Sequencing read alignment and data processing

Thomas Keane,

Head of European Genome-phenome Archive and European Variation Archive

EMBL-EBI

@drtkeane

tk2@ebi.ac.uk

Read alignment

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 genome

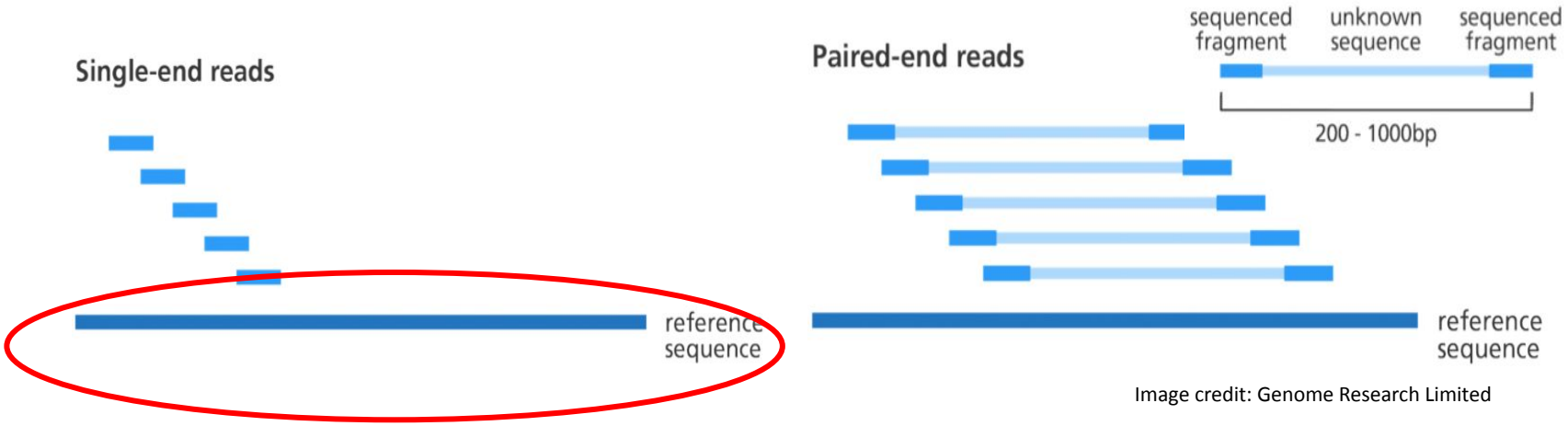


Image credit: Genome Research Limited

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?

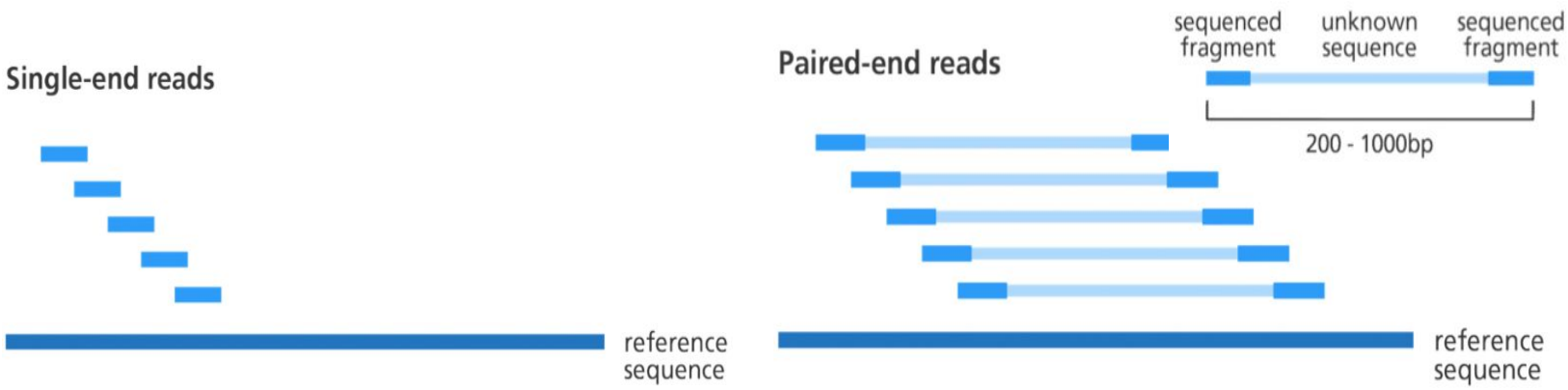
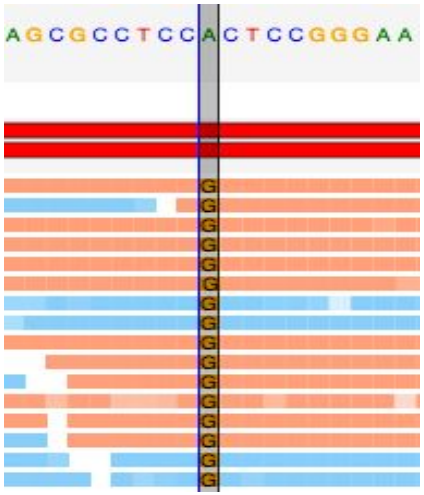
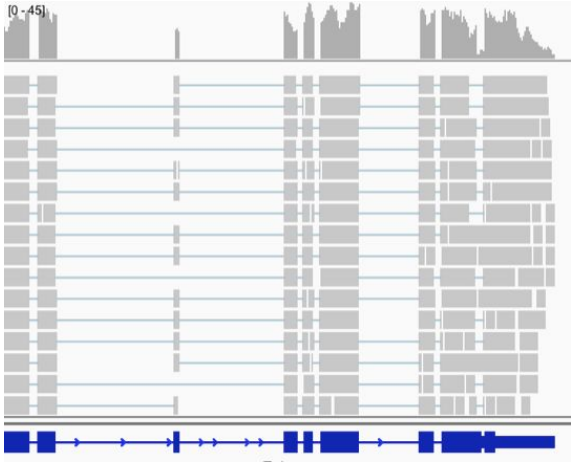


Image credit: Genome Research Limited

Align DNA: Identify variation



ALIGN RNA: Transcript abundance

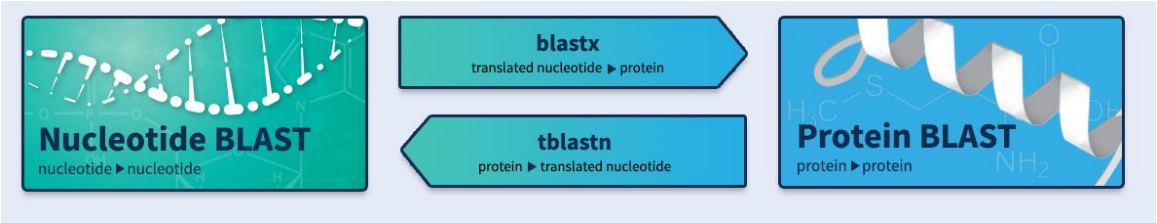


Sequence Alignment

Principles and approaches have not changed much since 80's

Basic Local Alignment Search Tool (BLAST)

- 'Seed and extend' approach
- Query sequences vs. larger database of sequences
- Split query sequences into short sequences (~10bp) and search for locations where these cluster in the larger database of sequences

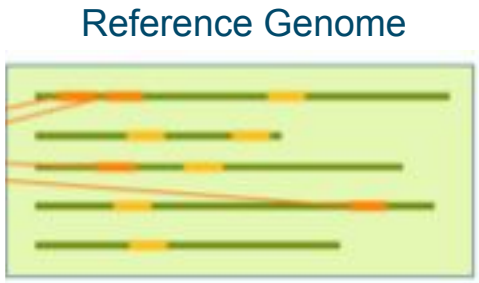


NGS: Nucleotide based alignment

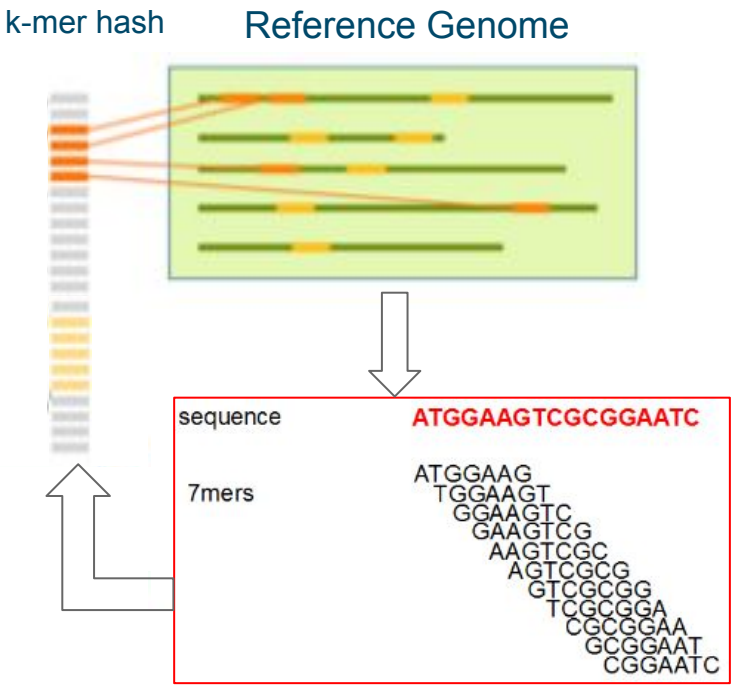
- Very small evolutionary distances (human-human, or related strains of the reference genome)
- Assumptions about the number of expected mismatches to speedup alignment programs
- Gapped vs ungapped alignment
  - Typically want to allow for possibility of indels: gapped alignment

