# Module 5: Read alignment

Presented by:
**Ximena Ibarra-Soria**

GSK

ximena.x.ibarra-soria@gsk.com
@xIbarraSoria

Based on materials by:
Anthony Doran and Vivek Iyer

**Next Generation Sequencing Bioinformatics Course**
18-22 January 2021 - Santiago - Chile

FACULTAD DE
CIENCIAS BIOLOGICAS
PONTIFICIA UNIVERSIDAD
CATÓLICA DE CHILE

WELLCOME GENOME CAMPUS
CONNECTING SCIENCE
ADVANCED COURSES+ SCIENTIFIC CONFERENCES

# NGS read alignment

- Introduction to read alignment.
- Main methods and aligners.
- Visualisation of aligned reads.
- NGS workflows, QC and BAM improvement.

# NGS read alignment

- Introduction to read alignment.
- Main methods and aligners.
- Visualisation of aligned reads.
- NGS workflows, QC and BAM improvement.

# NGS read alignment

- Next generation sequencing (NGS) generates millions of short reads.

- **Read alignment is the process of determining the most likely source of the read within a reference genome sequence.**

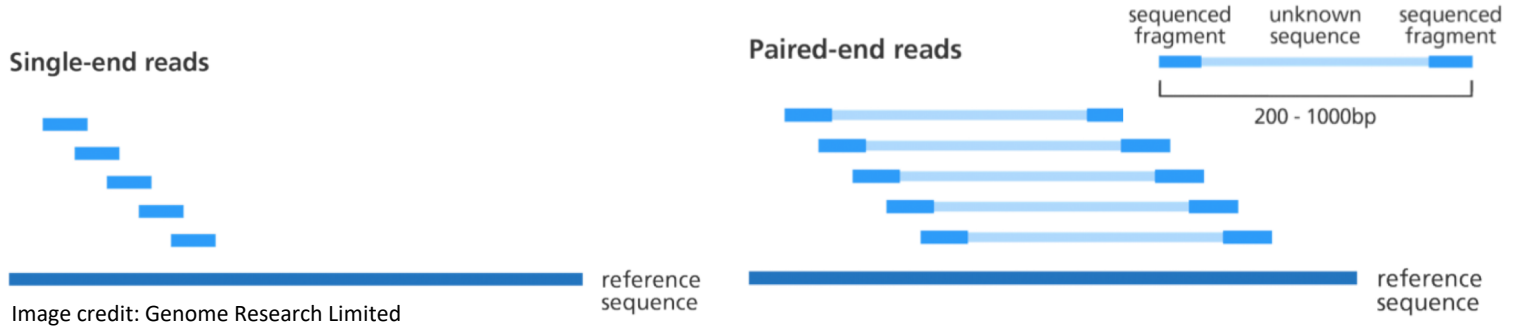



Image credit: Genome Research Limited

Flicek and Birney, *Nat. Methods* 6 (2009) doi.org/10.1038/nmeth.1376

# NGS read alignment

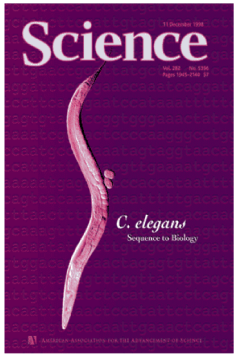
- Next generation sequencing (NGS) generates millions of short reads.

- **Read alignment is the process of determining the most likely source of the read within a reference genome sequence.**

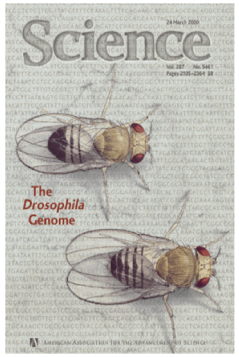
- Challenges:
  - Millions of short reads.
  - Very large search space.
  - Need to allow mismatches/indels (both PCR and sequencing errors; variation).

Flicek and Birney, *Nat. Methods* 6 (2009) doi.org/10.1038/nmeth.1376

# Reference genomes

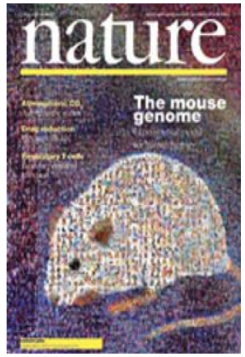



1998




2000


2001




2002


2004


2005

- Reference genomes of model organisms were generated using Sanger sequencing.
- Collaborative projects involving many institutes from different countries.
  - Sequencing and assembly.
  - Curation and annotation.
  - Distribution.

Initially done by NCBI.
Taken over by the Genome Reference Consortium.

# Reference genomes

human

| Release name | Date of release | UCSC equivalent |
|---|---|---|
| GRCh38 | Dec 2013 | hg38 |
| GRCh37 | Feb 2009 | hg19 |
| NCBI Build 36.1 | Mar 2006 | gh18 |
| NCBI Build 35 | May 2004 | hg17 |
| NCBI Build 34 | Jul 2003 | hg16 |

mouse

| Release name | Date of release | UCSC equivalent |
|---|---|---|
| GRCm39 | Jun 2020 | mm39 |
| GRCm38 | Dec 2011 | mm10 |
| NCBI Build 37 | Jul 2007 | mm9 |
| NCBI Build 36 | Feb 2006 | mm8 |
| NCBI Build 35 | Aug 2005 | mm7 |

- Reference genomes are constantly improved.
- When additional sequence becomes available to close gaps a **new assembly** is released.
- Alignment results depend on the assembly used (coordinates change).

# Why do we need to align reads

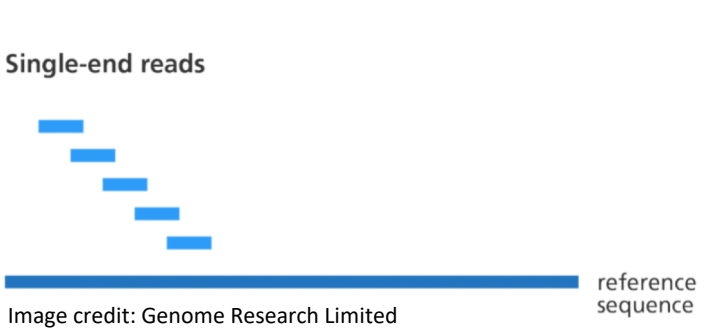
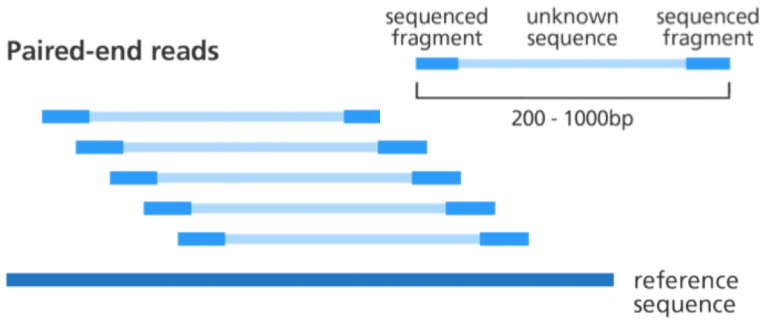
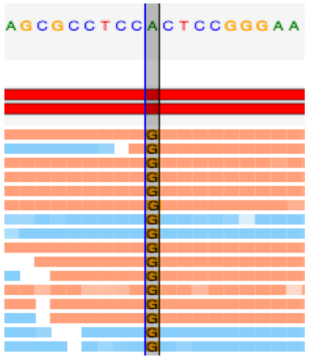


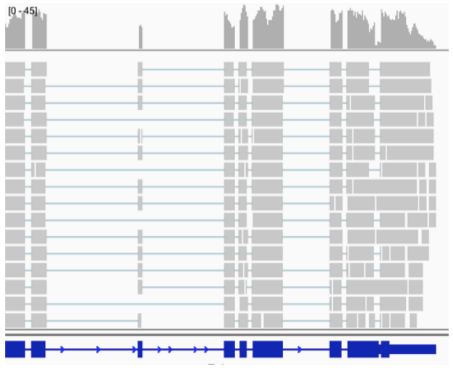

Image credit: Genome Research Limited



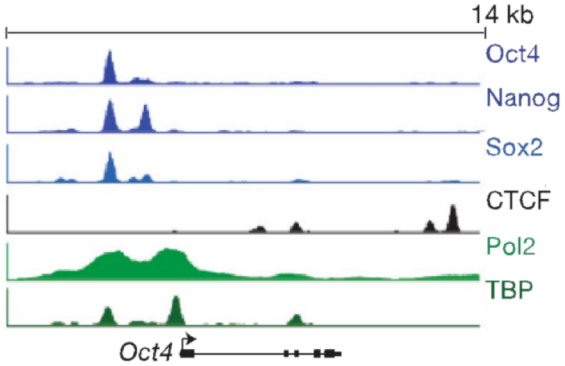

identify variation




measure transcript abundance



identify regions bound by TFs



# NGS read alignment

- Introduction to read alignment.
- Main methods and aligners.
- Visualisation of aligned reads.
- NGS workflows, QC and BAM improvement.

