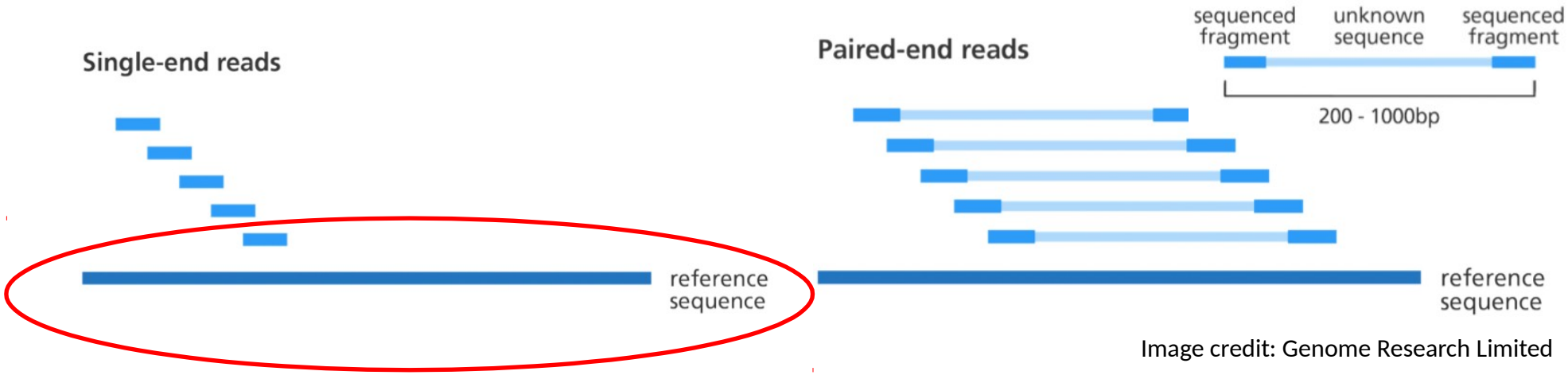


Image credit: Genome Research Limited

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?

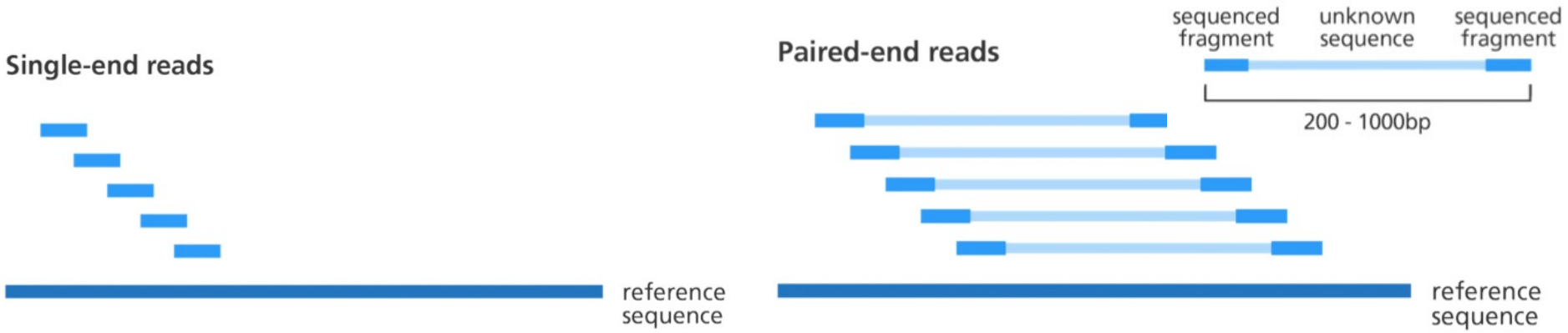
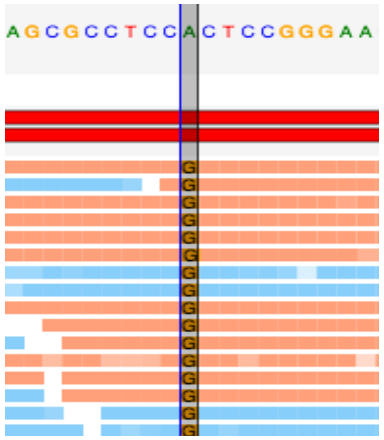
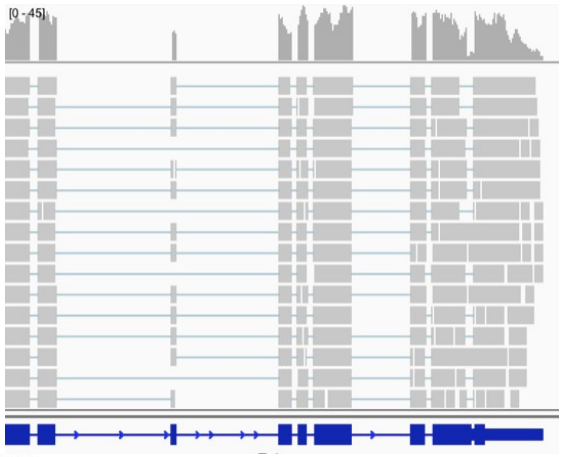


Image credit: Genome Research Limited

Align DNA: Identify variation



ALIGN RNA: Transcript abundance



Overview

Intro

Methods / Aligners

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

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

WRITE OUT ALIGNMENT:

S1 A C T G
S2 A C - G

S1

S2		A	C	T	G
	0	0	0	0	0
A	0	-1, -1 1 = 1	-1, 0 -1 =0	-1, -1 -1=0	-1 -1 -1=0
C	0	0 -1 0=0	-1 -1 2=2	-1 1,-1= 1	-1 0, -1= 0
G	0	-1 -1 -1= 0	1 -1 -1=1	0 0 1 = 1	-1 -1 2 = 2

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)

SMITH-WATERMAN is WAY TOO SLOW:

Two primary faster approaches:

- Hash-based alignment
- Suffix/prefix tries

Hash table alignment

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

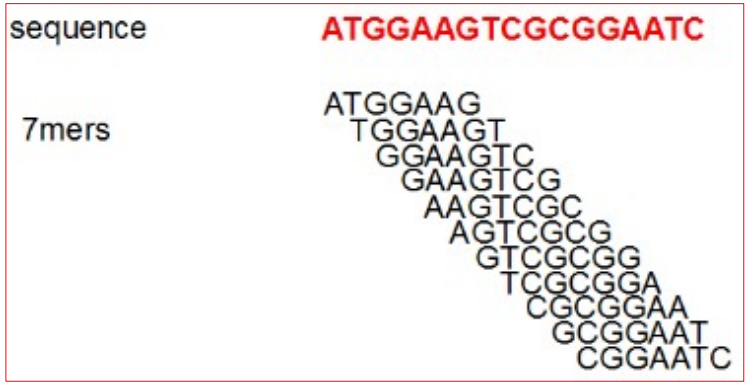
First thing: “wrap up” the reference genome to bring it to the reads – build a kmer hash

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:

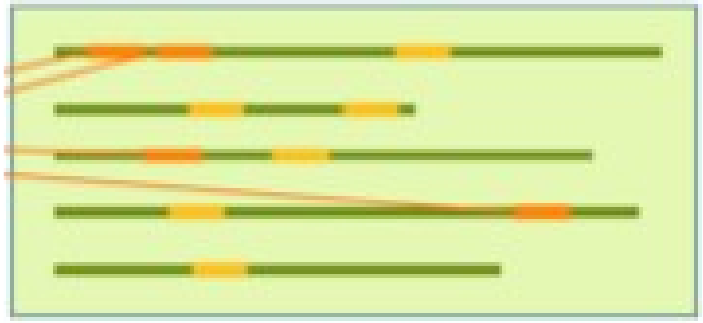
Hash table will be several Gbytes in size for human genome