NGS has just massively scaled up the challenge

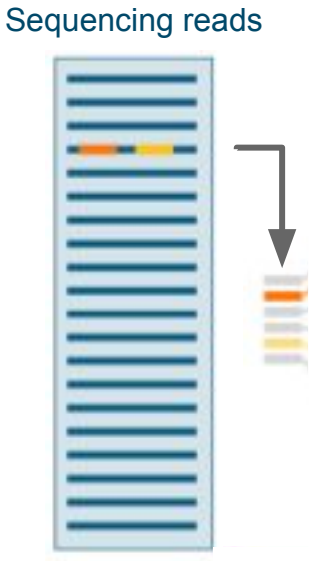
Hash Table Alignment



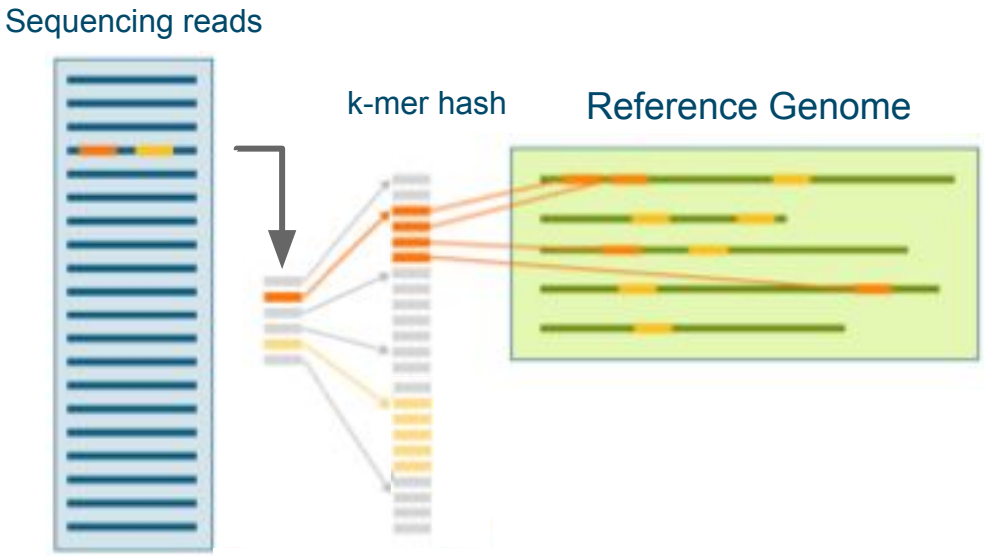
Hash Table Alignment



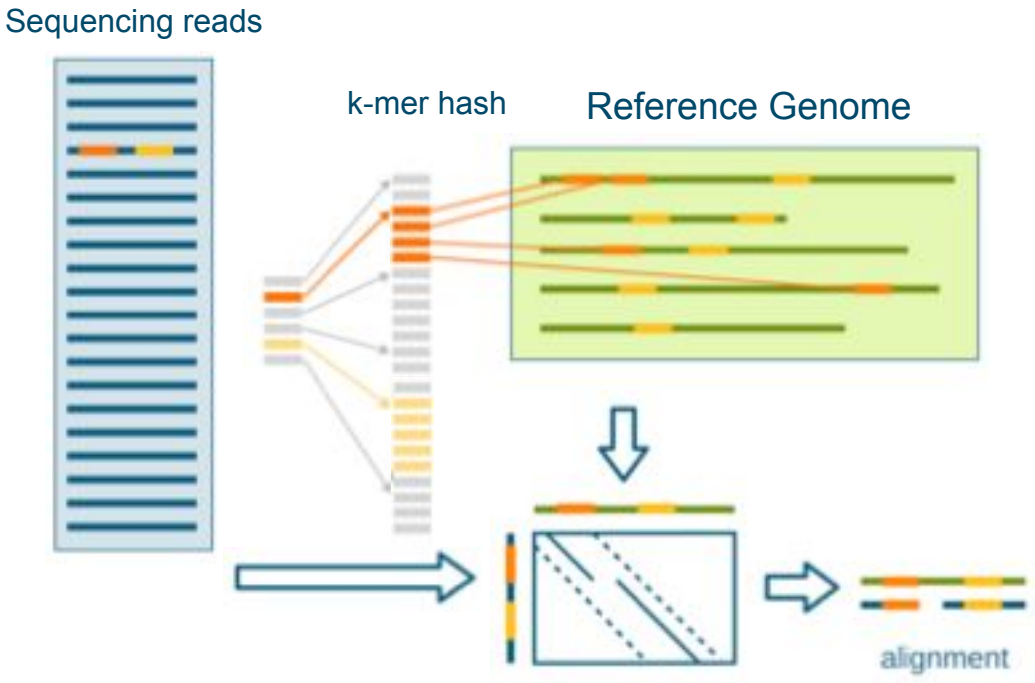
Hash Table Alignment



Hash Table Alignment



Hash Table Alignment



Hash Table Alignment

k-mer is a short fixed sequence of nucleotides

- e.g. a 31-mer is a string of 31 nucleotides

Typical algorithm

- Build a profile (index) of all possible k-mers of length *n* and the locations in the reference genome they occur
  - Several Gbytes in size for human genome
- Foreach sequence read
  - Split into k-mers of length *n*
  - Lookup the locations in the reference via the index (**seed phase**)
  - Pick location 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

Suffix/Prefix Tree Based Aligners

Store all possible suffixes or prefixes to enable 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)

FM-Index based

- Small memory footprint

Examples

- MUMmer, BWA, bowtie

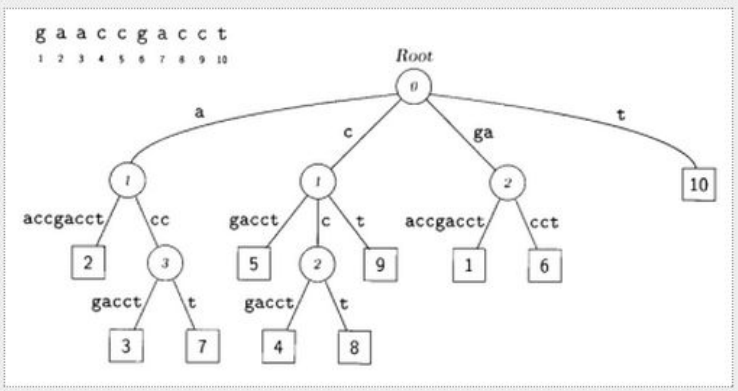


Figure 2
Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. In this example the longest repeated sequence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

Still require a final step to generate local alignment

Delcher et al (1999) *NAR*

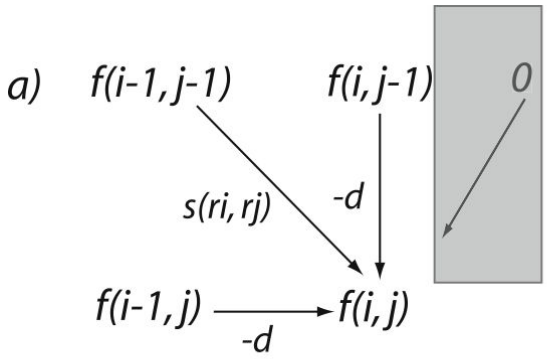
Local Alignment: Smith-Waterman Algorithm (1981)

Take approximate location of where read aligns to and refine the precise alignment to determine the cigar string

Generates the optimal pairwise alignment between two sequences

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



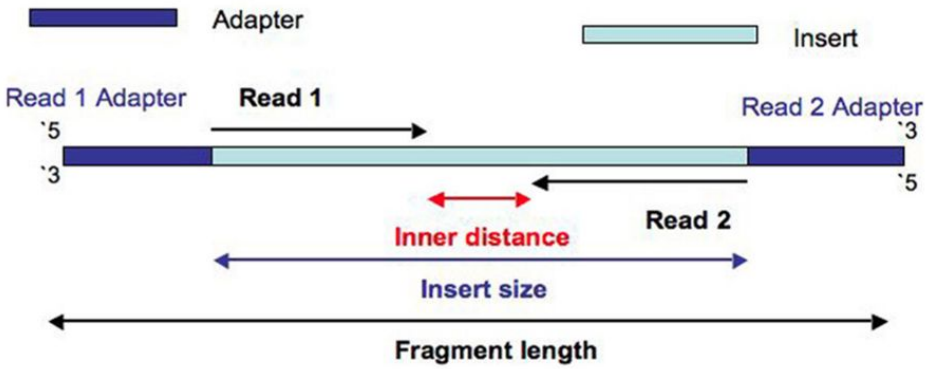
b)