# Smith-Waterman algorithm (1981)

- Algorithm to **identify the *optimal* pairwise alignment** between two sequences.
  - optimal = the most correspondences and least differences
- Pairwise comparisons between letters are scored based on a **substitution matrix and gap penalties**.
- High sensitivity but **slow**.

# Smith-Waterman algorithm (1981)

- For example, if we have the following sequences we can align them by eye.

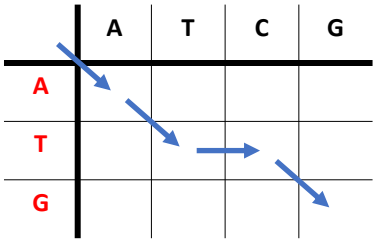
**REFERENCE SEQUENCE:**
ATCG

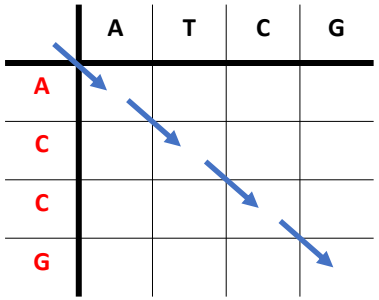
**QUERY SEQUENCES:**
ATG ACCG

**ALIGNMENT:**
ATCG
|| |
AT–G

**ALIGNMENT:**
ATCG
||||
ACCG

Traverse the grid. Moving
- diagonally match/mismatch
- horizontally/vertically gap



# Smith-Waterman algorithm (1981)

| | | A | T | C | G |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 |
| A | 0 | -2,-2,3<br>3 | 1,-2,-3<br>1 | -1,-2,-3<br>-1=0 | -2,-2,-3<br>-1=0 |
| T | 0 | -2,1,-3<br>1 | | | |
| G | 0 | | | | |

Enter a cell:

- from the **left**
- from the **top**

} gap

- from **diagonal** - match/mismatch

Scoring:

- match +3
- mismatch -3
- gap -2

Enter each cell and score, adding to the previous cell score.

Retain the largest score* along with the entry point that generated it.

* If largest score is negative replace with 0.

# Smith-Waterman algorithm (1981)

| | | A | T | C | G |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 |
| A | 0 | -2,-2,3<br>3 | 1,-2,-3<br>1 | -1,-2,-3<br>-1=0 | -2,-2,-3<br>-1=0 |
| T | 0 | -2,1,-3<br>1 | -1,-1,6<br>6 | 4,-2,-2<br>4 | 2,-2,-3<br>2 |
| G | 0 | -2,-1,-3<br>-1=0 | -2,4,-2<br>4 | 2,2,3<br>3 | 1,0,7<br>7 |

Enter a cell:

- from the **left**
- from the **top**

} gap

- from **diagonal** - match/mismatch

Scoring:

- match +3
- mismatch -3
- gap -2

Enter each cell and score, adding to the previous cell score.

Retain the largest score* along with the entry point that generated it.

* If largest score is negative replace with 0.

# Smith-Waterman algorithm (1981)

| | | A | T | C | G |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 |
| A | 0 | -2,-2,3<br>3 | 1,-2,-3<br>1 | -1,-2,-3<br>-1=0 | -2,-2,-3<br>-1=0 |
| T | 0 | -2,1,-3<br>1 | -1,-1,6<br>6 | 4,-2,-2<br>4 | 2,-2,-3<br>2 |
| G | 0 | -2,-1,-3<br>-1=0 | -2,4,-2<br>4 | 2,2,3<br>3 | 1,0,7<br>7 |

A T C G
| | | |
A T - G

To find the optimal alignment:

- Start at the highest scoring cell.
- **Traceback** the alignment with the longest score.

Entry from left or top = gap.

Entry from diagonal = match/mismatch.

# Alignment of NGS data

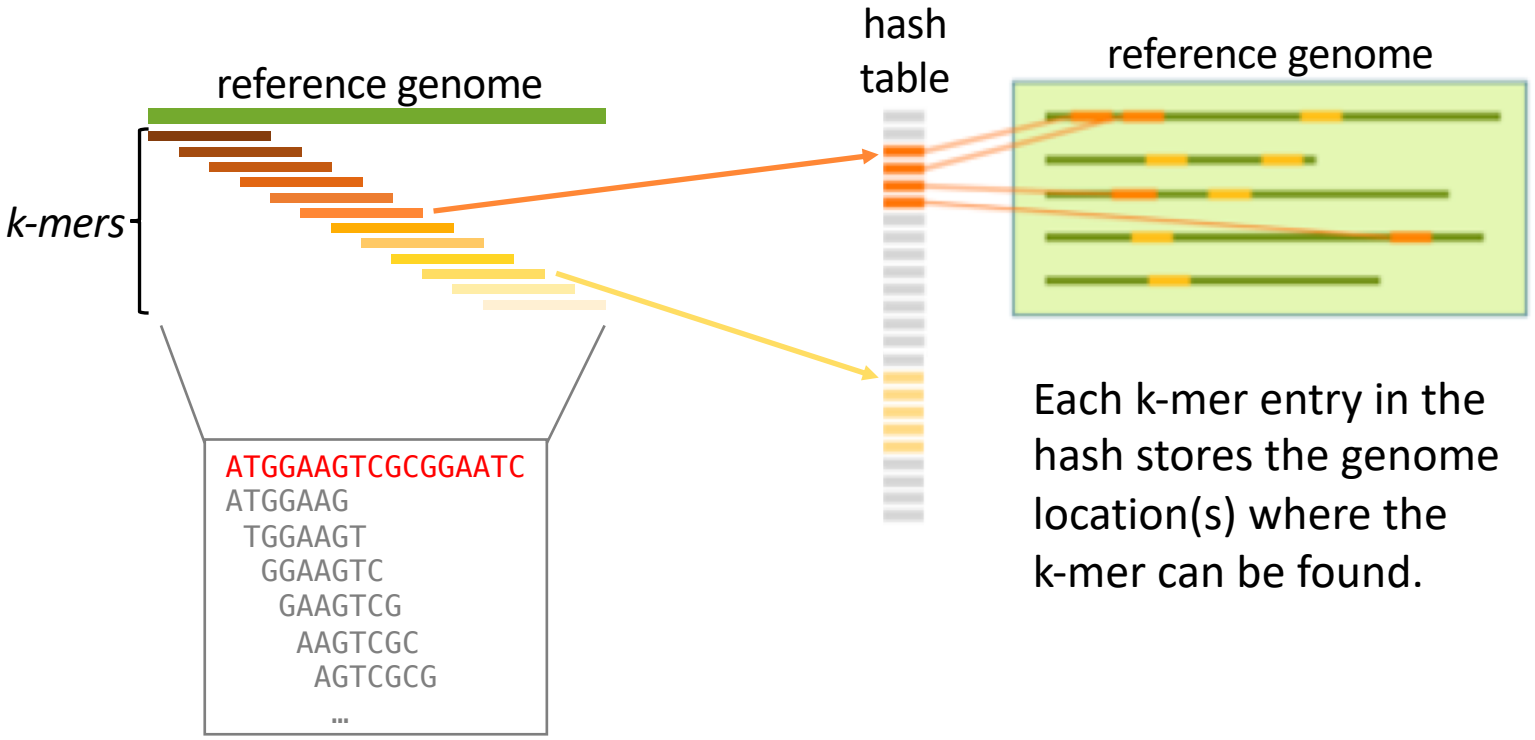
- **Challenges** of aligning NGS data:
- There are tens to hundreds of millions of sequencing reads to be mapped.
- The search space (reference genome) can be very large.
  - Human haploid genome: 3,234.83 Mb
  - Mouse haploid genome:  2,653.99 Mb

# Alignment of NGS data

- **Solutions** to these problems use two basic principles.
- **Filtering**: quickly exclude large regions of the reference where matches cannot be found.
  - Take a substring of the read (seed) and find perfect matches.
- **Indexing**: involves pre-processing the reference to speed-up matching without having to scan the whole reference.
  - Hash tables.
  - Suffix/prefix trees.
  - FM indices with Burrows-Wheeler transform.