Hash table alignment

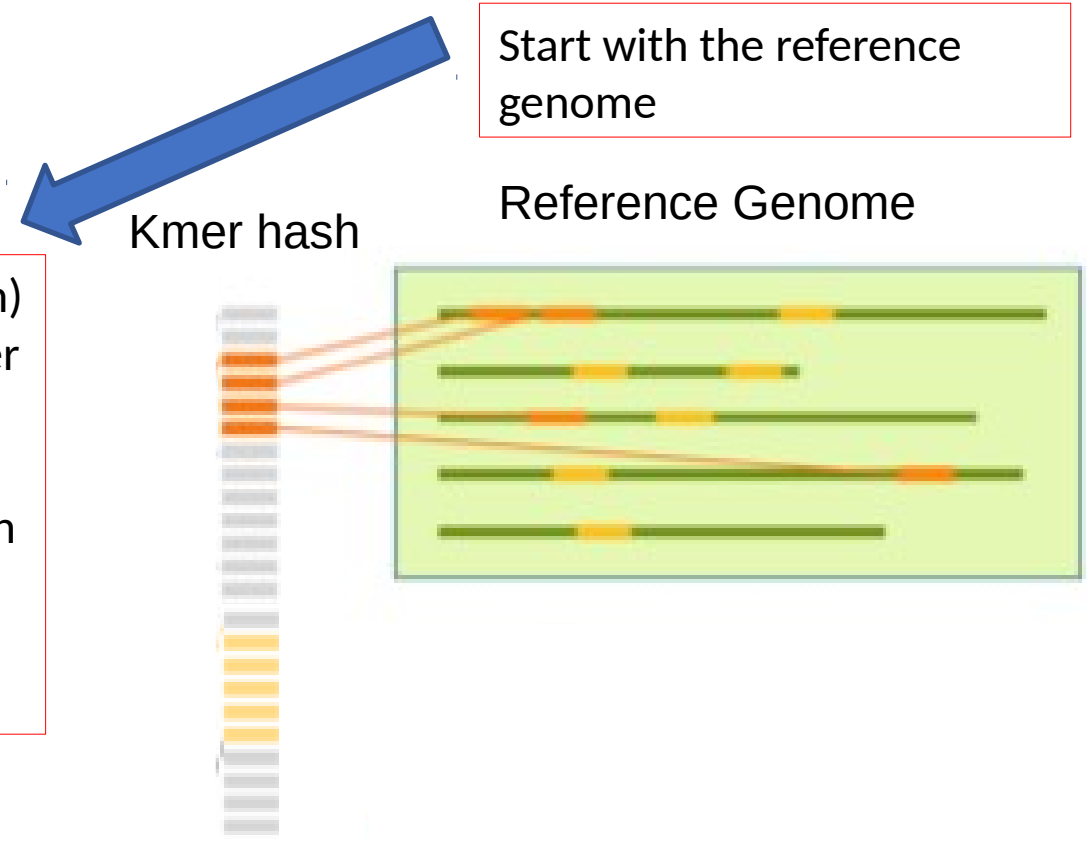
Start with the reference genome

Reference Genome



Hash table alignment

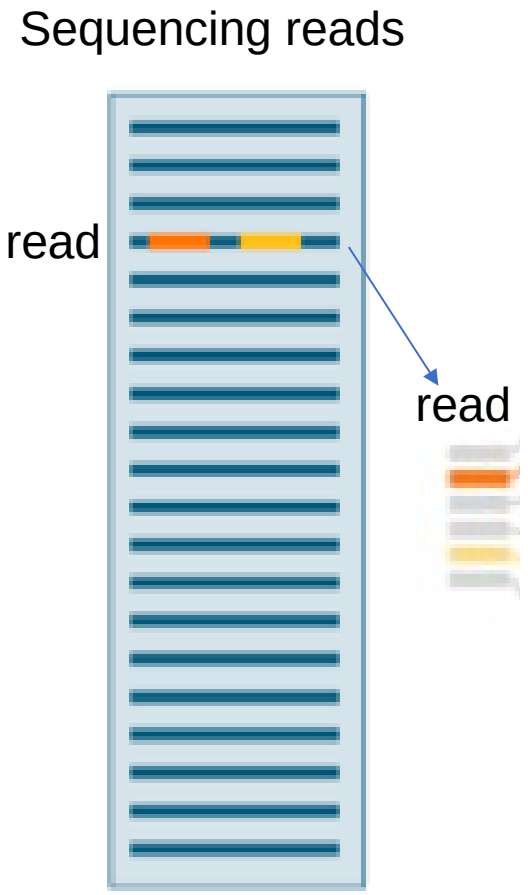
Start with the reference genome



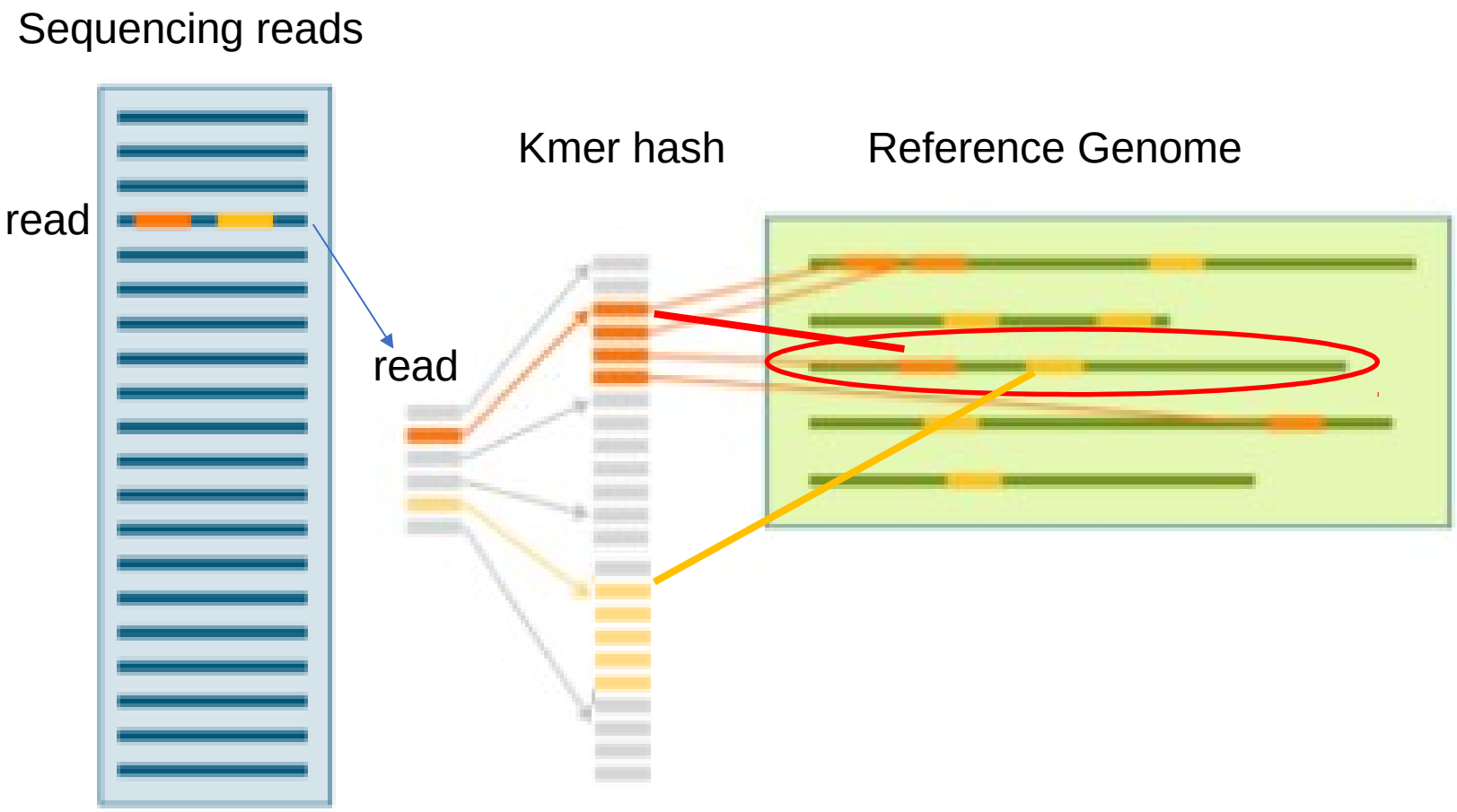
Build a data structure (hash) allows rapid lookup by kmer sequence.