		A	G	G	T	T	G	C
	1	0	-1	-2	-3	-4	-5
A	0	2	1	0	-1	-2	-3
G	-1	1	3	2	1	0	-1
G	-2	0	2	4	3	2	1
T	-3	-1	1	3	4	3	2
C							

Match: +1
Mismatch: 0
Gap open: -1

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

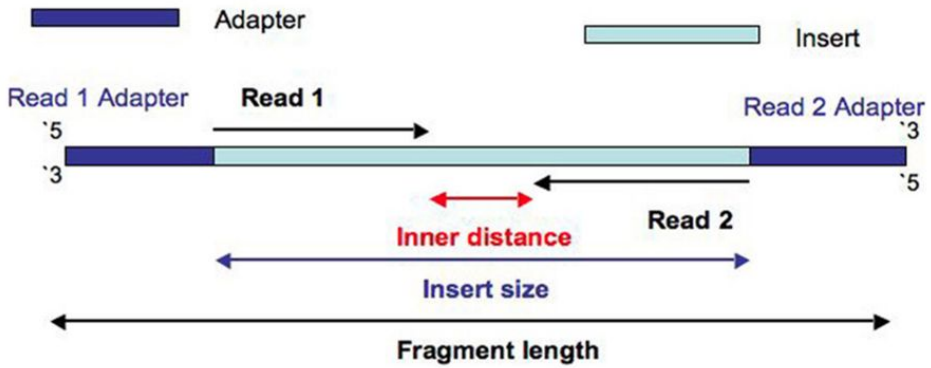
Some Terminology



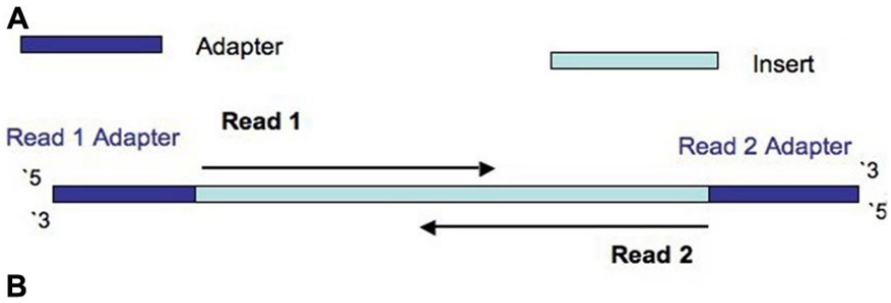
Turner, 2014. PMID:24523726

Some Terminology

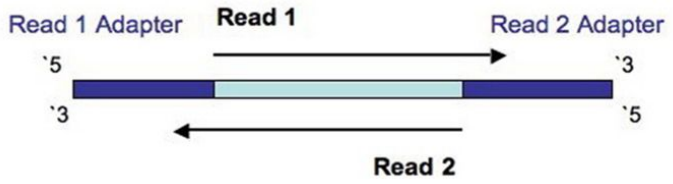
Insert size > length(read 1 + read 2)



Insert size < length(read 1 + read 2)



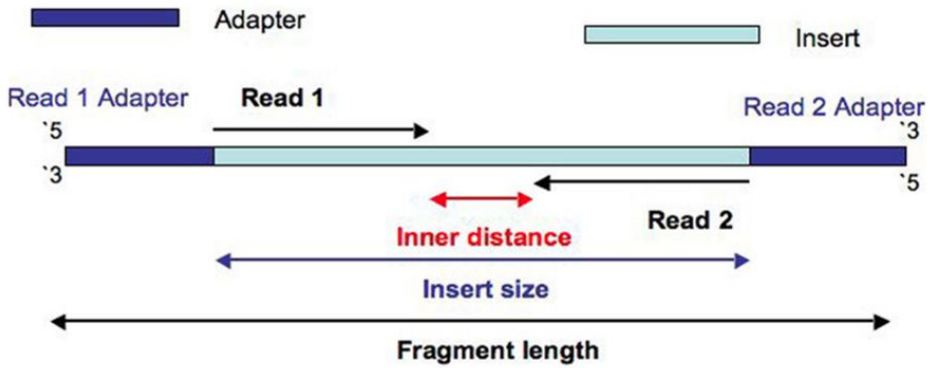
Insert size < length(read 1);
Insert size < length(read 2)



Turner, 2014. PMID:24523726

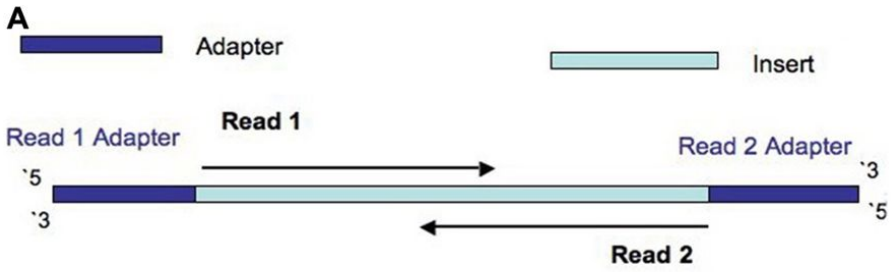
Some Terminology

Insert size > length(read 1 + read 2)

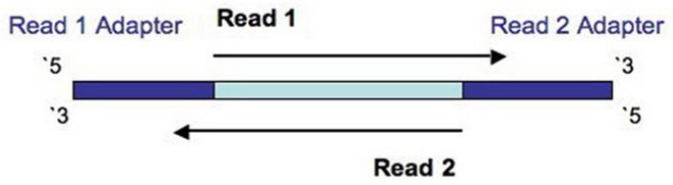


Both reads in a pair get the same "name"

Insert size < length(read 1 + read 2)



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



Turner, 2014. PMID:24523726

Mapping Qualities

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

Genomes contain many different types of repeated sequences

- Transposable elements (40-50% of vertebrate genomes)
- Low complexity sequence
- Reference errors and gaps

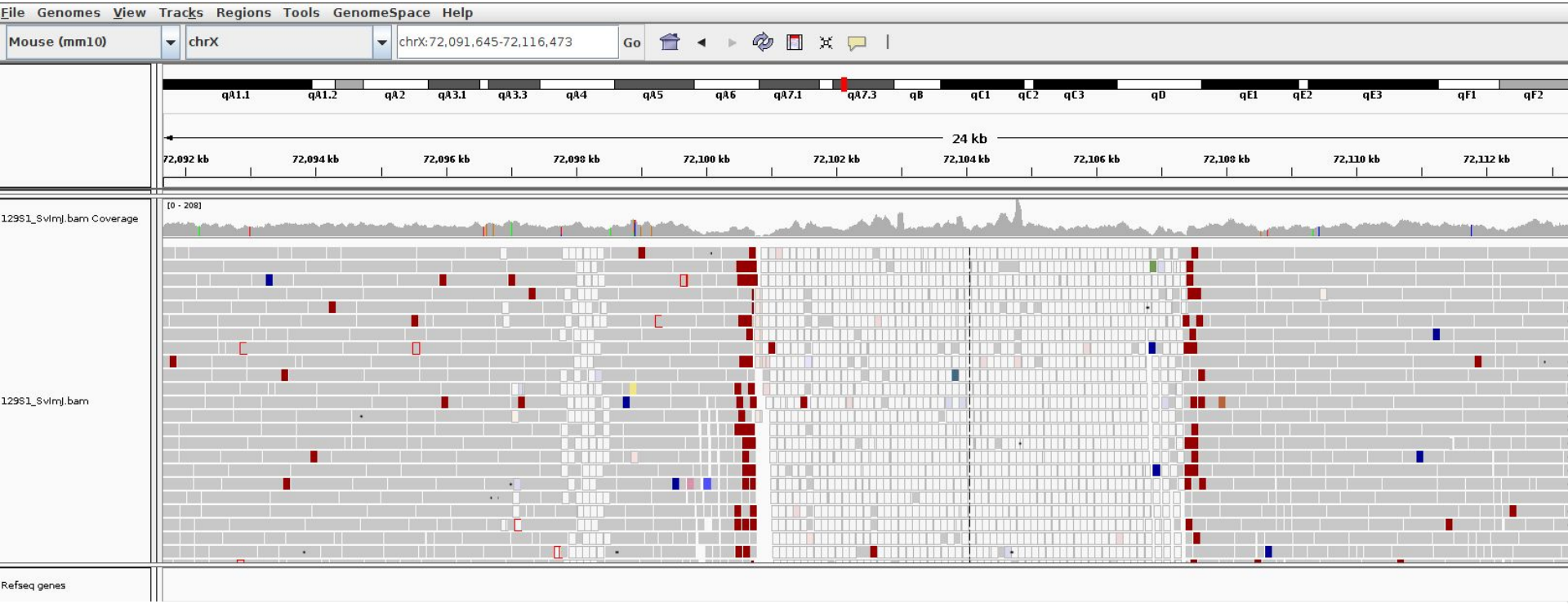
Typically represented as a phred score (log scale)