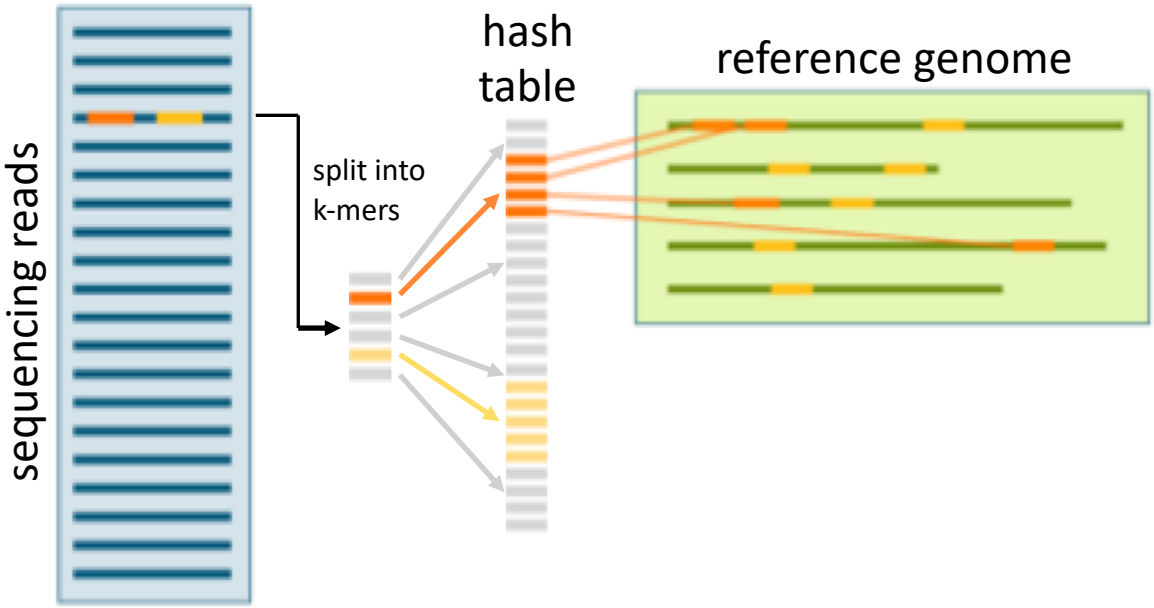
# Hash tables

- Hash table = index of all locations of a k-mer in a sequence.
  - k-mer = short fixed sequence, *k* nucleotides long.



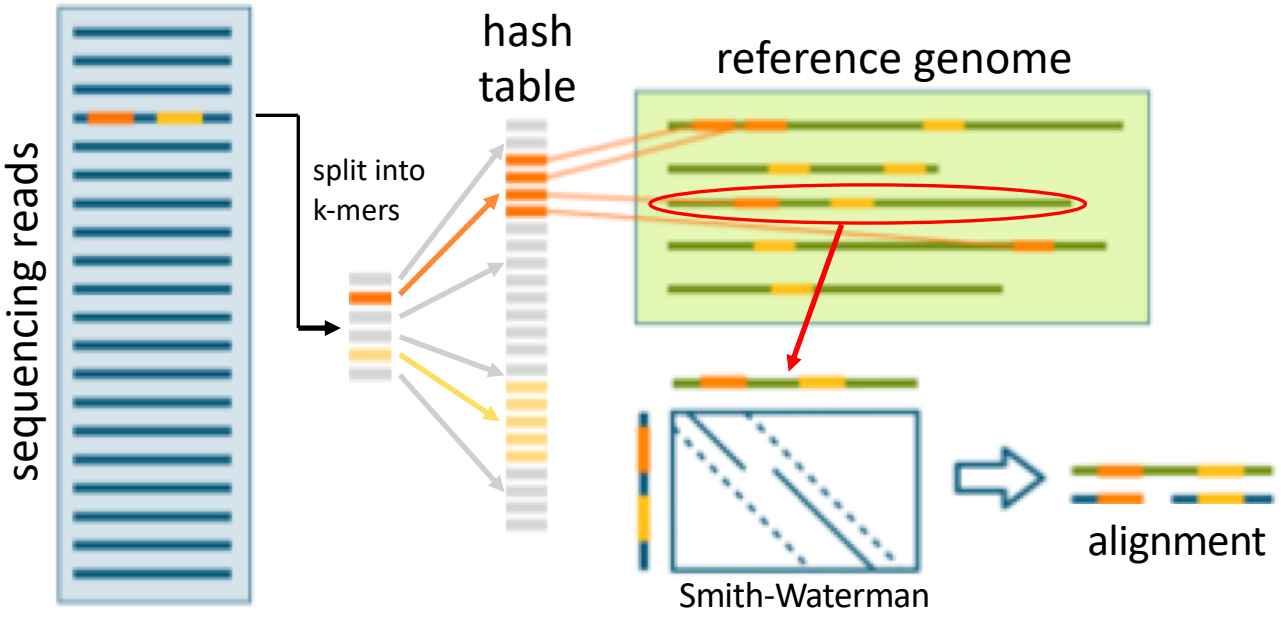
# Hash tables

- To align reads:
  - Split into k-mers
  - Look up in the hash table -> possible locations of the read in the genome.



# Hash tables

- Smith-Waterman is used to extend the possible alignments and find the best one.



# Suffix trees

- A **suffix tree** stores all the suffixes of a string.
  - It contains the **locations** where the subsequences exist.
  - Allows very **fast** string matching.
  - But it requires a lot of memory.
- The **FM index** retains the speed of a suffix tree but requires much less memory.
  - Uses Burrows-Wheeler transform (BWT) sequence instead.

    BWT reorders the genome such that sequences that exist multiple times appear together in the data structure.

Flicek and Birney, *Nat. Methods* 6 (2009) doi.org/10.1038/nmeth.1376

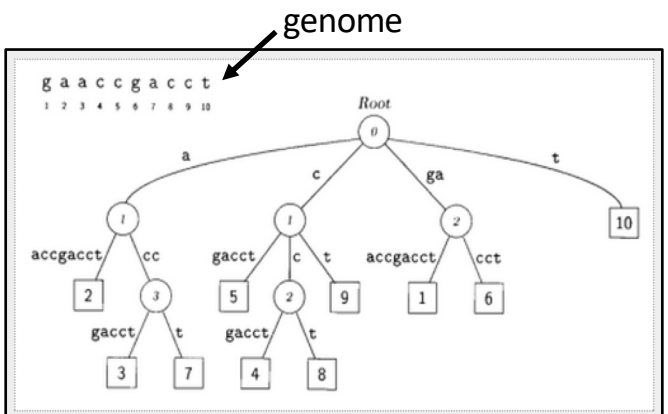



**Figure 2**
Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. In this example the longest repeated sequence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

Delcher et al., *Nucleic Acids Res.* 1 (1999) PMID:10325427

# Alignment limitations

- Correctly aligning around indels is hard.
  - Incorrect SNPs are often added around indels.
  - Smith-Waterman prefers a SNP over opening a gap.
- Regions with high density of SNPs.
  - Some aligners have an upper bound of allowed mismatches.
- Sequencing reads are short and the genome is complex.
  - Low-complexity regions.
  - High content of repeated sequences (repeat elements, paralogs, etc).
  - Determining the correct location is hard.

# Mapping quality

- **Mapping quality:** measures the confidence of the alignment by considering all possible locations discovered.