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

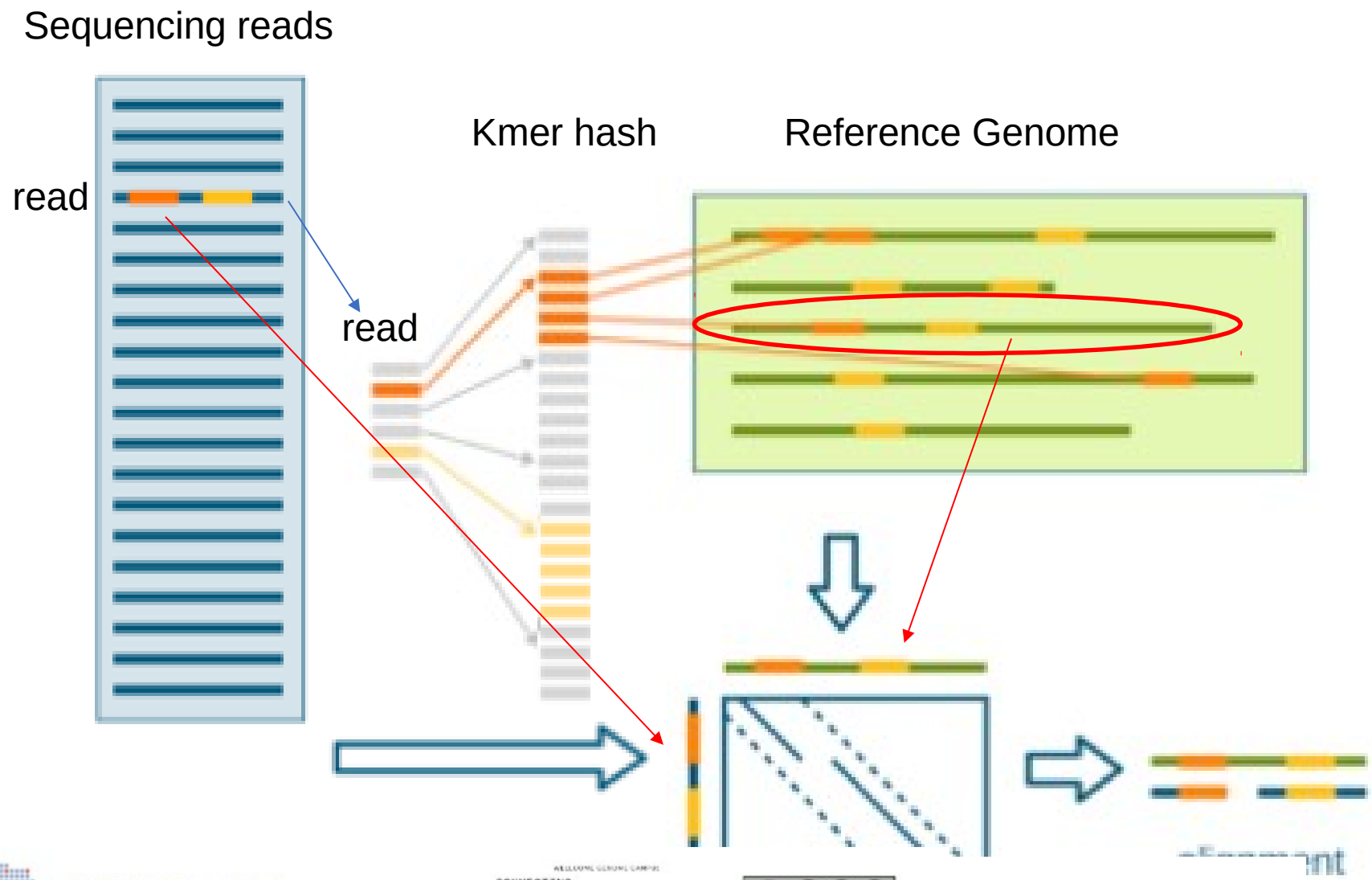
Hash table alignment



Hash table alignment



Hash table alignment



Hash table alignment

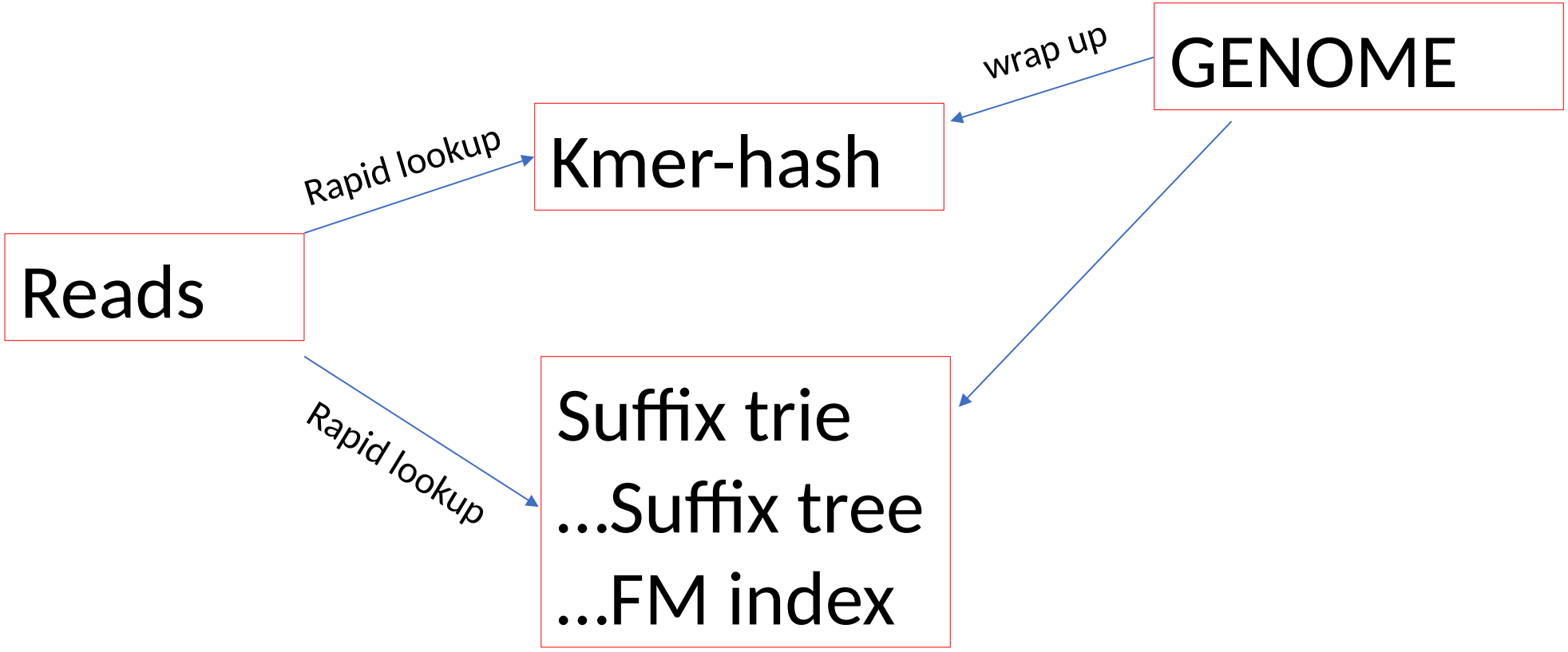
Note: K-mer is a short fixed sequence of nucleotides

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

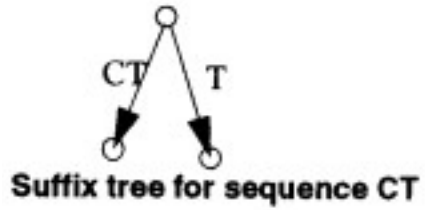
Perform Smith-Waterman alignment to fully align the read to the region
Output the alignment of each read onto the reference in BAM (or equivalent) format.

Wrapping up the genome and bringing it to your 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.



Suffix tree for sequence CT

Genome

Trie based on genome (slightly compacted)

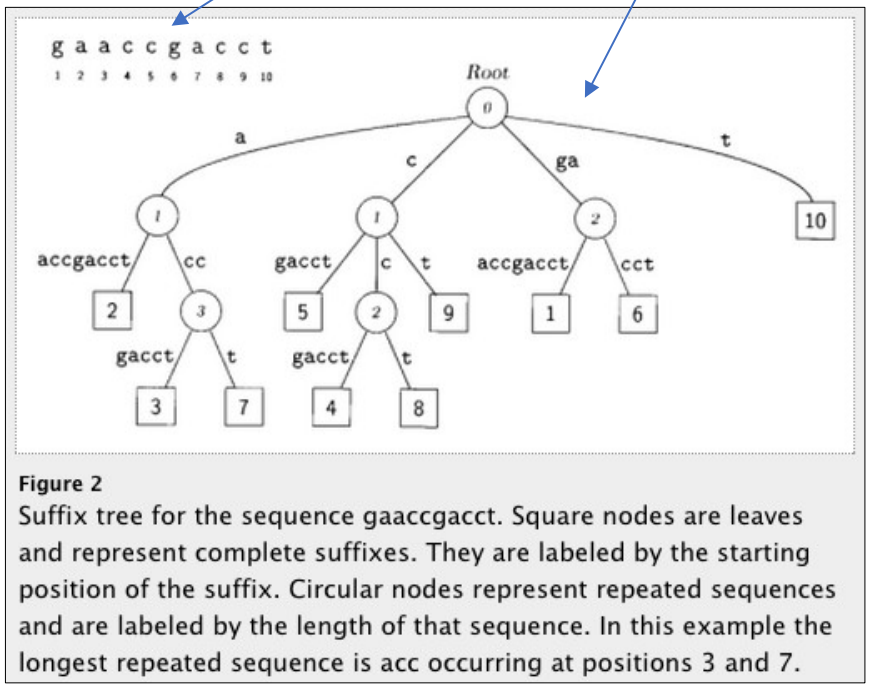


Figure 2
Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. In this example the longest repeated sequence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

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

Delcher et al (1999) *NAR*

Alignment Limitations

Read Length and complexity of the genome

- Very short reads difficult to align confidently to the genome
- Low complexity genomes present difficulties
  - Malaria is 80% AT - lots of low complexity AT stretches

Alignment around indels

- Next-gen alignments tend to accumulate false SNPs near true indel positions due to misalignment
- Smith-Waterman scoring schemes generally prefer a SNP rather than a gap open
  - Tools developed to do a second pass on a BAM and locally realign the reads around indels and 'correct' the read alignments

High density SNP regions

- Seed and extend based aligners can have an upper limit on the number of consecutive SNPs in seed region of read (e.g. Maq - max of 2 mismatches in first 28bp of read, bowtie has a max mismatch arg)
- BWT based aligners work best at low divergence

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 read alignment

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

Typically represented as a phred score (log scale) – probability that the read maps elsewhere

Q0 = read placed with identical score elsewhere

Q10 = 1 in 10 probability of being incorrect

Q20 = 1 in 100 probability of being 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 t

Hence prefer paired-end sequencing

Mapping qualities

Mapping qualities

MapQ: 0 MapQ: 60

Some alignment options (what's changing …)

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)

Some alignment options – near future

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**MEM2**	Yes	SE, PE	FASTQ FASTA	SAM	No	No	BWT + extend: Local alignment
DRAGEN ALIGNER	**Yes**	**SE,PE**	**FASTQ, BAM, CRAM**	**CRAM, BAM**	**No**	**Yes**	**Hashtable lookup + extend**
Bowtie2	Yes	SE, PE	FASTQ FASTA qseq	SAM	Yes	No	BWT + 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

Spliced aligns (RNASeq)

Independently performs spliced aligns (no preprocessing)

Alignments – scaling up

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

Alignments – scaling up

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

ANSWER:

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.***

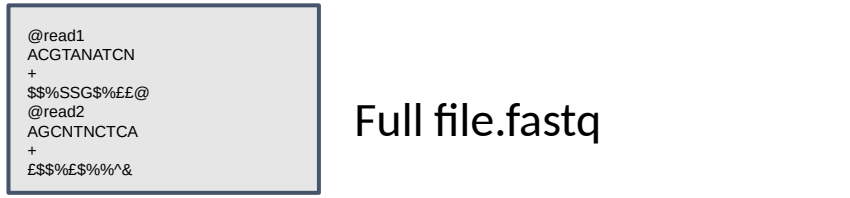
Alignments – scaling up

This is why we like compute clusters!