- Q0 used to indicate no confidence in the alignment of the read to this location
- Q10 = 1 in 10 incorrect (90% accurate)
- Q20 = 1 in 100 incorrect (99% accurate)
- Q30 = 1 in 1000 incorrect (99.9% accurate)

Paired-end sequencing is useful

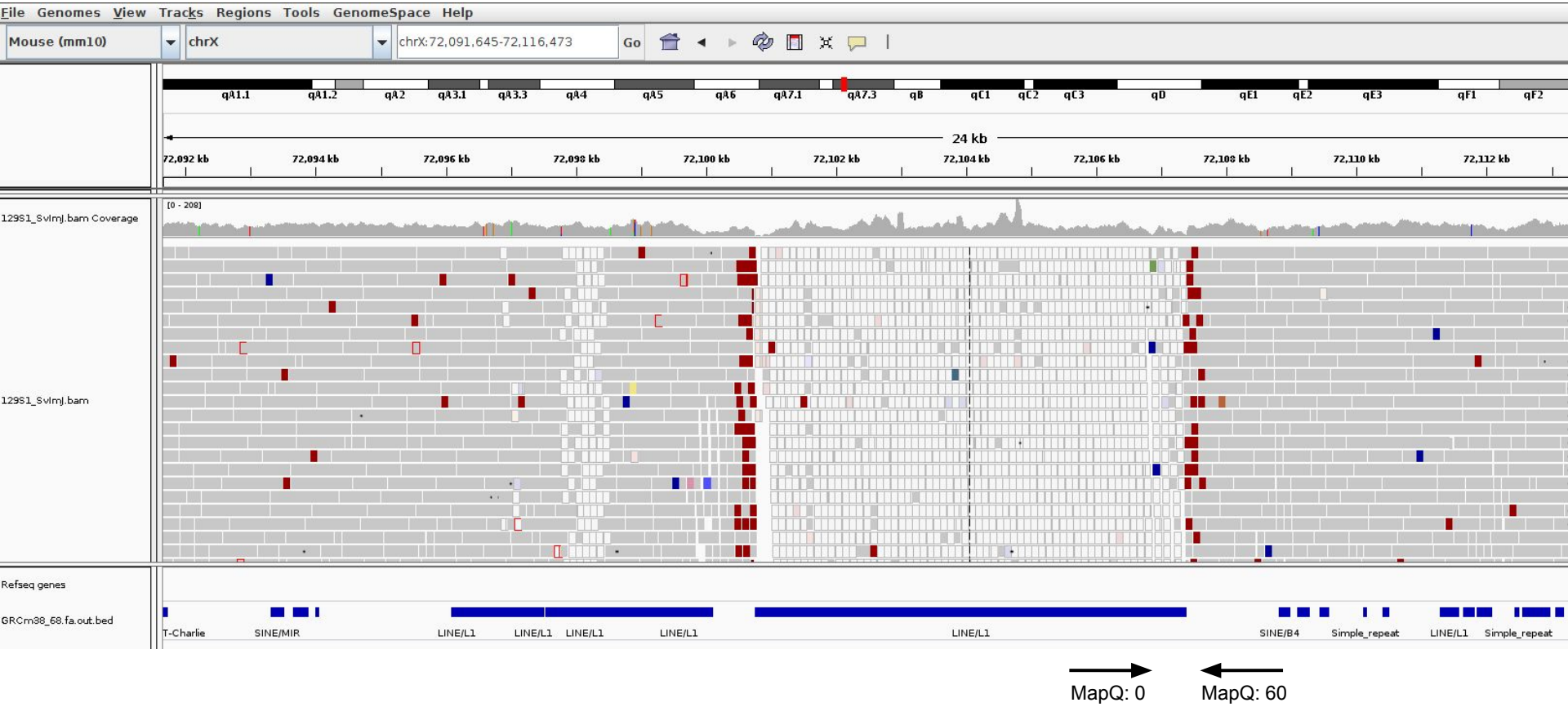
- One end maps inside a repetitive elements and one outside in unique sequence
- Then the combined mapping quality can still be high
- **Hence prefer paired-end sequencing**

Mapping Qualities



Mapping Qualities

Mapping Qualities



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 - many low complexity AT stretches

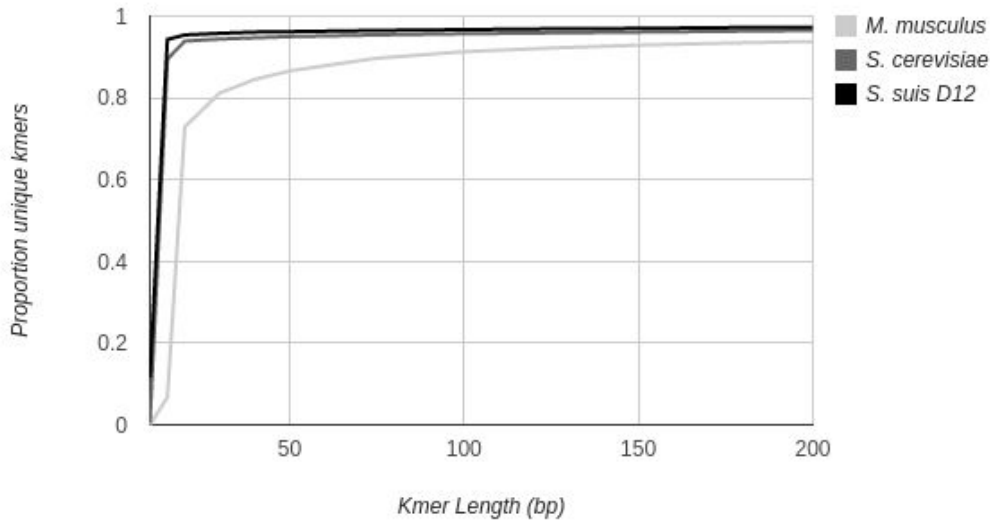
Alignment around indels

- Next-gen read 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
- New 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)
- BWT based aligners work best at low divergence