$p_{cor}$ = probability alignment is correct

$Q = -10 \log_{10}(1-p_{cor})$

| | |
|---|---|
| Q0 | read placed with identical score elsewhere = **multi-mapped**. |
| Q10 | 1 in 10 chance the alignment is wrong. |
| Q20 | 1 in 100 chance the alignment is wrong. |

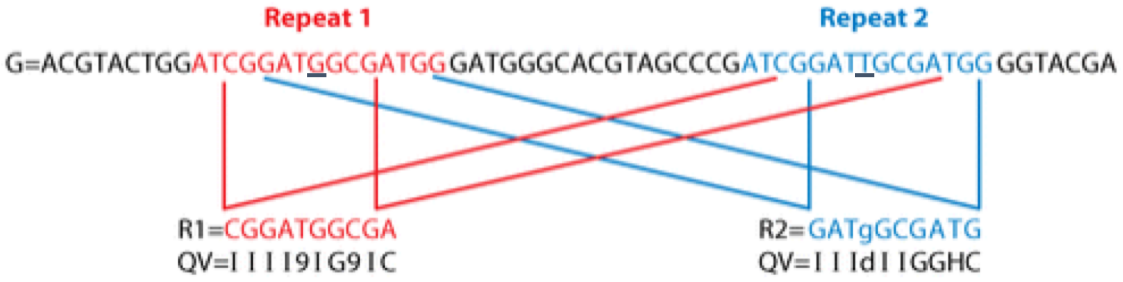
# Mapping quality

- **Mapping quality:** measures the confidence of the alignment by considering all possible locations discovered.

  $p_{cor}$ = probability alignment is correct

  $Q = -10\ log_{10}(1-p_{cor})$

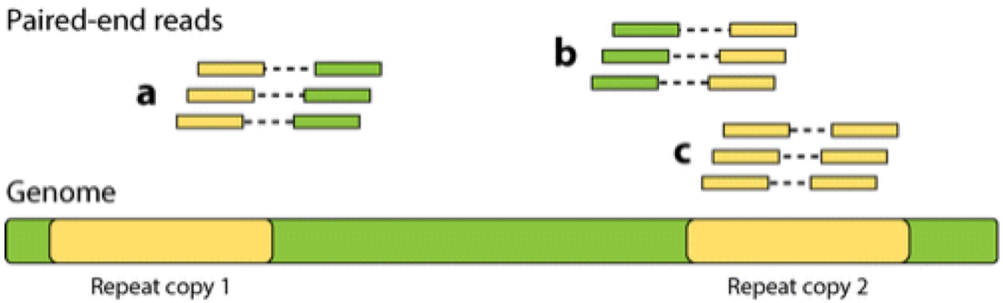
  Q0 read placed with identical score elsewhere = **multi-mapped**.




Reinert et al., *Annu. Rev. Genomics Hum. Genet.* 16 (2015) doi.org/10.1146/annurev-genom-090413-025358

# Mapping quality

- **Mapping quality:** measures the confidence of the alignment by considering all possible locations discovered.
- One way to lessen the burden of multi-mapping reads is to use paired-end reads.

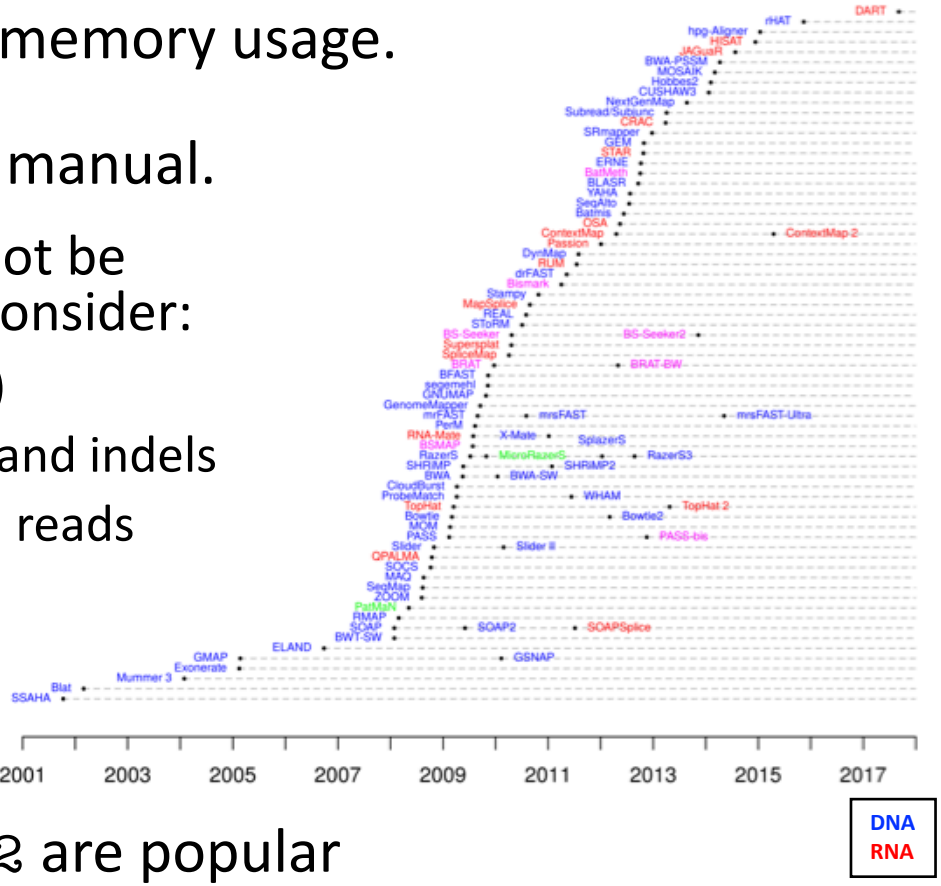



Reinert et al., *Annu. Rev. Genomics Hum. Genet.* 16 (2015) doi.org/10.1146/annurev-genom-090413-025358

# Aligners

- There are many aligners available.
- Differences in speed and memory usage.
- Read the publication and manual.

  The default settings might not be appropriate for your data. Consider:
  - type of data (DNA vs RNA)
  - penalties for mismatches and indels
  - handling of multi-mapped reads
  - read length
  - mapping paired-end reads




- For DNA, `bwa` and `bowtie2` are popular choices.

https://www.ebi.ac.uk/~nf/hts_mappers/

# Parallelisation

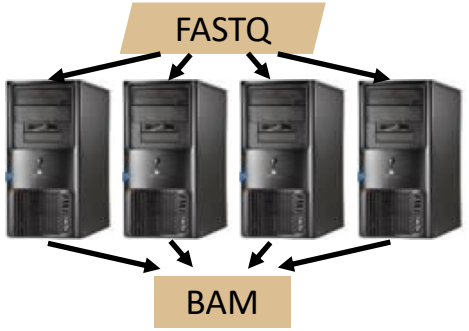
- One HiSeq X10 lane produces 150Gb of data.
- BWA-mem aligns 50Mb in 5 CPU minutes.


150Gb / 50Mb = $150 \times 10^9 / 50 \times 10^6 = 3 \times 10^3$ 50Mb chunks per HiSeq X10 lane.

5 min x $3 \times 10^3$ = 15,000 CPU minutes

= 250 hours = **10.4 days**

- Solution: use parallel computing.
  - Split the input into *N* pieces.
  - Process each piece **in parallel**.
  - Join the results together.




- In a cluster with 200 nodes, one lane can be aligned in ~1.25 hrs.