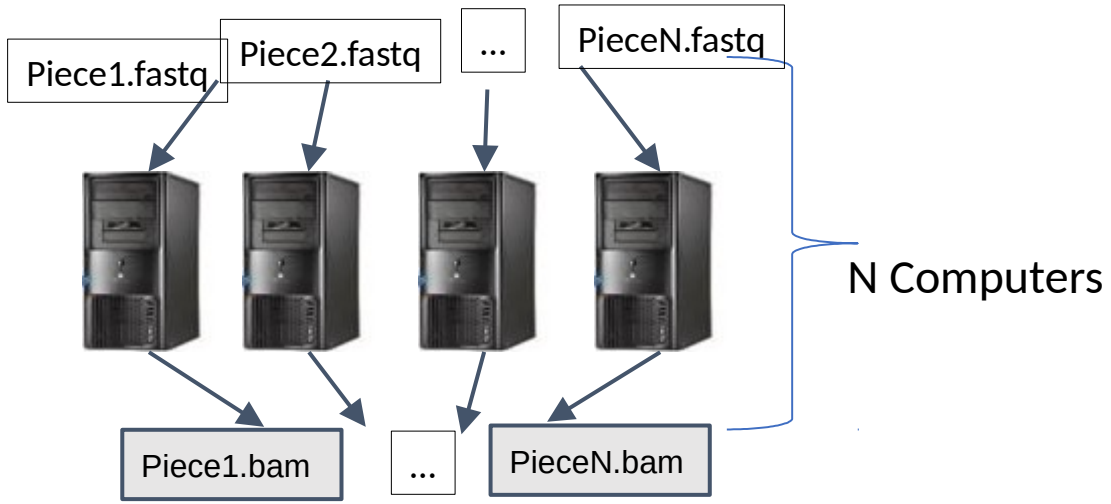
AS ALWAYS: The strategy is to:

- 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



N Computers

- **OR OR Utilise multiple processors on single computer**
  - Modern computers have >1 processing core or CPU
  - Most aligners can use more than one processor on same computer
  - Much easier for user
    - Just supply the number of processors to use (e.g. bwa-mem -t option)

Alignments – scaling up

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

ANSWER:
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 nodes, 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 WSI, alignment is more much more efficient (dedicated resources)

IT Costs of NGS

NGS generates a LOT of sequencing data

- HiSeq lane ~60 Gbp, X10 lane ~100 Gbp, MiSeq lane ~15 Gbp

Two main dimensions for estimating IT costs

- Compute - number of computers/server (CPUs) required to do data processing in a reasonable amount of time
- Storage - the physical disks that your sequencing data is stored on (including backup copies)

Estimating storage requirements

- BAM ~1 byte per bp sequenced
- 1 primary copy, 1 backup copy of raw data: 2 bytes per bp
- 1 processed/merged copy of the data: 1 byte per bp
- Output from variant calling programs: 1-2 bytes per bp
- 4-5 bytes per bp in total
- e.g. experiment will generate 10 HiSeq lanes of sequencing: 10 x 60 x 5 = 3000 Gbytes = 3 Tbytes

Estimating compute requirements

- More difficult to estimate as it depends on the type of analysis being carried out and the software being used
- Estimate 20-40 CPU hours per Gbp
- e.g. experiment will generate 10 HiSeq lanes of sequencing: 10 x 60 x 40 = 24,000 CPU hours