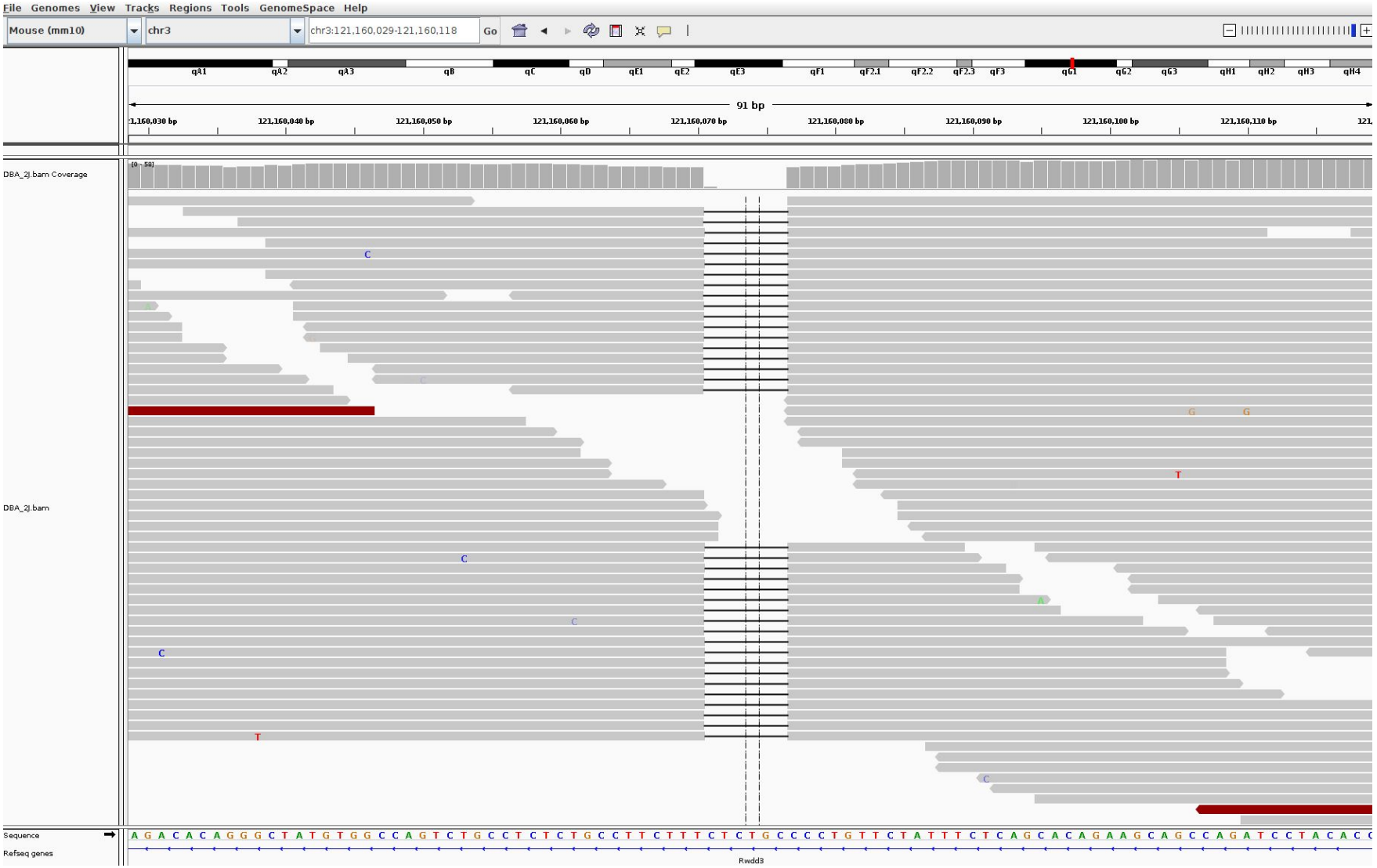
Read Length vs. Uniqueness



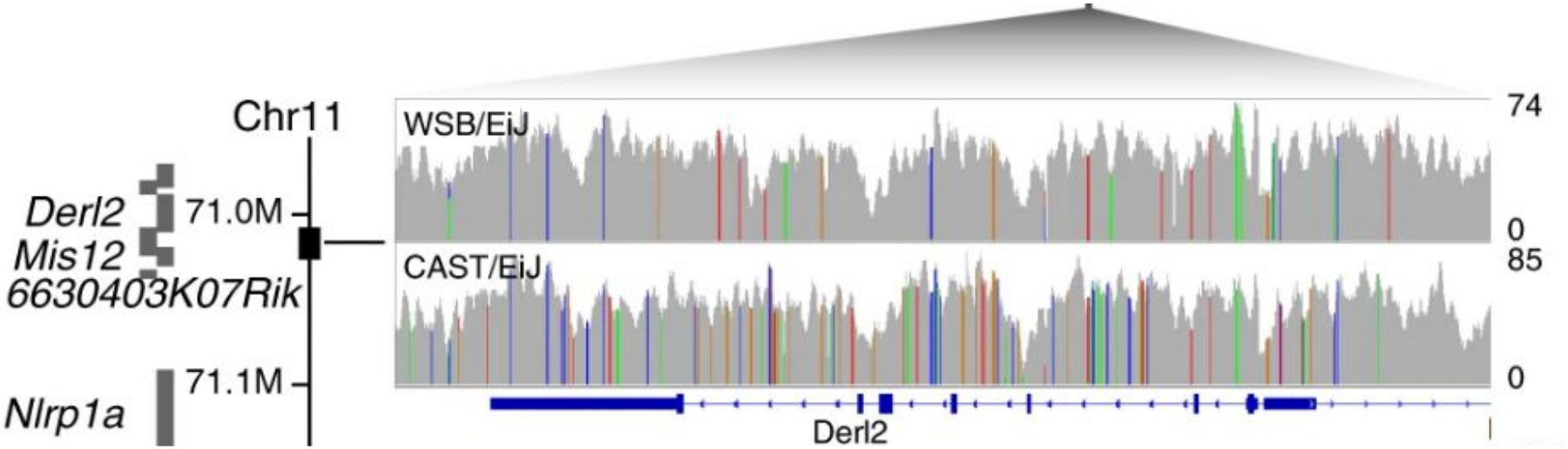
Reference genome mappability: Proportion of reads/kmers unique with error-free reads at various read lengths for *M. musculus*, *S. cerevisiae*, *S. suis* genomes