# NGS read alignment

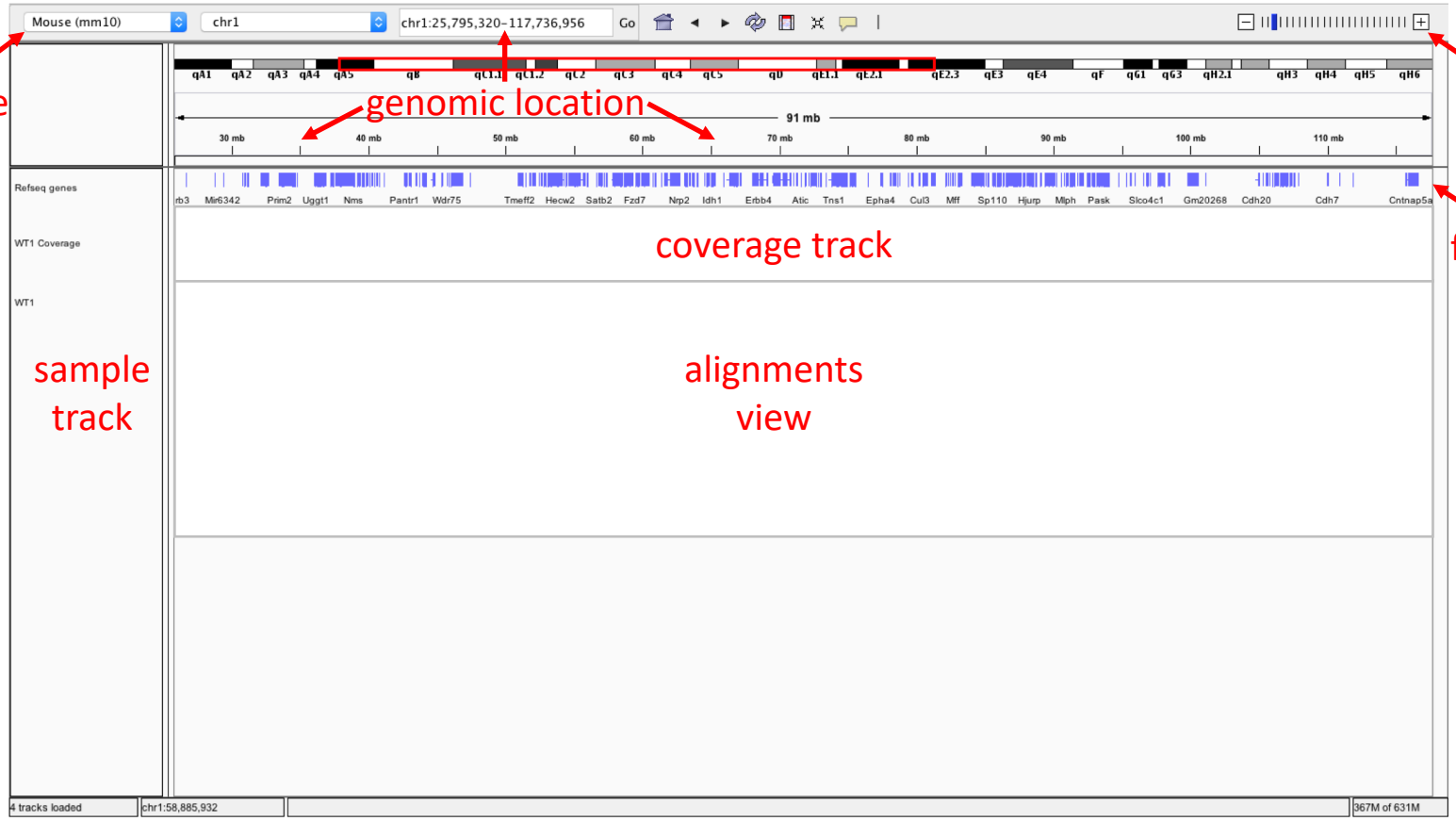
- Introduction to read alignment.
- Main methods and aligners.
- Visualisation of aligned reads.
- NGS workflows, QC and BAM improvement.

# Visualisation of alignments

- Once data is aligned we can perform downstream analyses.
  - Variant calling (genome/exome DNA-seq).
  - Peak calling (ChIP-seq, ATAC-seq).
  - Differential expression (RNA-seq).

- It is important to *see* the alignments.
  - Verify that the alignment was successful.
  - Detect artefacts or specific problems with the data.

# Integrative Genomics Viewer (IGV)




http://software.broadinstitute.org/software/igv/

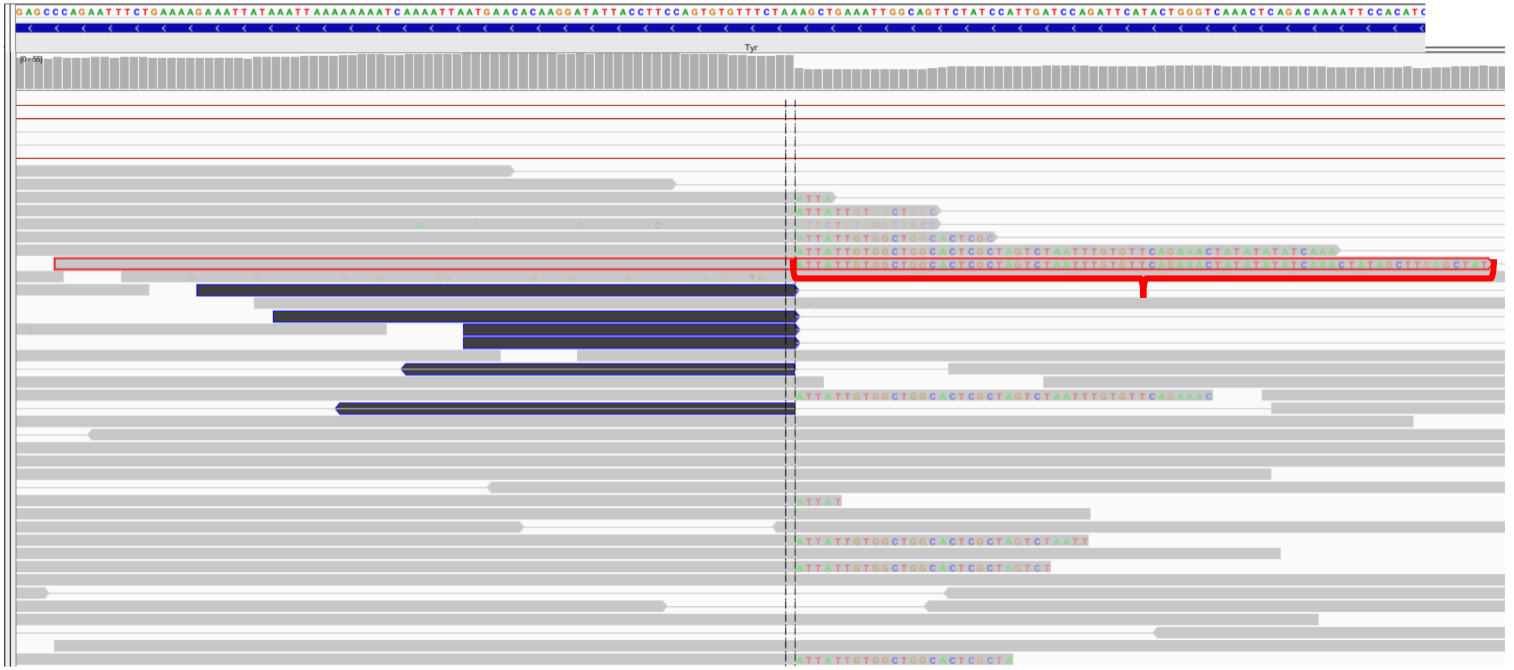
# Integrative Genomics Viewer (IGV)