Overview

Intro

Methods / Aligners

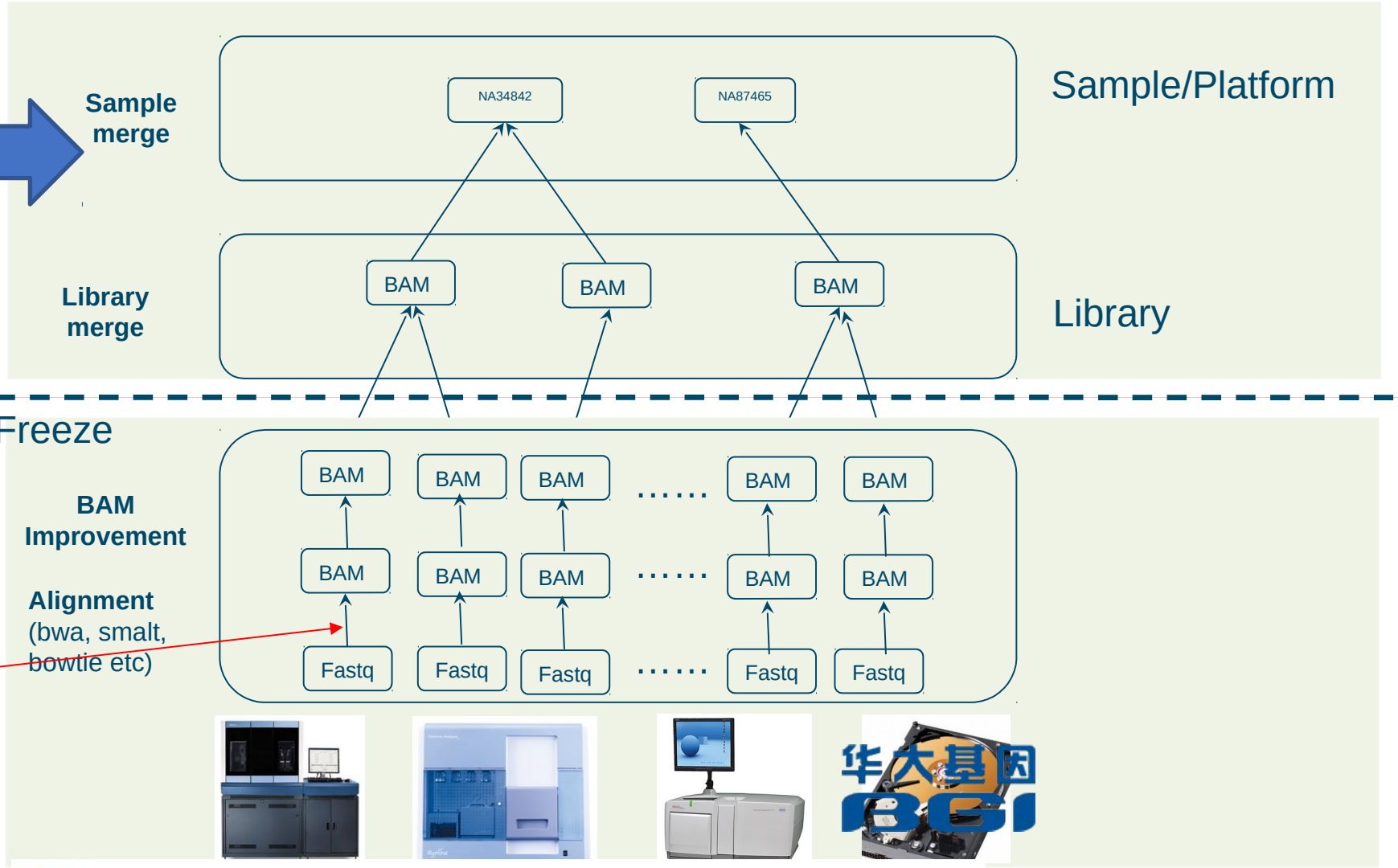
NGS Workflows, QC and BAM Improvement

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

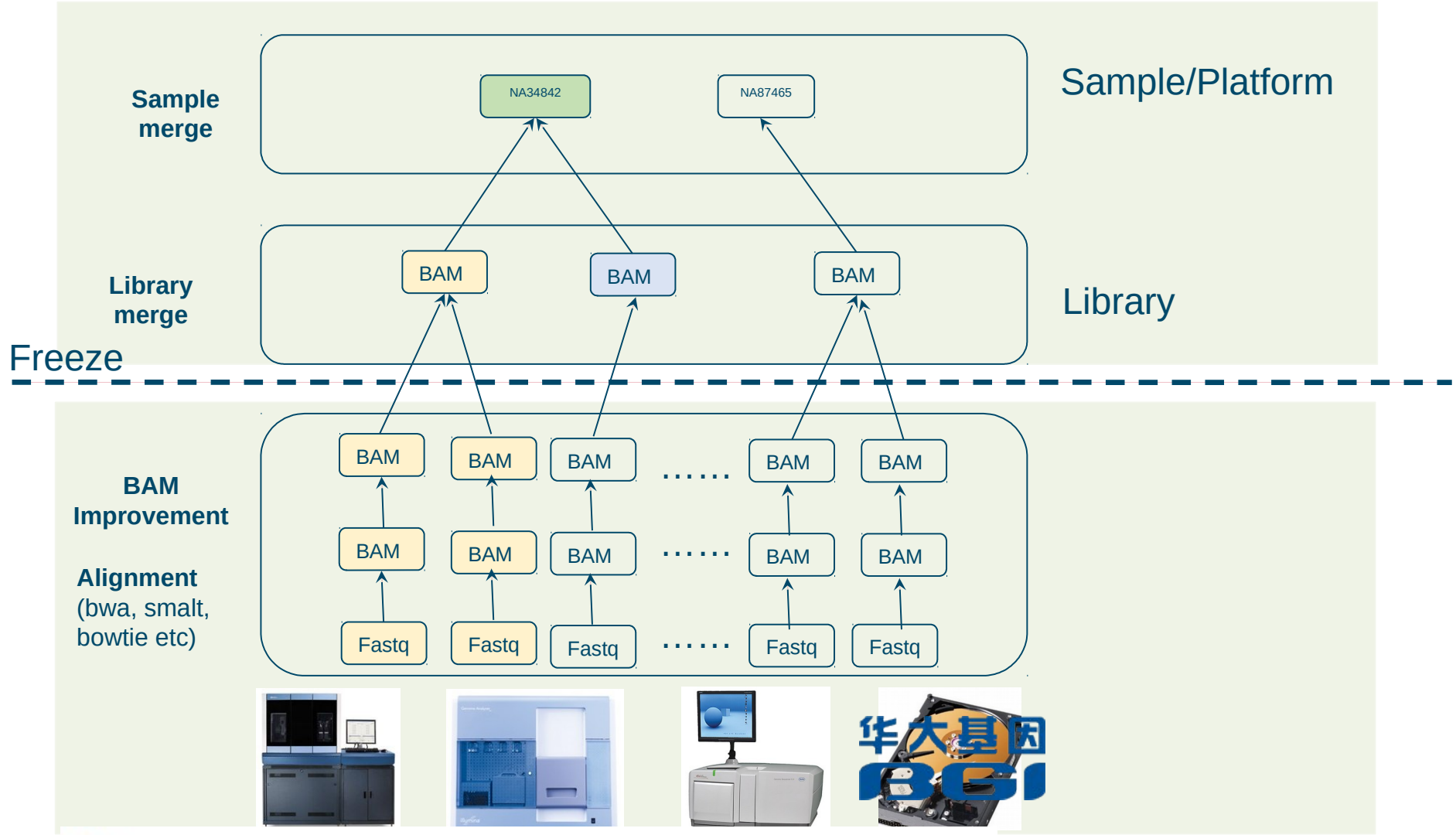


A typical NGS workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



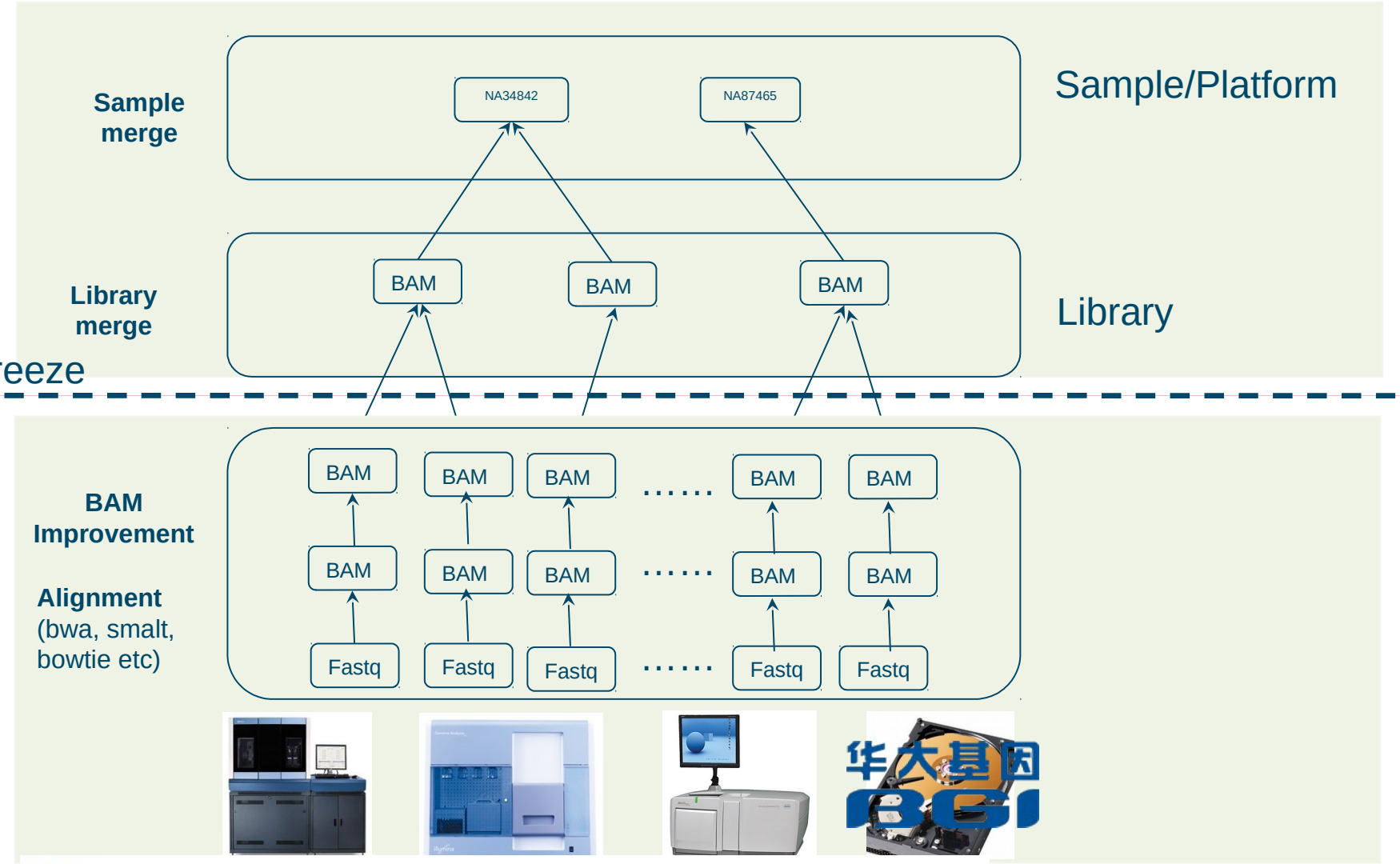
Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)

WHY?