Example Indel - 2010

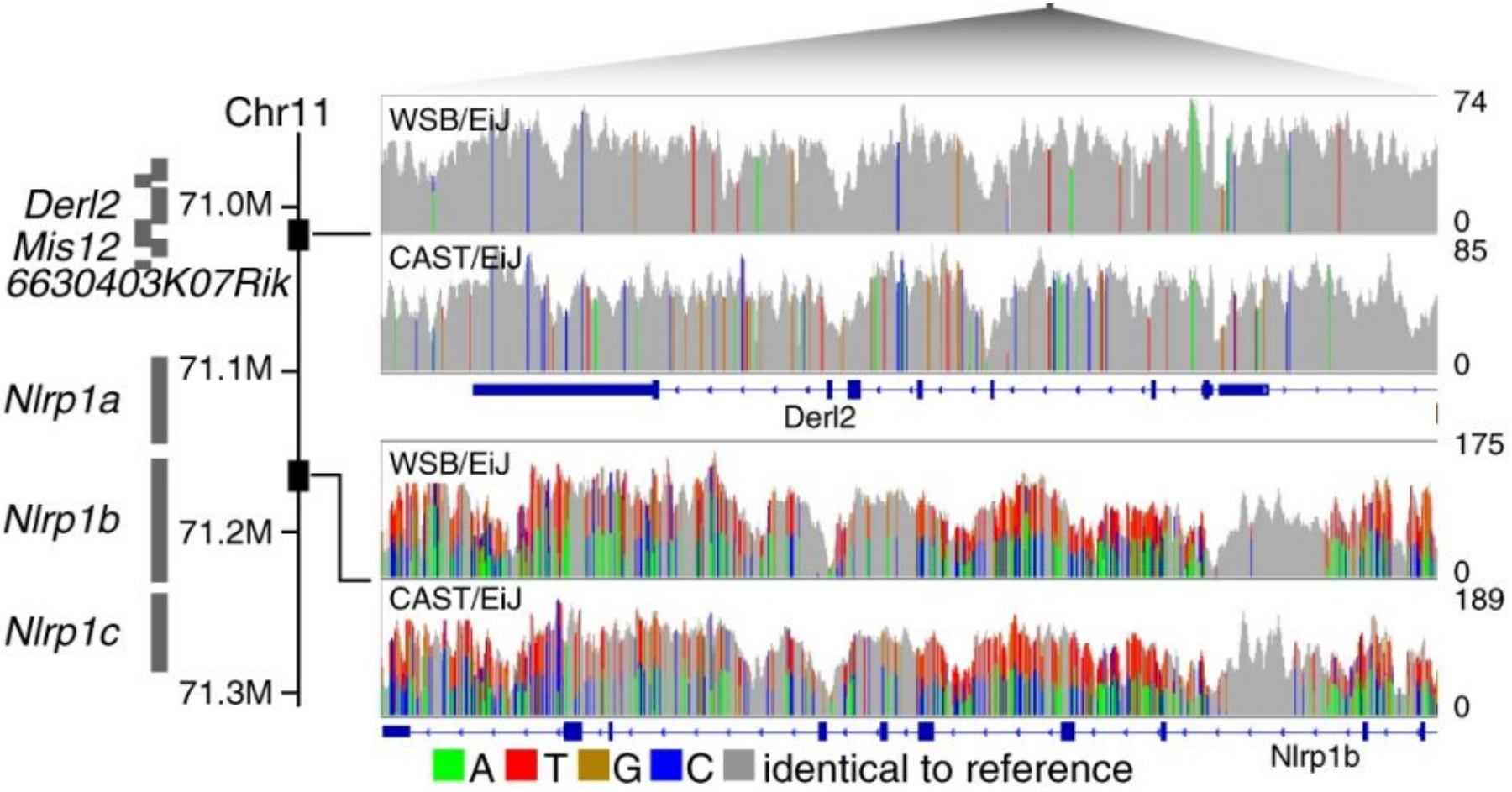
Same Indel - 2014



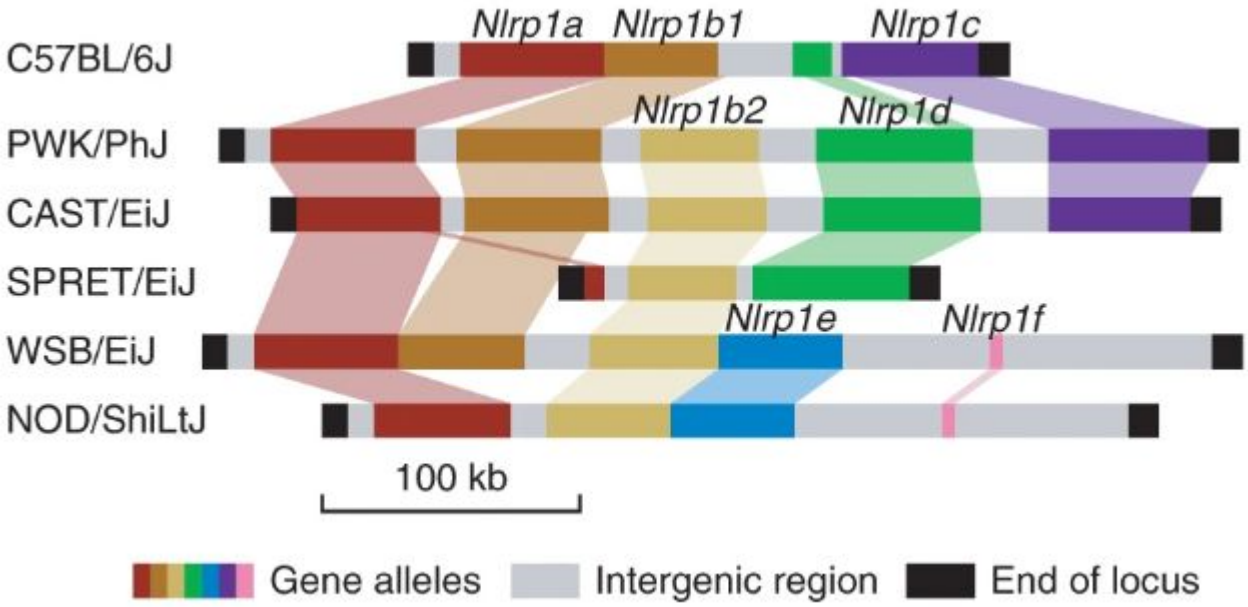
High Divergence



High Divergence



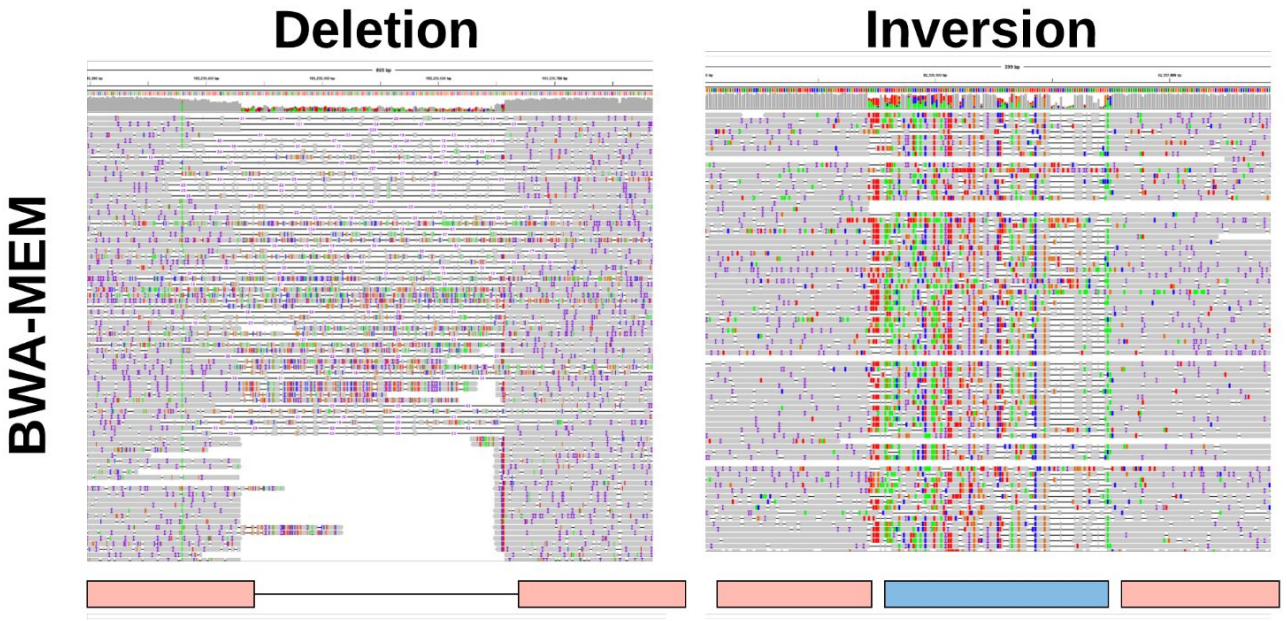
High Divergence



Long read sequencing

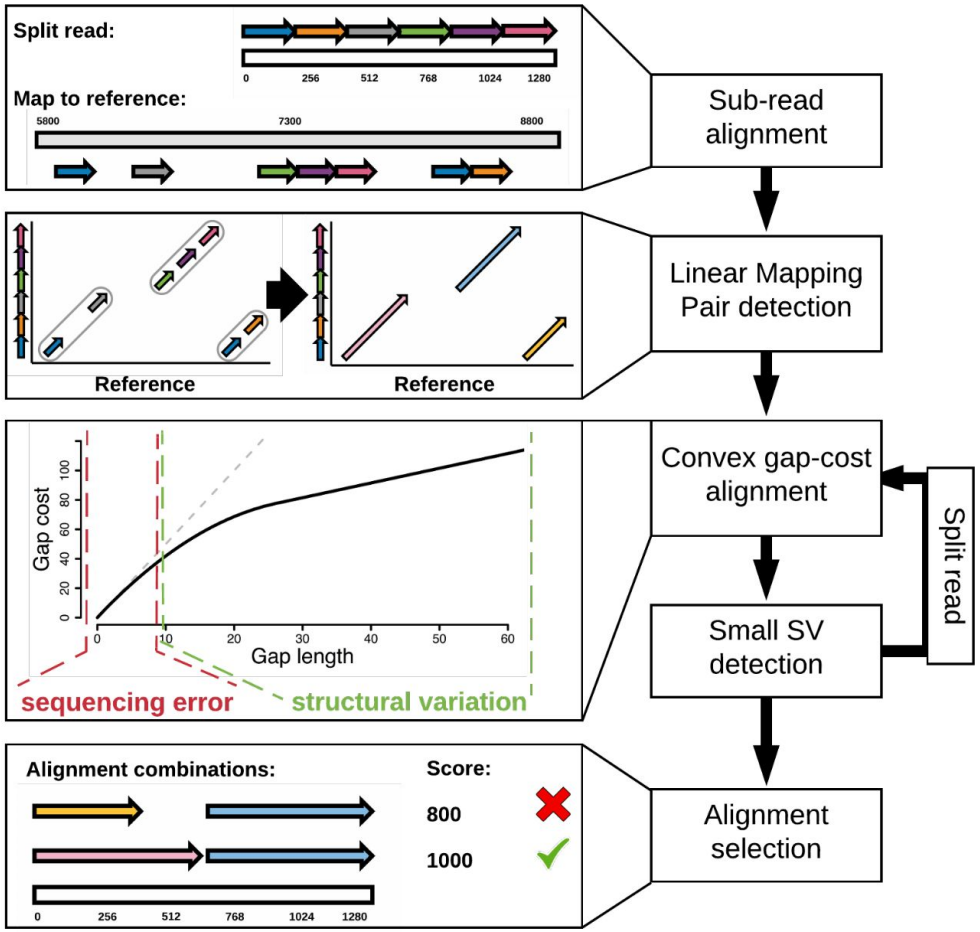
- Single molecule sequencing of large DNA fragments
- Platforms: Oxford nanopore and Pacific Biosciences
- Read lengths 10-20Kbp routinely
- Longer than most common transposable element repeats
- Some new challenges
  - Reads are error prone, 0-10% error
  - Challenging to align the reads correctly