http://software.broadinstitute.org/software/igv/

# Integrative Genomics Viewer (IGV)


http://software.broadinstitute.org/software/igv/

# Integrative Genomics Viewer (IGV)




soft-clipped bases do not align

http://software.broadinstitute.org/software/igv/

# Integrative Genomics Viewer (IGV)

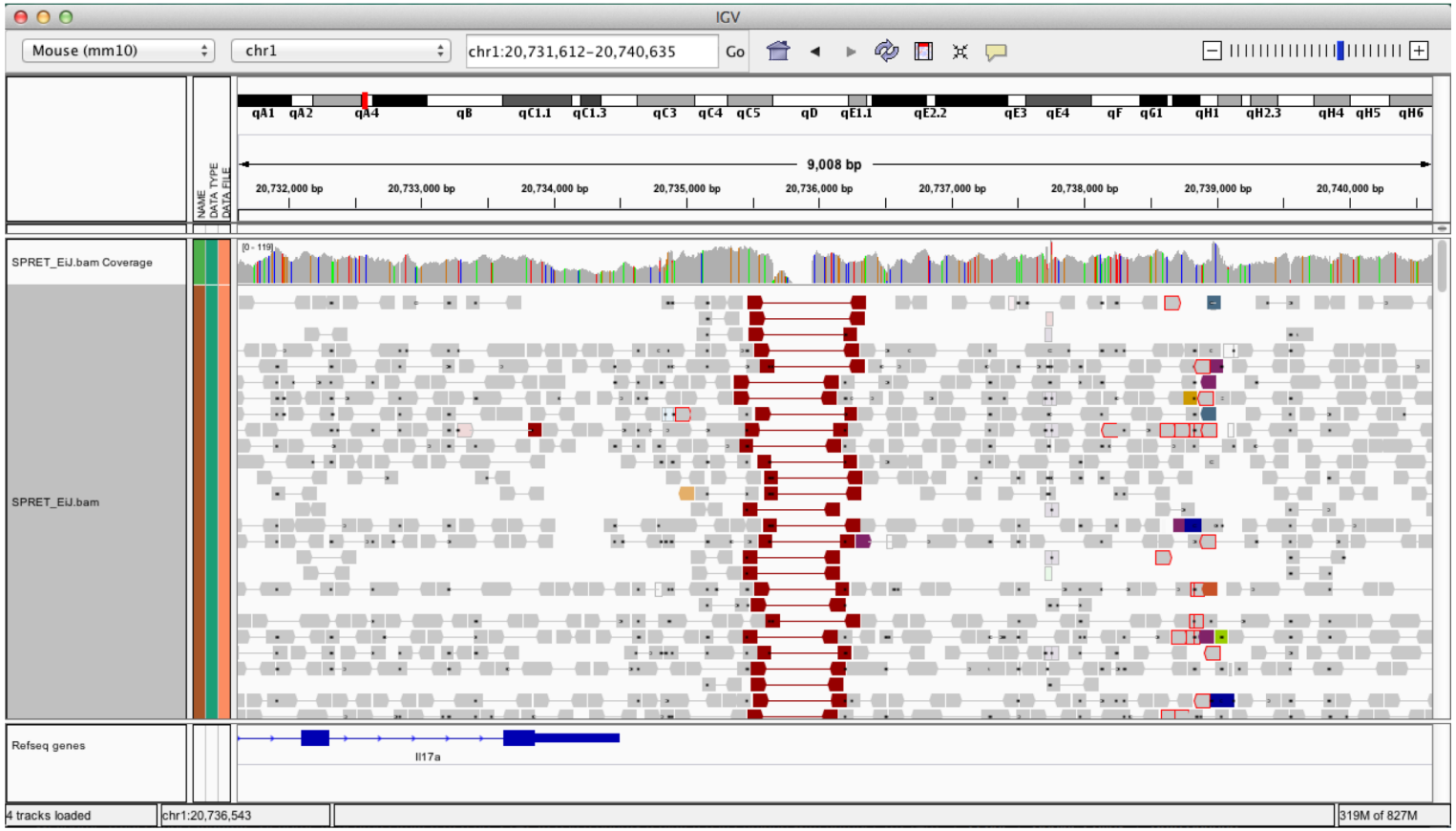
- The left read has 78M and then is softclipped.
- The right read aligns perfectly.
- The middle read alignment is the ‘right half’ of the left read. It is hard-clipped and marked as ‘secondary’ (see flags).


this read has three alignments

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

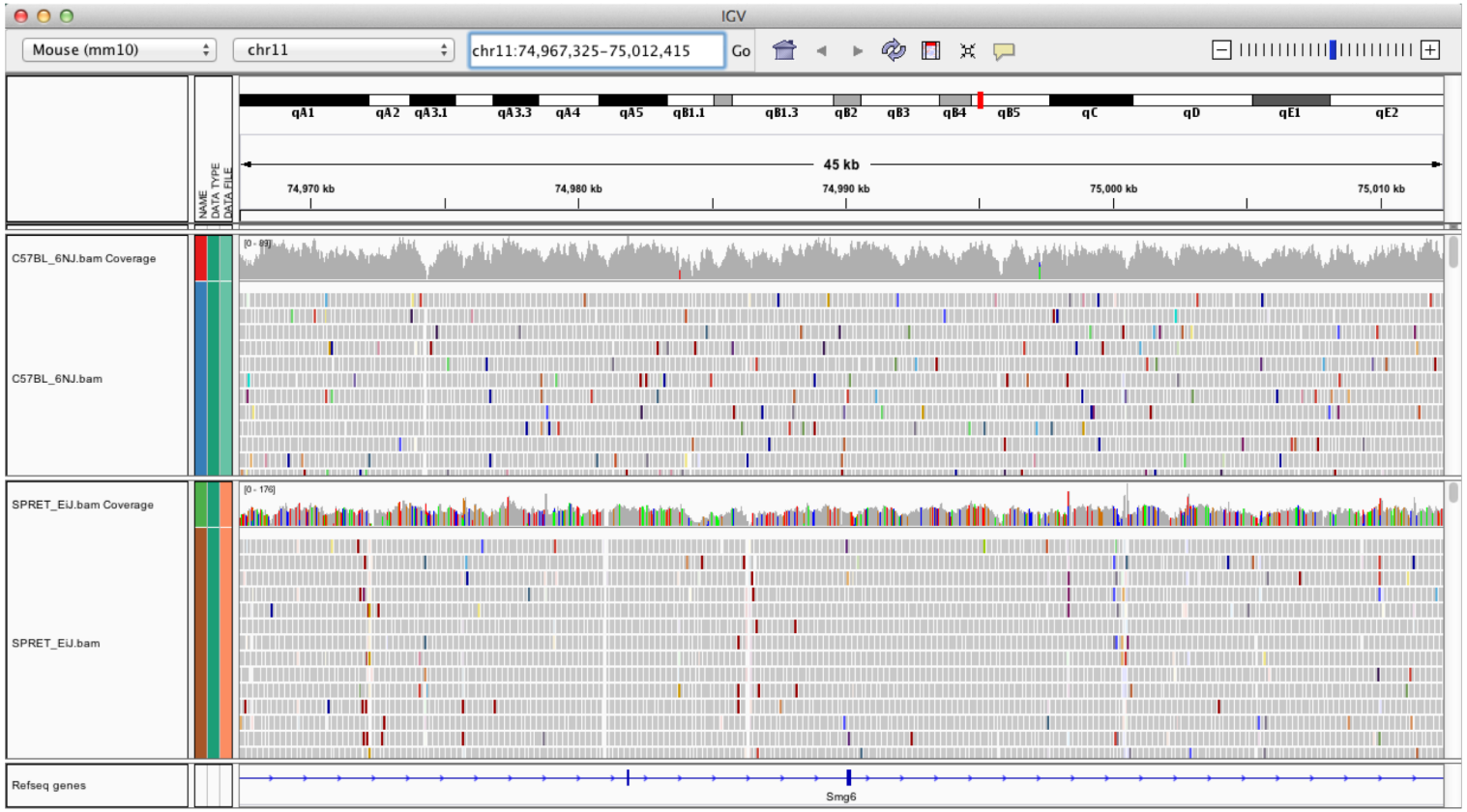
http://software.broadinstitute.org/software/igv/

# Viewing reads as pairs




http://software.broadinstitute.org/software/igv/

# Integrative Genomics Viewer (IGV)




You can view multiple samples at the same time.

http://software.broadinstitute.org/software/igv/

# NGS read alignment

- Introduction to read alignment.
- Main methods and aligners.
- Visualisation of aligned reads.
- NGS workflows, QC and BAM improvement.

# A typical NGS workflow

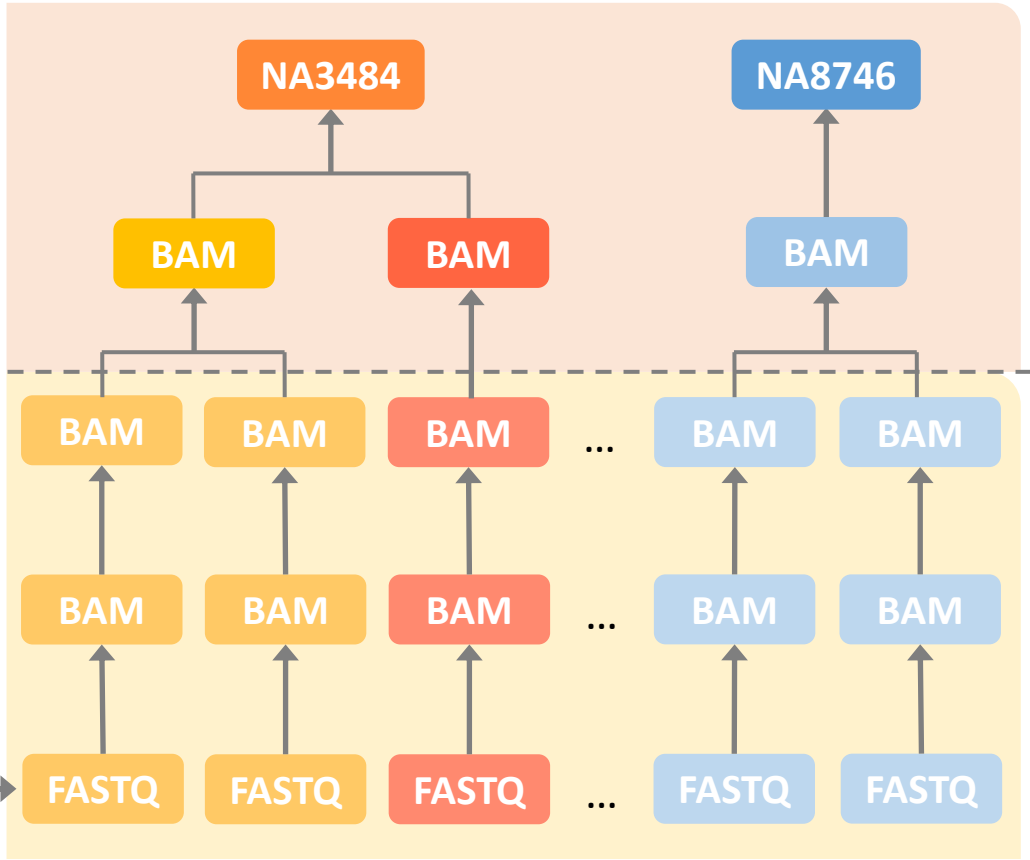
NGS experiments often have tens to hundreds/thousands of samples.