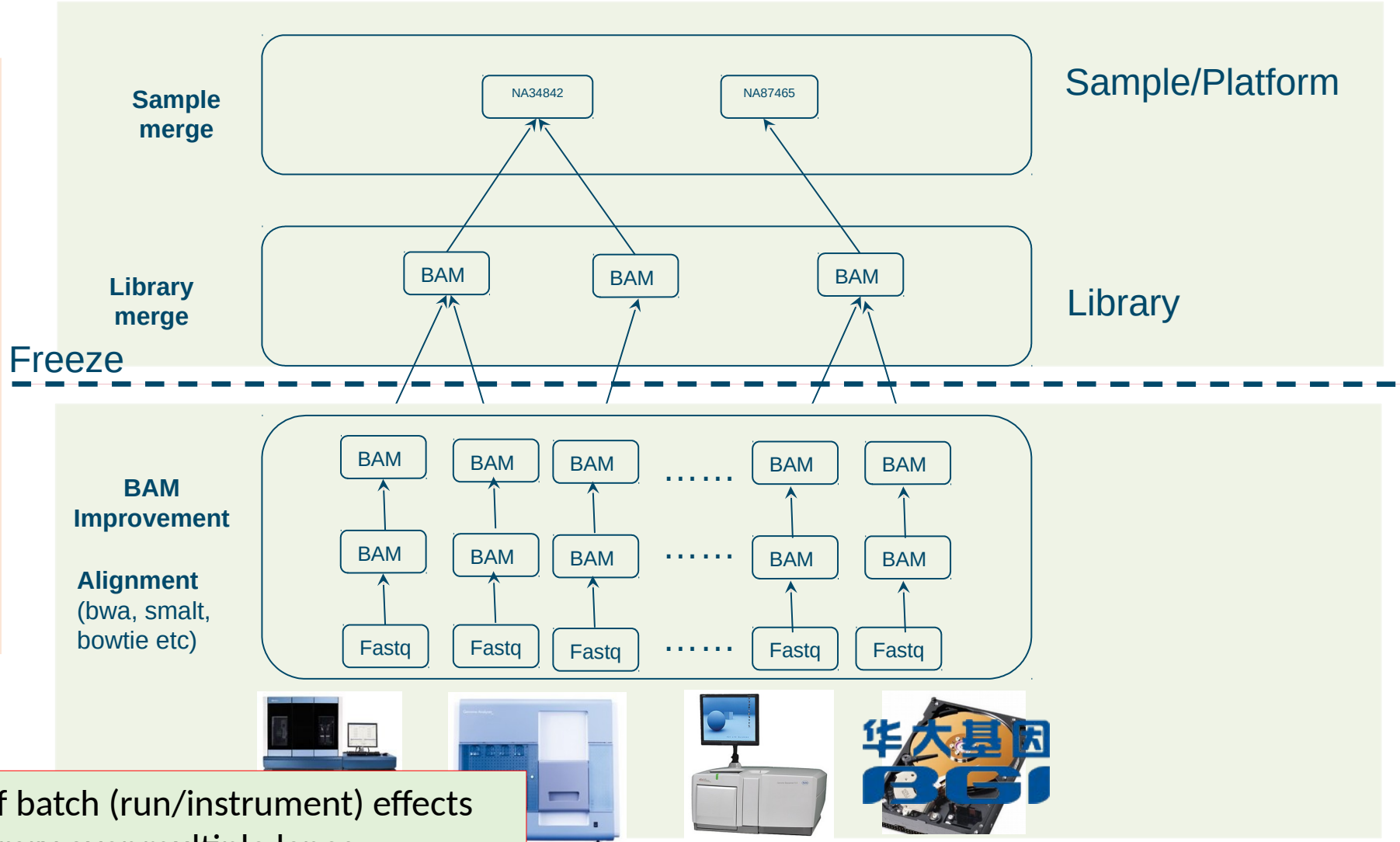


Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



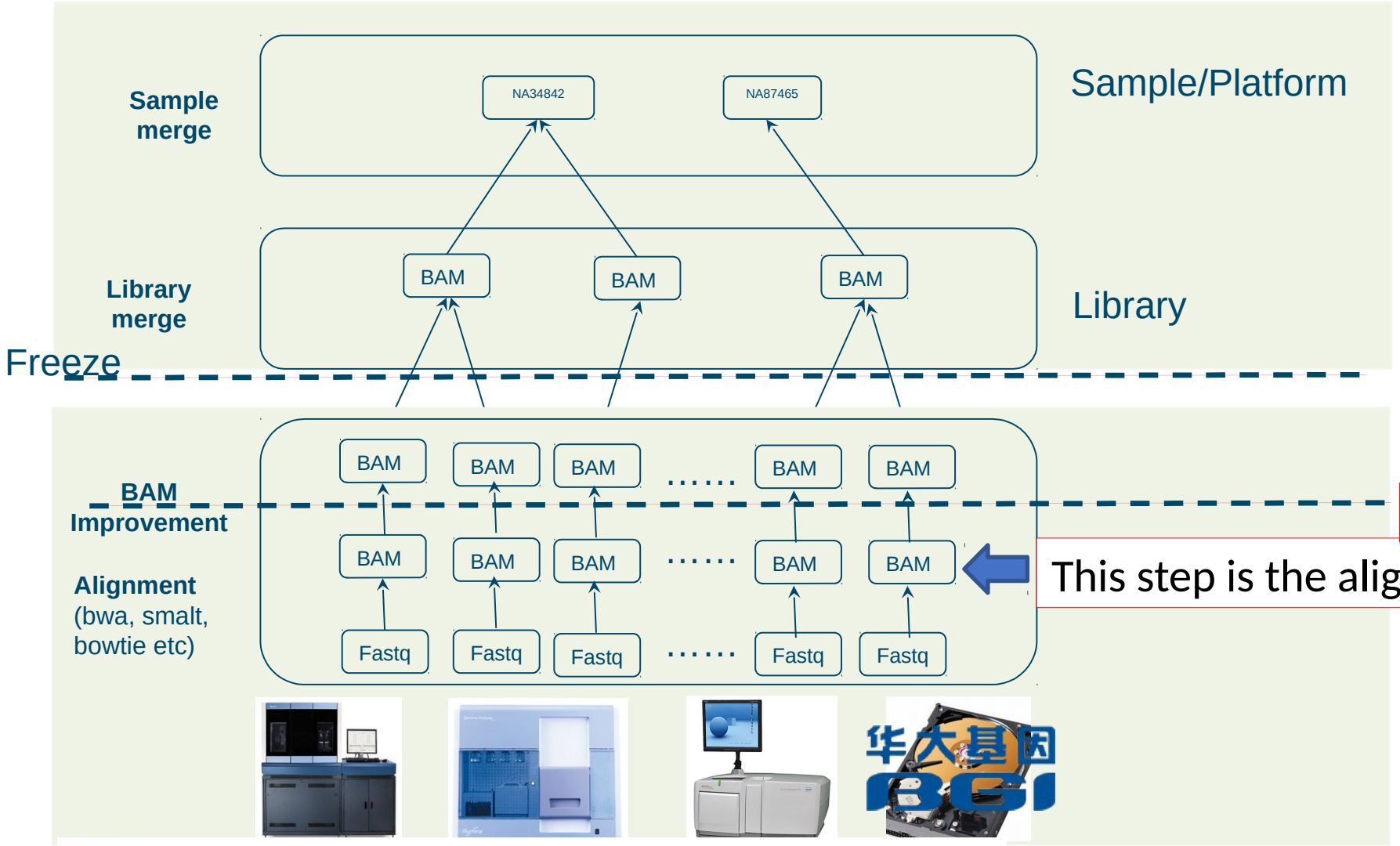
- Mitigation of batch (run/instrument) effects
  - Split library over multiple lanes
- Increasing sequencing depth (more libraries)

Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)

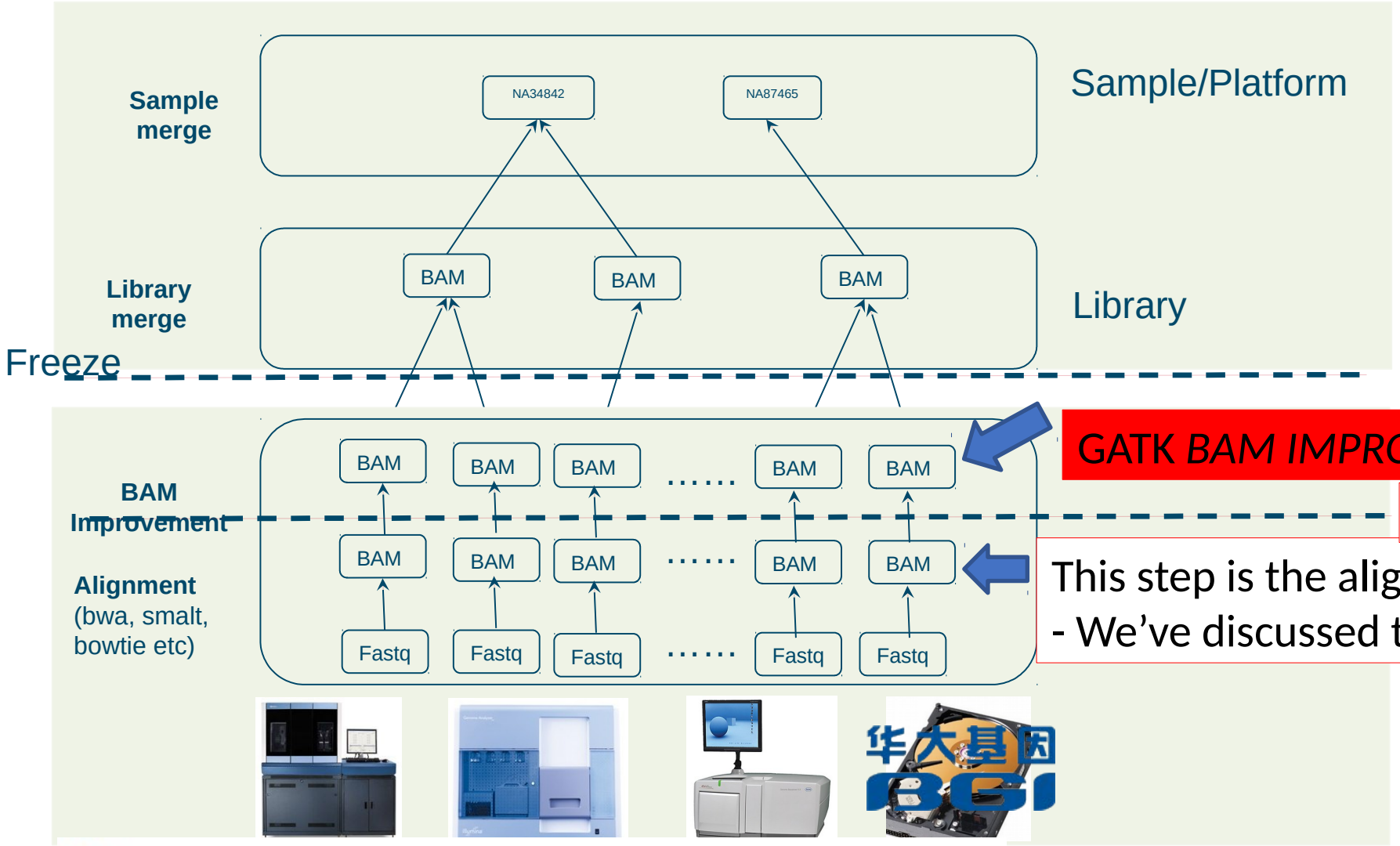


Data production workflow

Typically in a production workflow:

One sample spread over multiple seq libraries.

One library spread over multiple seq runs (lanes)



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 - Base Quality Score Recalibration

Each base call has an associated base call quality (find this in the fastq and sam record)

- 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

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 quality 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

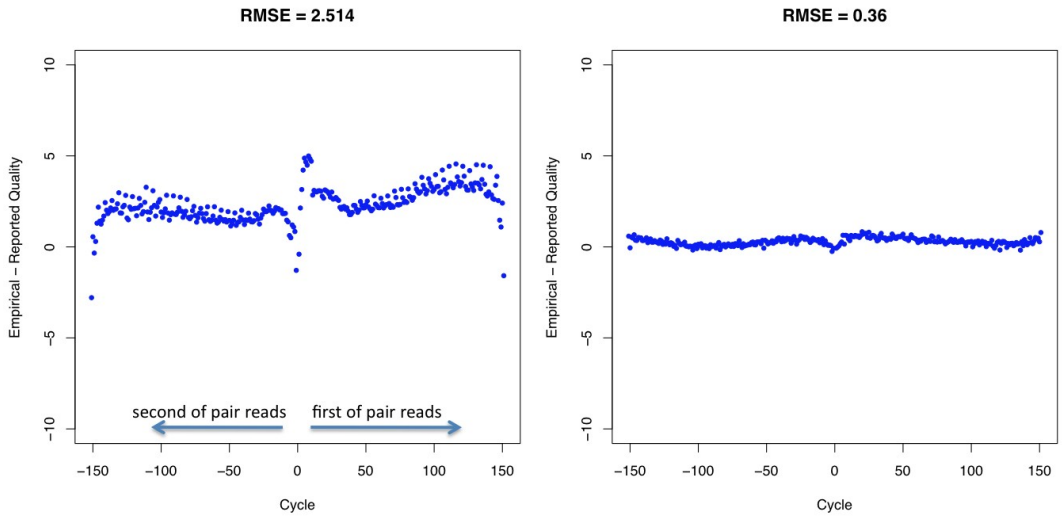
e.g. Reported: Sequencer reports Q25 base calls for a "T" in context "AT"