Alignment challenges

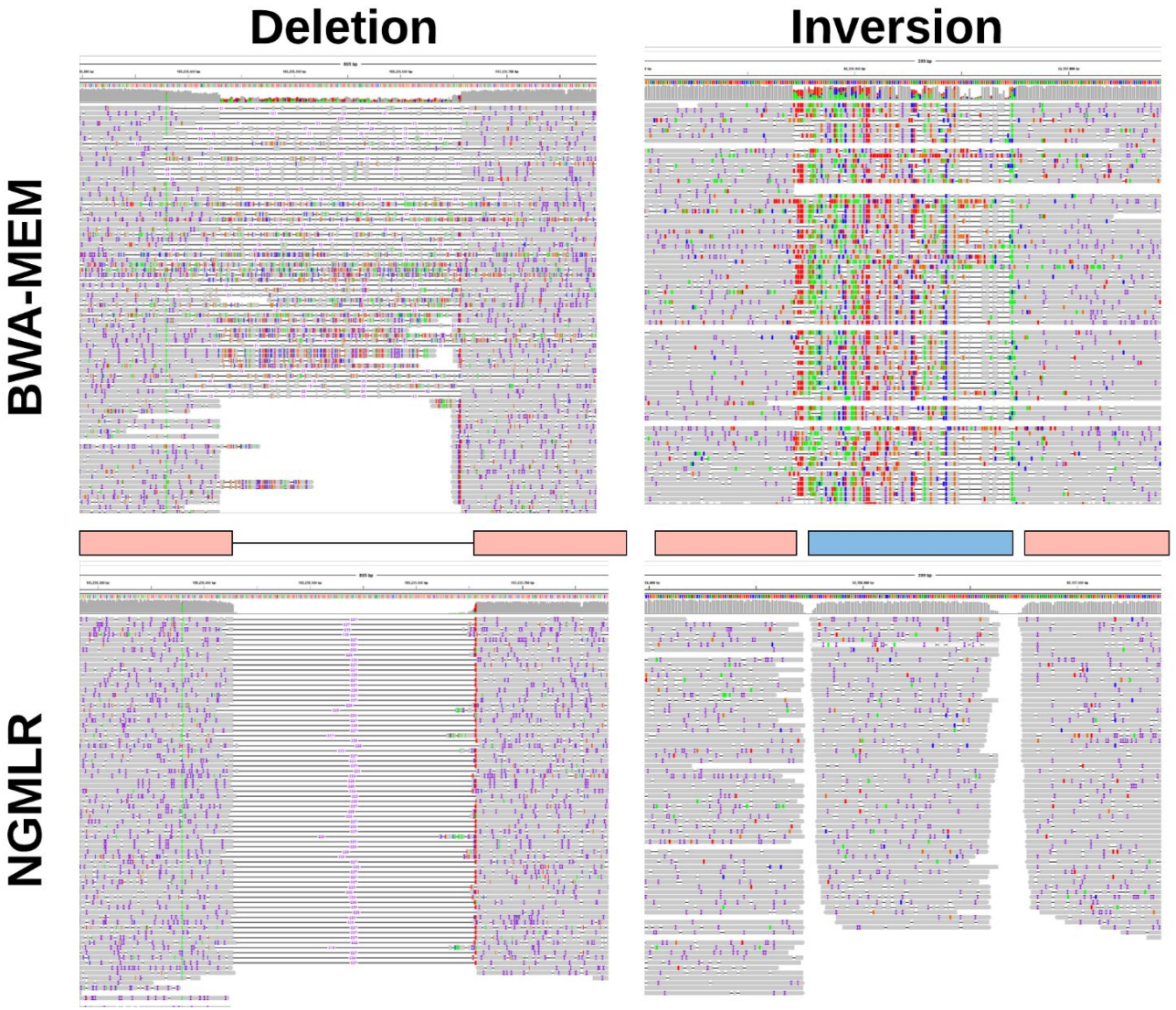


CoNvex Gap-cost alignMents for Long Reads (NGMLR)

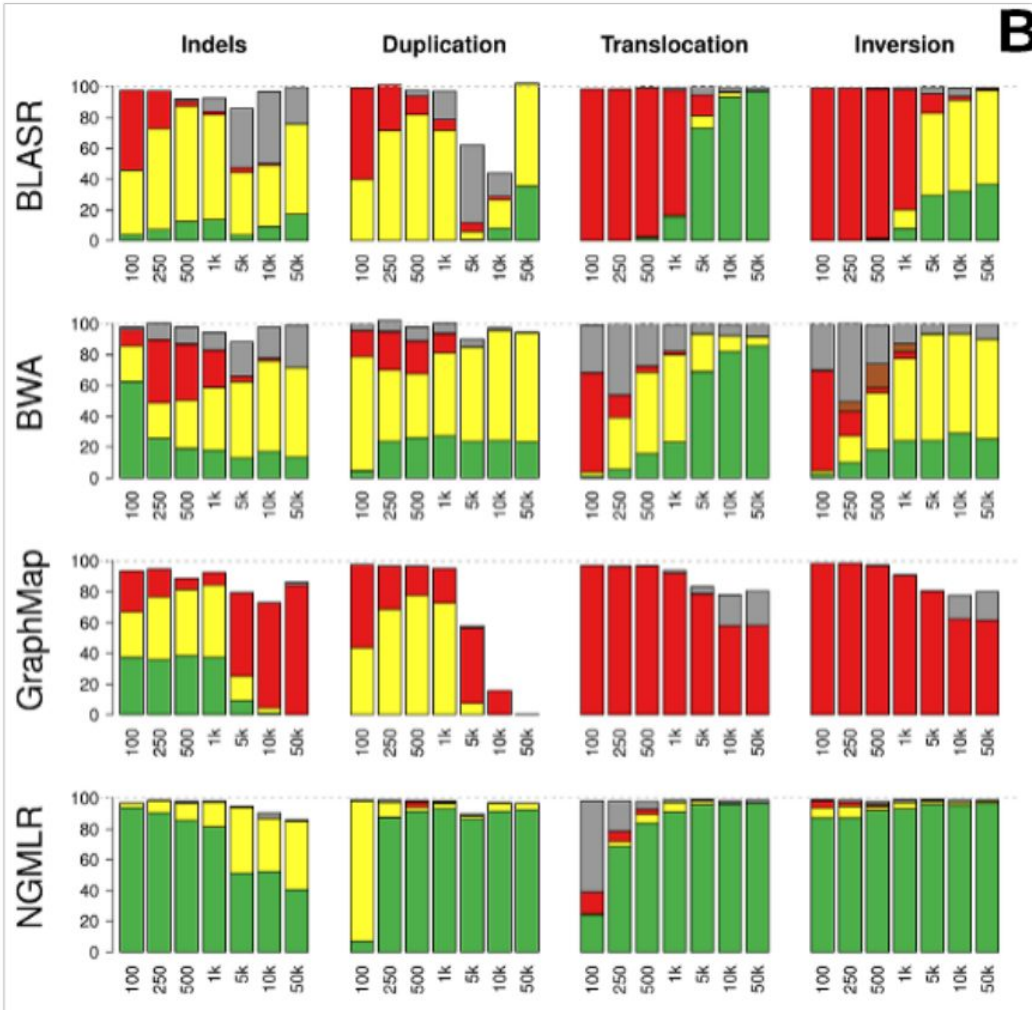
- NGMLR - aligner specifically designed for long reads
- Convex scoring model
  - Extending an indel is penalized proportionally less the longer the indel is



Alignment challenges

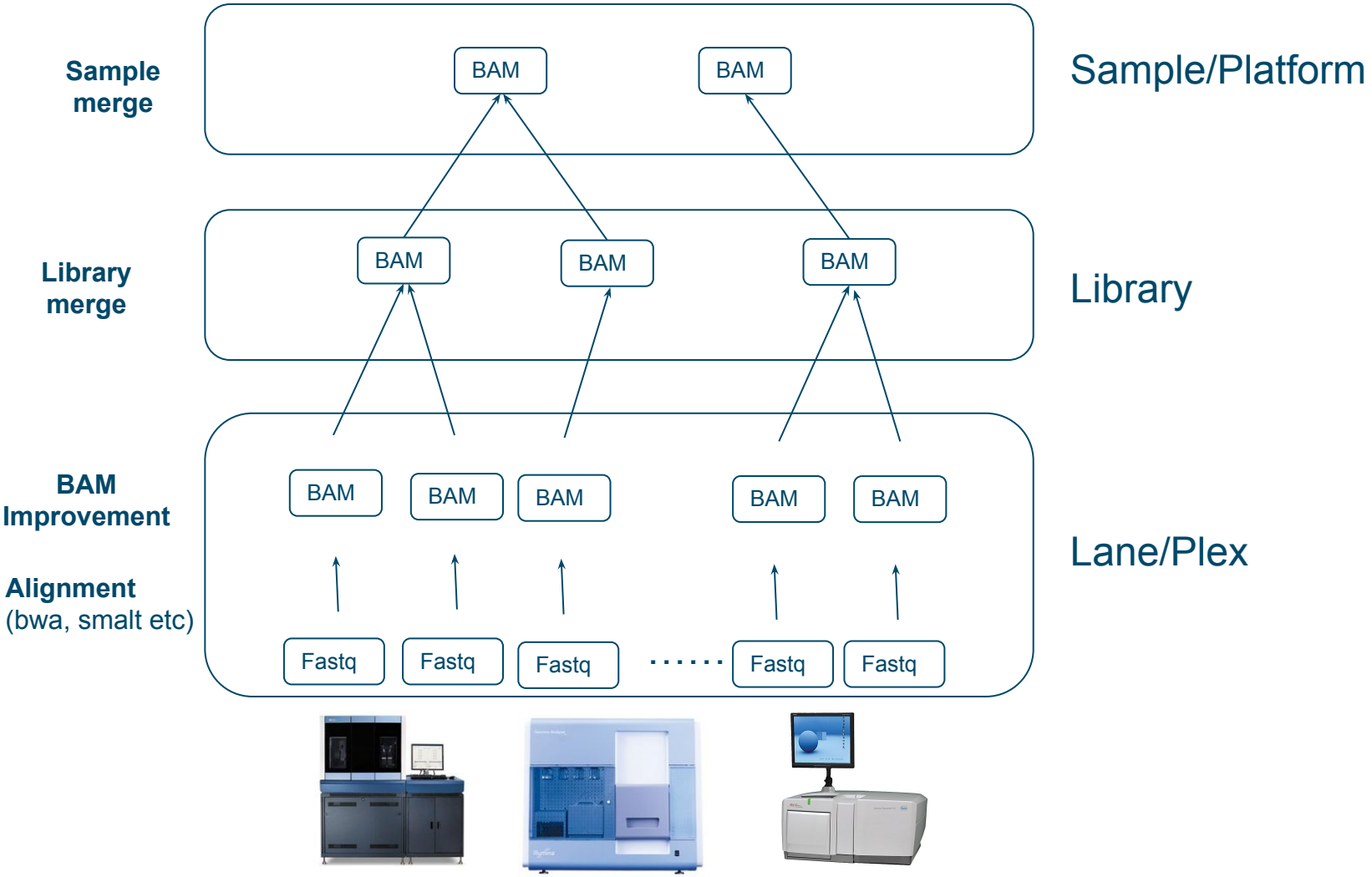


Comparison of aligners (simulated data)



Alignment status: Precise (green), indicated (yellow), forced (red), unaligned reads (white), or trimmed but not aligned through the SV (grey).