**to increase depth**

Sometimes a sample is split across several libraries.

Often each library is split across multiple lanes.

**to mitigate batch effects; protect against run failures**



# Post-alignment QC

- Several useful metrics to assess the quality of the data and alignments produced:
  - Mapping rate (should be > 90%).
  - Duplication rate.
  - GC content vs read depth.
  - Fragment size distribution.
  - Indels by cycle.

- Samtools produces diagnostic plots.

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

http://www.htslib.org/doc/samtools-1.6.html

# A typical NGS workflow

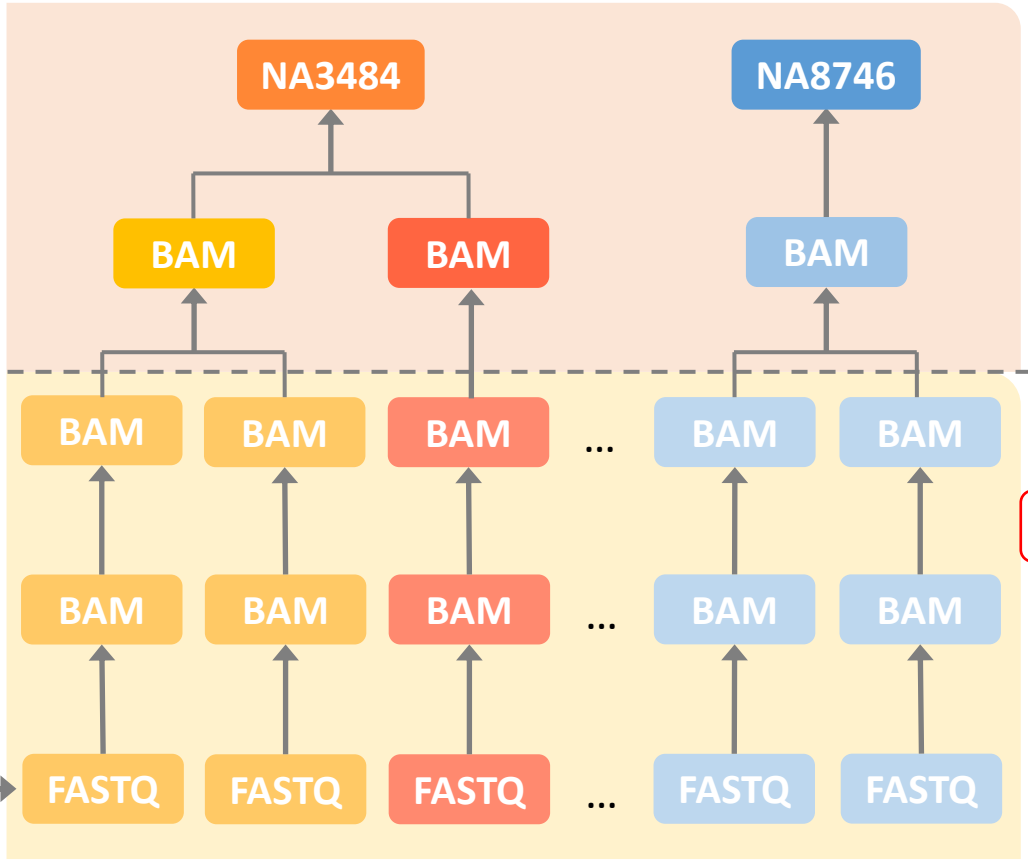
NGS experiments often have tens to hundreds/thousands of samples.

**to increase depth**

Sometimes a sample is split across several libraries.

Often each library is split across multiple lanes.

**to mitigate batch effects; protect against run failures**




SAMPLE MERGE

LIBRARY MERGE

FREEZE

for variant calling

BAM IMPROVEMENT

post-alignment QC

ALIGNMENT

# BAM improvement

- Variant calling relies on the quality score of each base to distinguish sequencing errors from true polymorphisms.
  - Base quality = chance that the the base call is incorrect.

    Phred values (log scale).

    Q10 = 1 in 10 chance the base call is incorrect.

    Q20 = 1 in 100 chance the base call is incorrect.
- 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, quality scores need to be recalibrated.

  BQSR = base quality score recalibration.

# Base recalibration

- **Remove** all known variants (dbSNP, 1000G, etc.)
- **Assume** all other mismatches are sequencing errors.
- Infer the mean observed (empirical) error rate for bins:
  - Position of base in read (from 1 to read length).
  - Dinucleotide sequence context (AA, AT, …, CA, CT).
  - **Empirical rate** = number of mismatches / total number in bin.
- Compare empirical rate to reported sequencer base quality in each of these categories and derive the adjustment for each.

https://gatkforums.broadinstitute.org/gatk/discussion/44/base-quality-score-recalibration-bqsr

# Base recalibration

- **Example:**
  - The sequencer reports Q25 base calls for a “T” in context “AT”.

    After alignment, we can see that a “T” in context “AT” actually mismatches the reference 1 in 100 times.

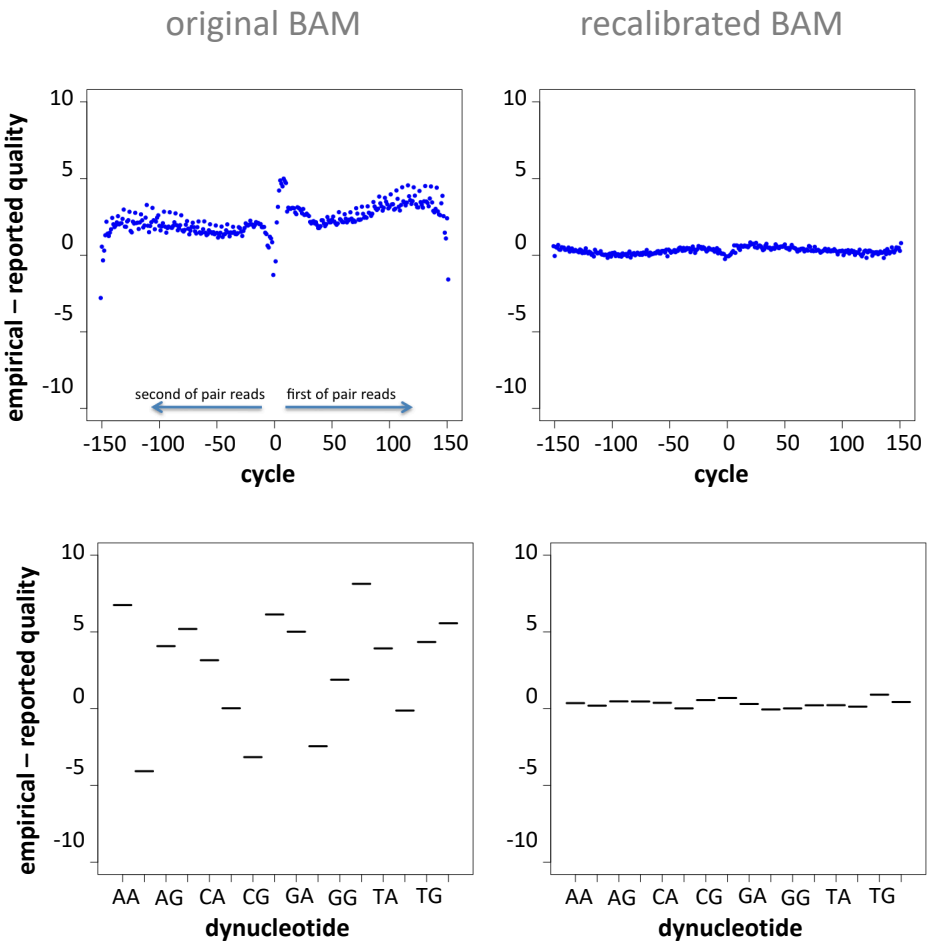
    The base quality should be Q20.