- Empirical: 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

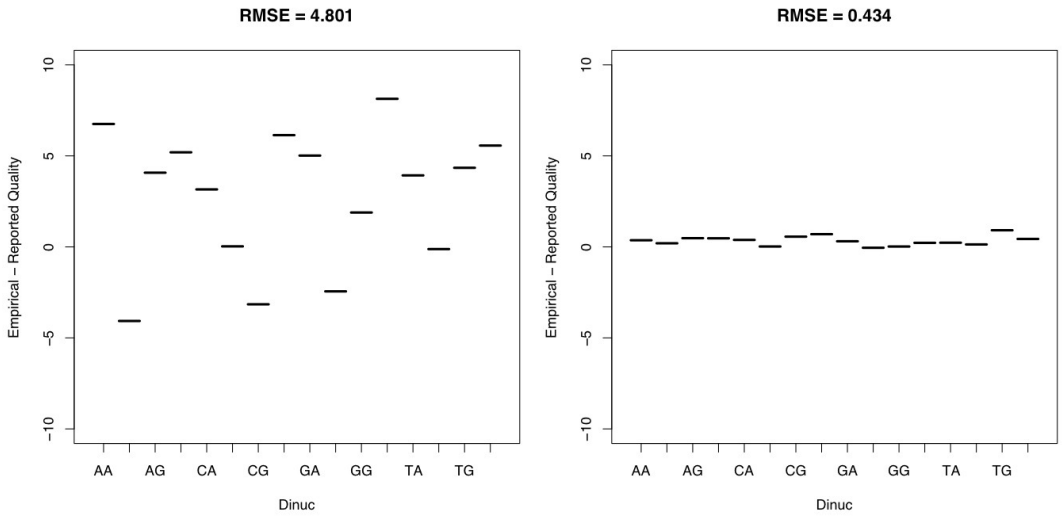
Residual Error by Machine Cycle



Original Data

After GATK Recalibration

Residual Error by Dinucleotide



Original Data

After GATK Recalibration

Data production workflow

One sample

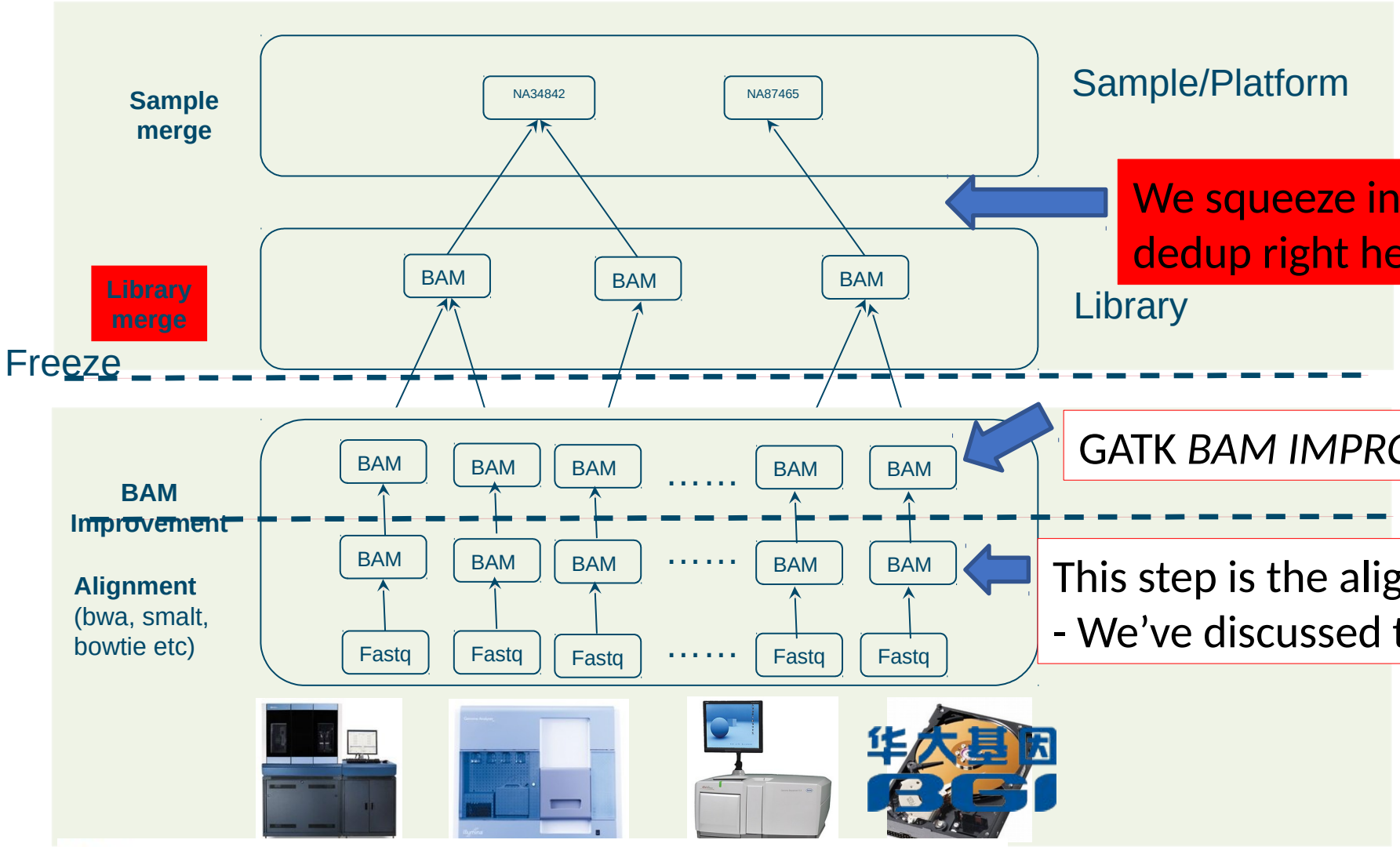
…spread over…

Multiple libraries

One library

…spread over…

Multiple runs (lanes)



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

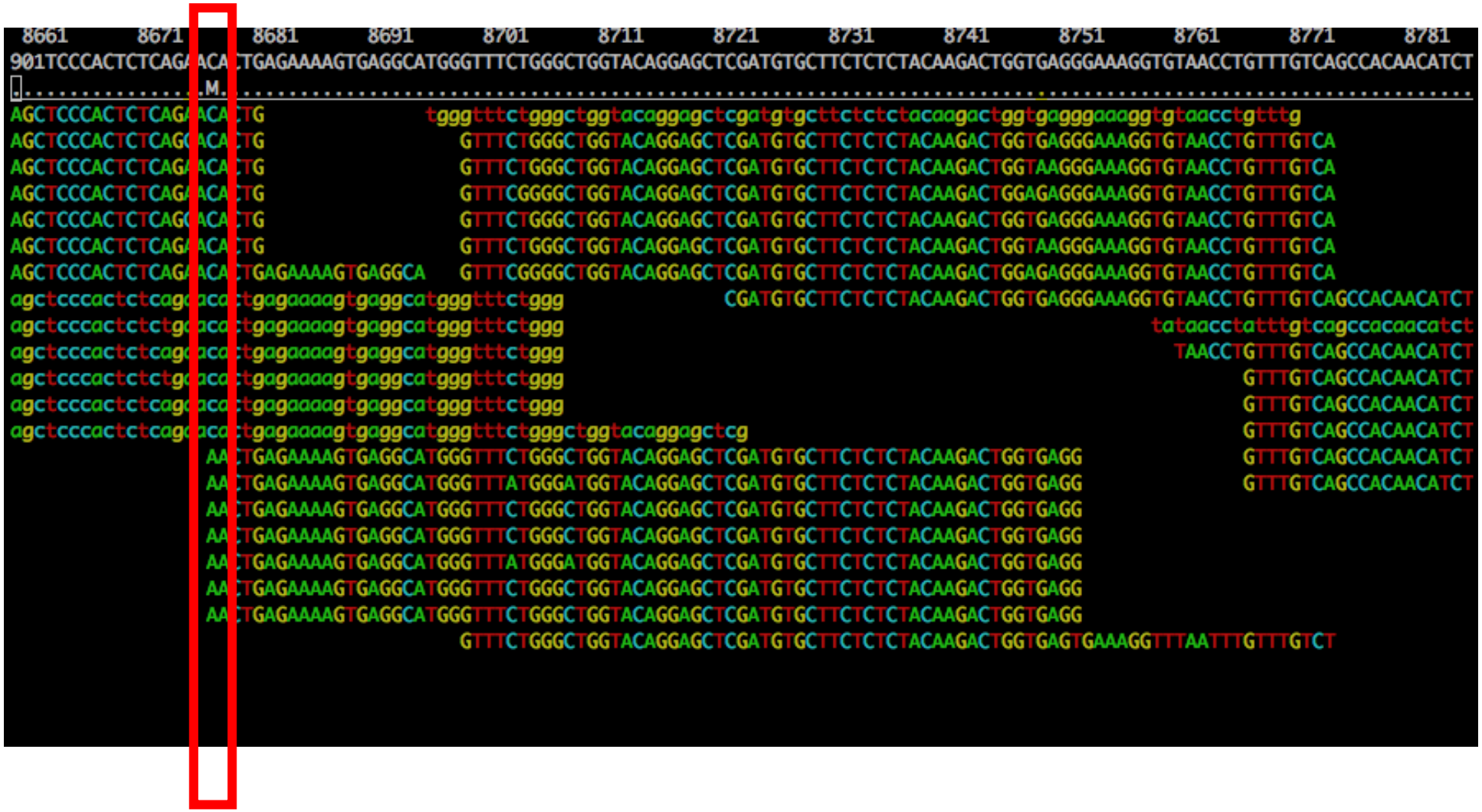
- Duplicates manifest themselves as high read depth support