    Adjustment => for every “T” in context “AT” subtract 5 from their base quality.

**NOTE**: requires a reference genome and a catalogue of variable sites.

https://gatkforums.broadinstitute.org/gatk/discussion/44/base-quality-score-recalibration-bqsr

# Base recalibration effects




https://gatkforums.broadinstitute.org/gatk/discussion/44/base-quality-score-recalibration-bqsr

# A typical NGS workflow

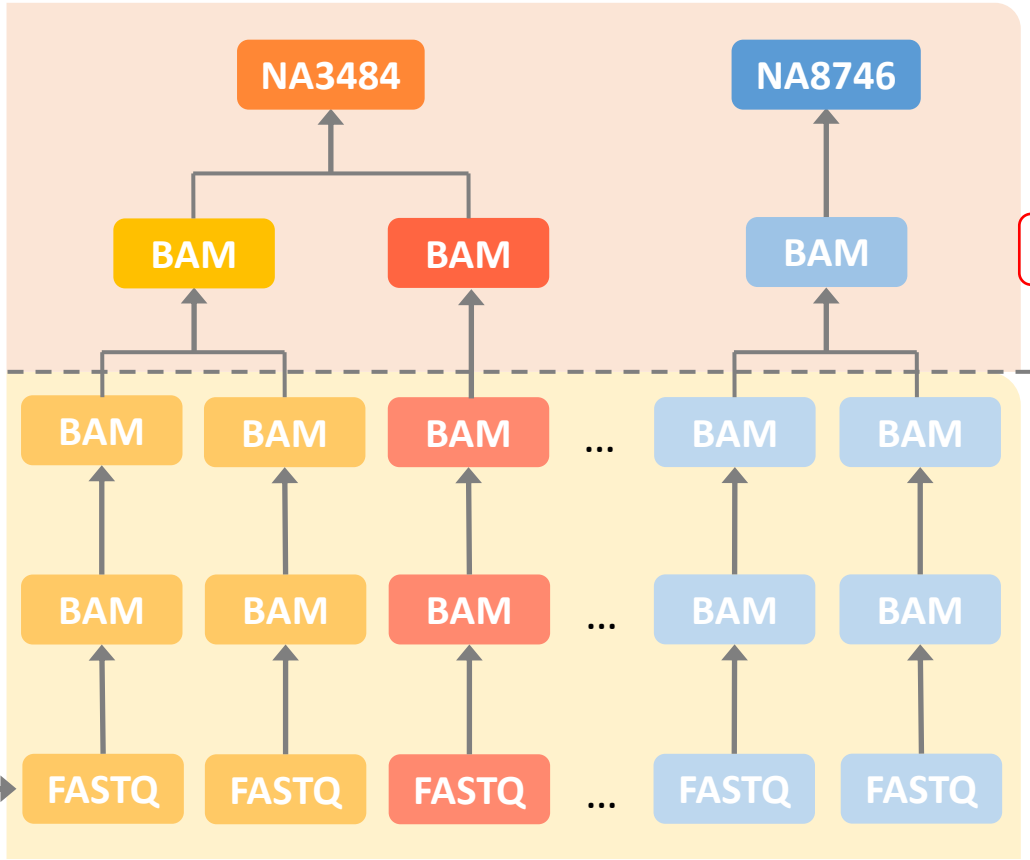
NGS experiments often have tens to hundreds/thousands of samples.

**to increase depth**

Sometimes a sample is split across several libraries.

Often each library is split across multiple lanes.

**to mitigate batch effects; protect against run failures**




**SAMPLE MERGE**

library deduplication

**LIBRARY MERGE**

FREEZE

**BAM IMPROVEMENT**

post-alignment QC

**ALIGNMENT**

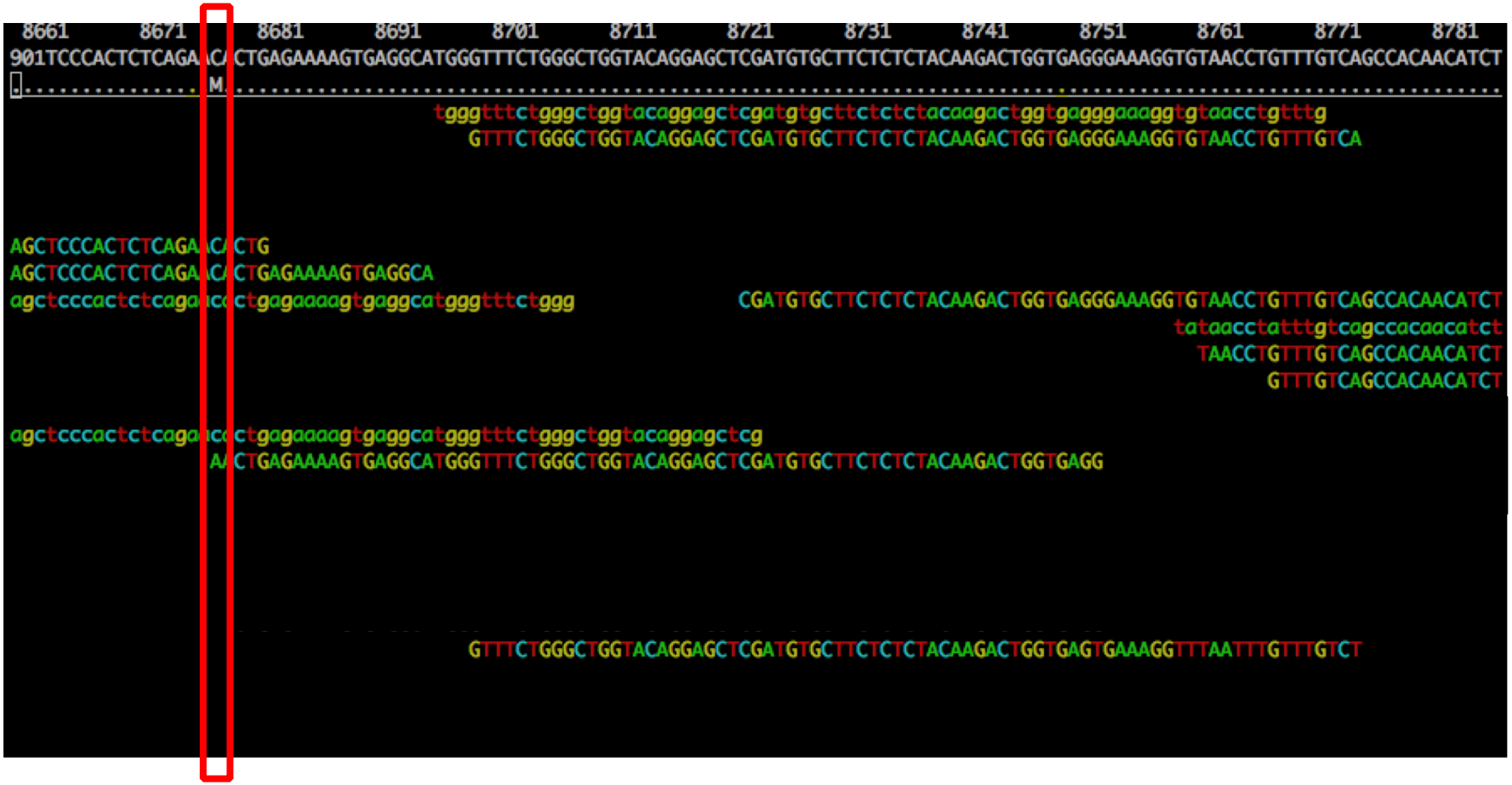
# Read duplicates

- During library prep there is often a **PCR** amplification step.
- This results in duplicate DNA fragments in the final library.
  - If PCR errors are introduced, these can result in false SNP calls.
- **Mark duplicates:**
  - Identify read-pairs where the outer ends have the exact same coordinates.
  - Keep one copy and mark all others as duplicates.

```
samtools rmdup
Picard/GATK MarkDuplicates
```

# Read duplicates

# Read duplicates



# The exercise

- Align NGS data with `bwa`.
- Mark PCR duplicates with `Picard`.
- Generate and analyse post-alignment QC stats.
- Visualise the alignments in IGV.

- If you have time
  - Process data from a library with several lanes of sequencing data.
  - Merge the results.