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%*

Duplicates and False SNPs



Duplicates and False SNPs

Data production workflow

One sample

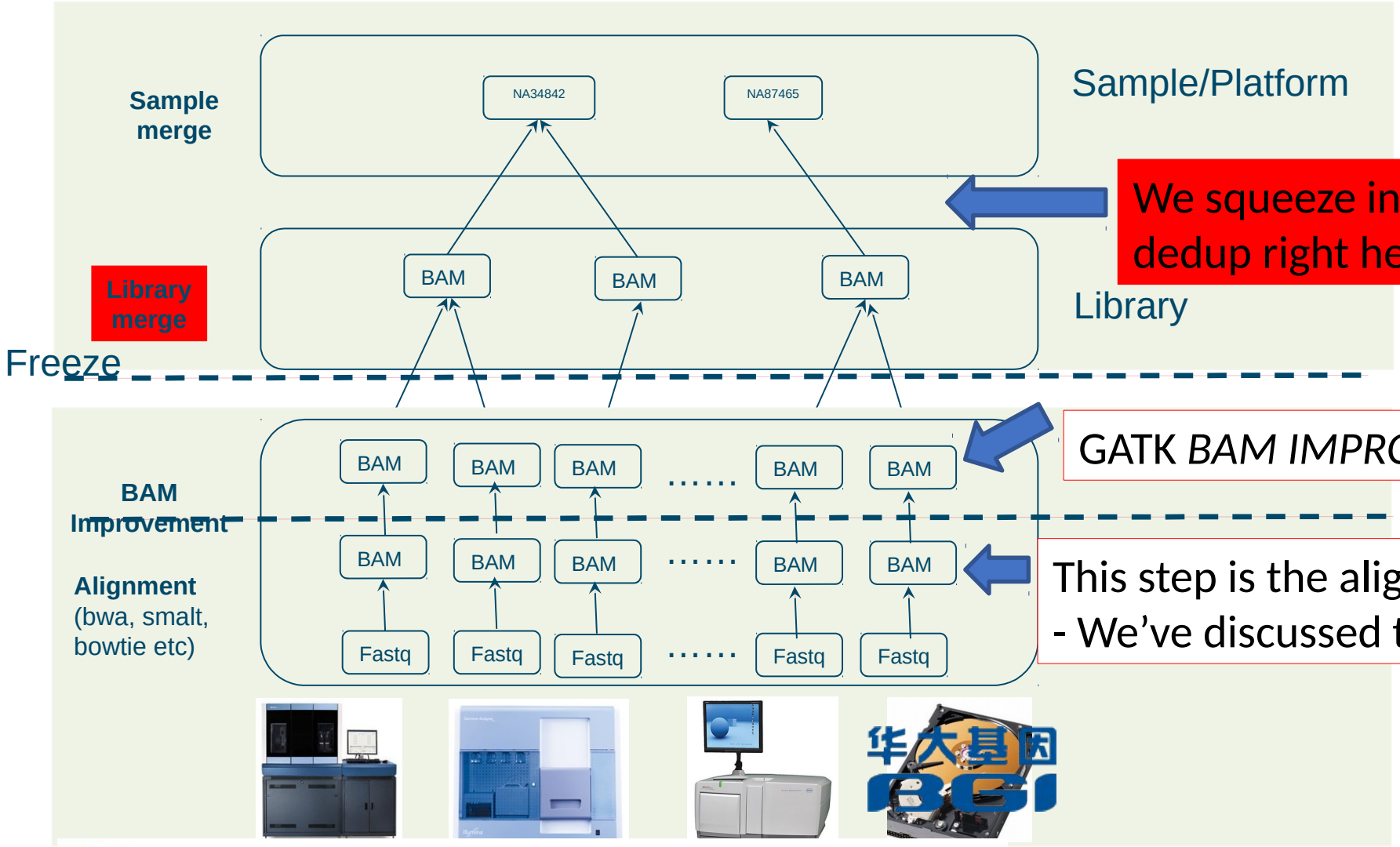
…spread over…

Multiple libraries

One library

…spread over…

Multiple runs (lanes)



Overview

Intro

Methods / Aligners

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.

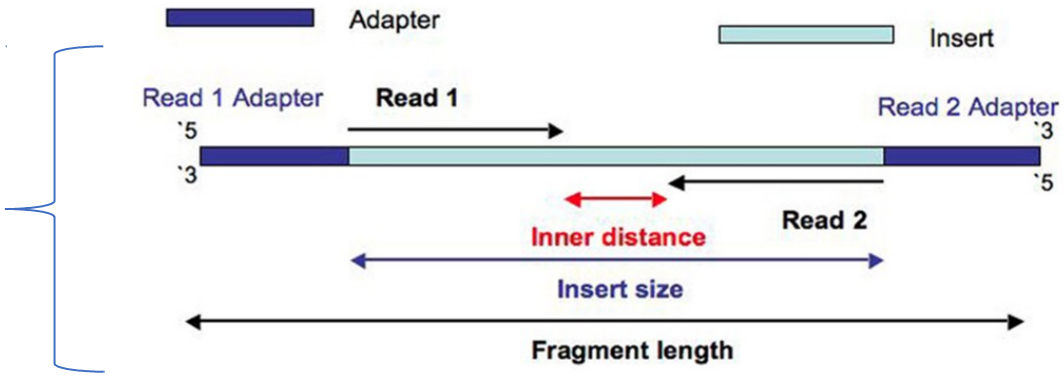
QUESTIONS?

FUNNY STORIES?

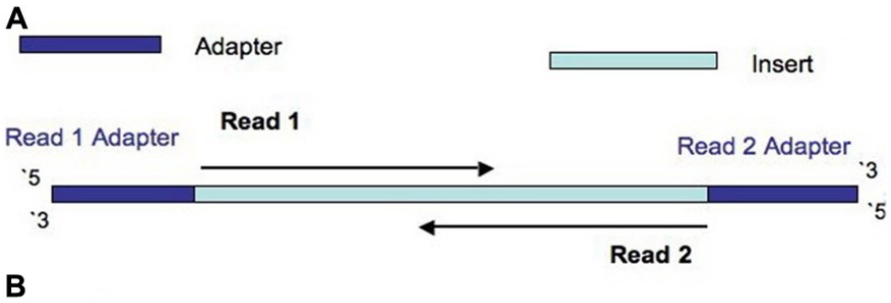
INTERLUDE – PAIRED END SEQUENCING PARAMETERS

Fragment length, insert size and inner mate distance

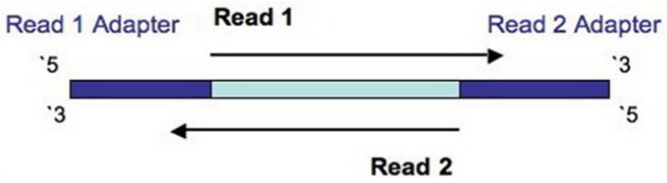
Insert size > length(read 1 + read 2)



Insert size < length(read 1 + read 2)



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

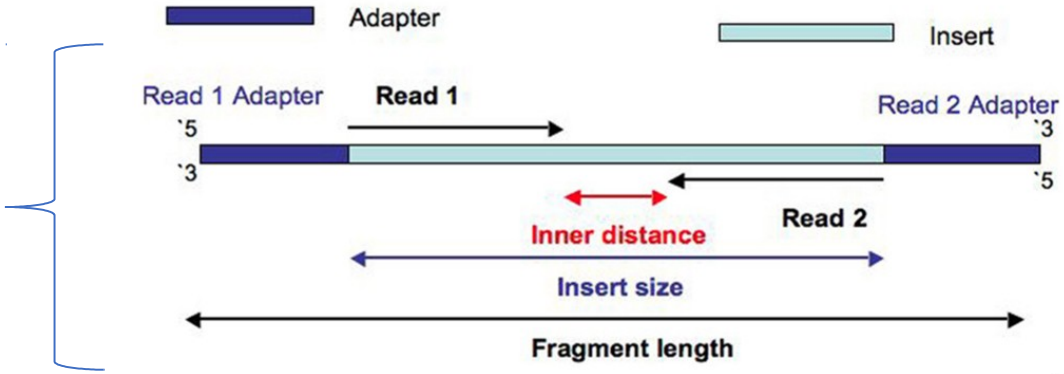


Turner, 2014. PMID:24523726

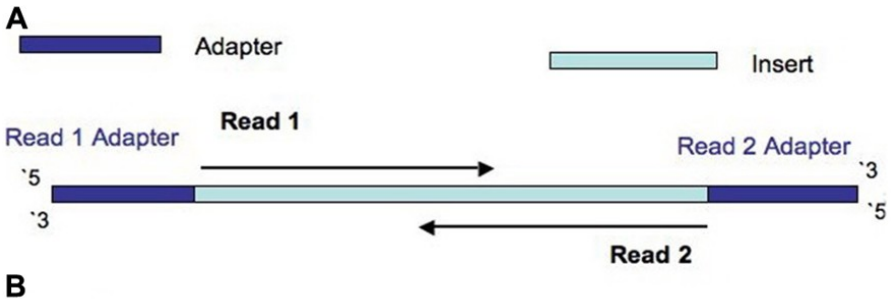
Fragment length, insert size and inner mate distance

Insert size > length(read 1 + read 2)

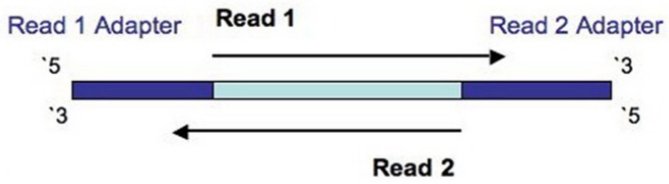
Both reads get the same "name"



Insert size < length(read 1 + read 2)



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



Turner, 2014. PMID:24523726

Overview

Intro

Methods / Aligners

NGS Workflows, QC and BAM Improvement