Data Production Workflow



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

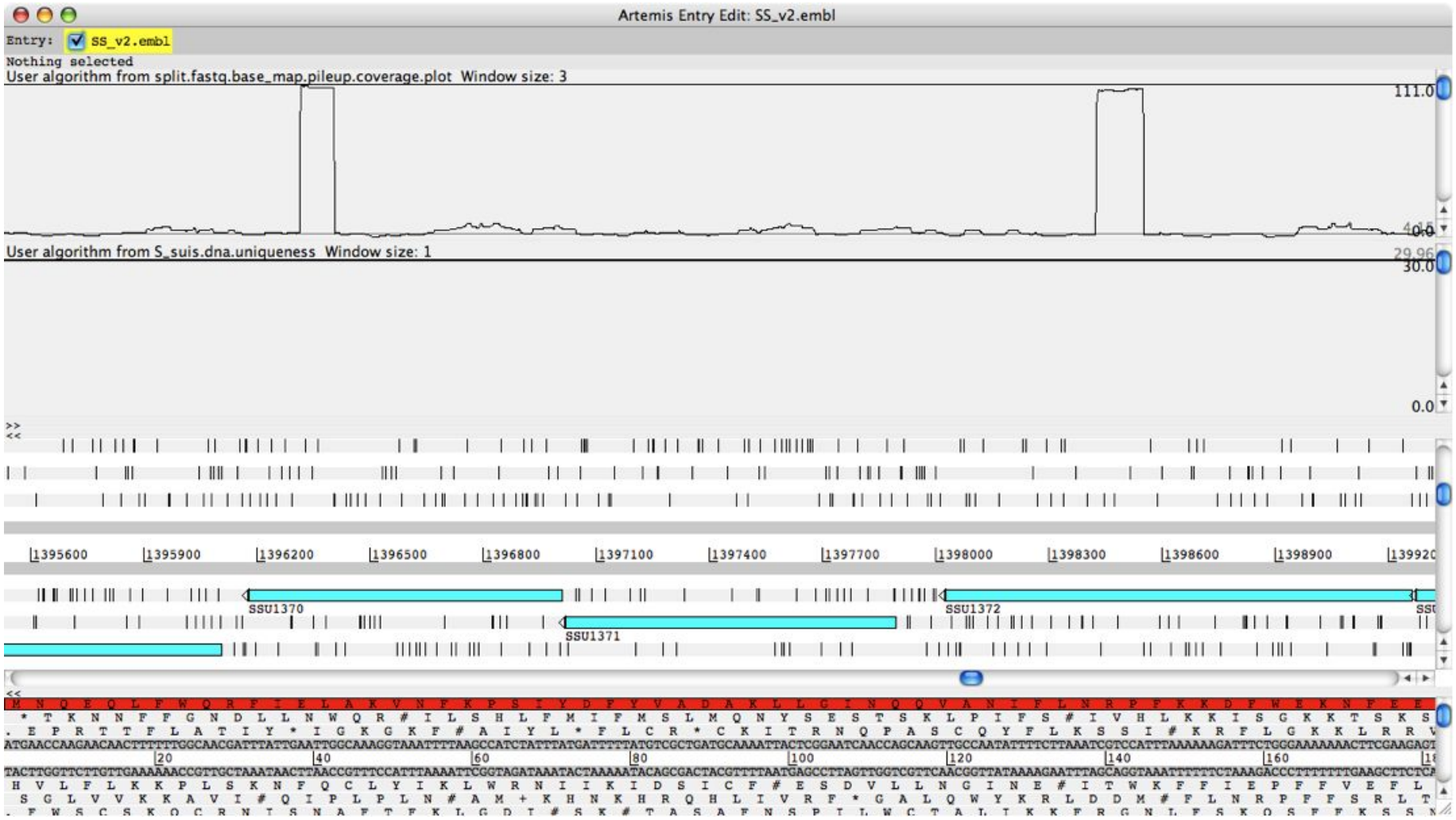
Generally low number of duplicates in good libraries (<5%)

- 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
  - GATK: MarkDuplicates

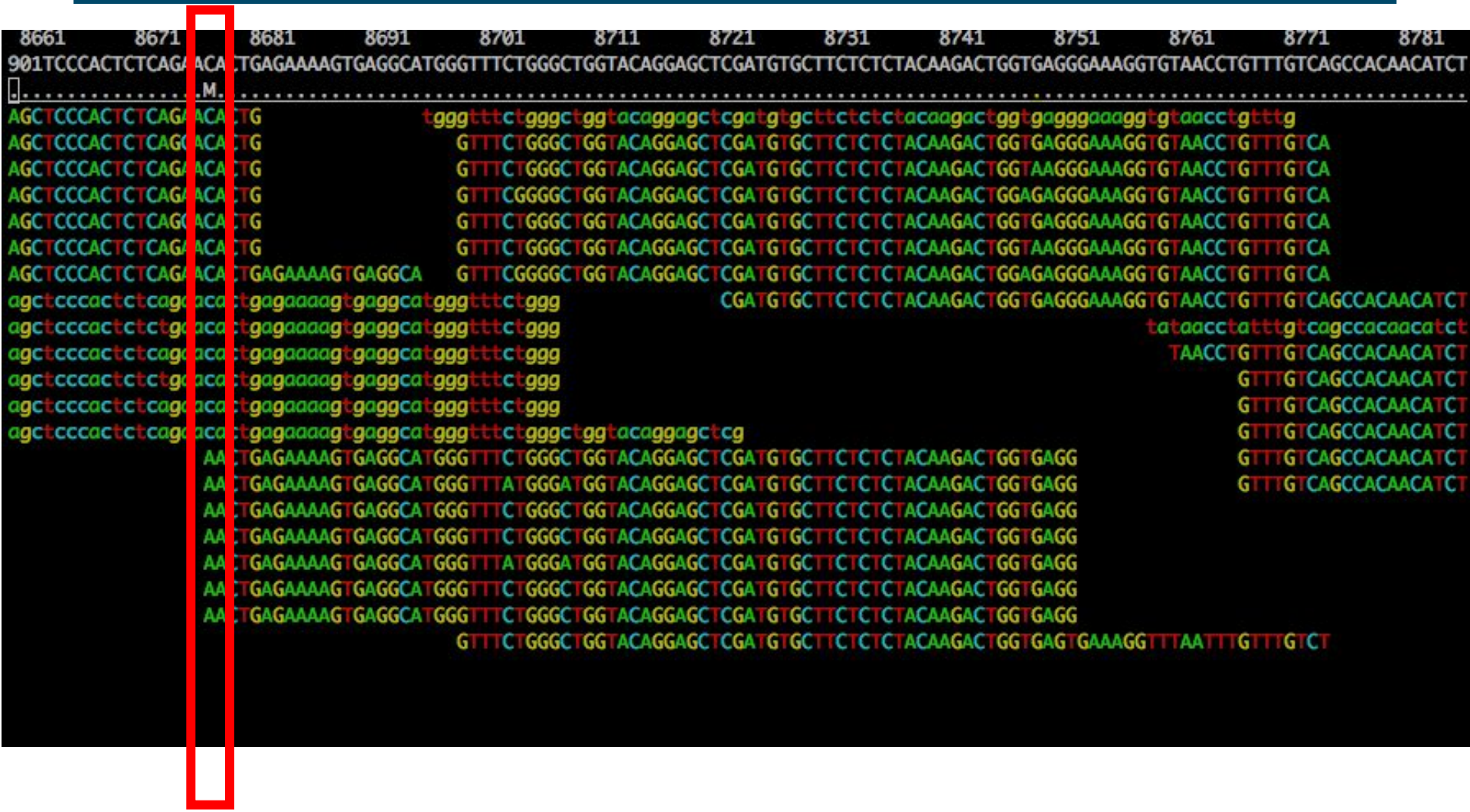
Can result in false SNP calls

- Duplicates manifest themselves as high read depth support

Library Duplicates



Duplicates and False SNPs



Duplicates and False SNPs

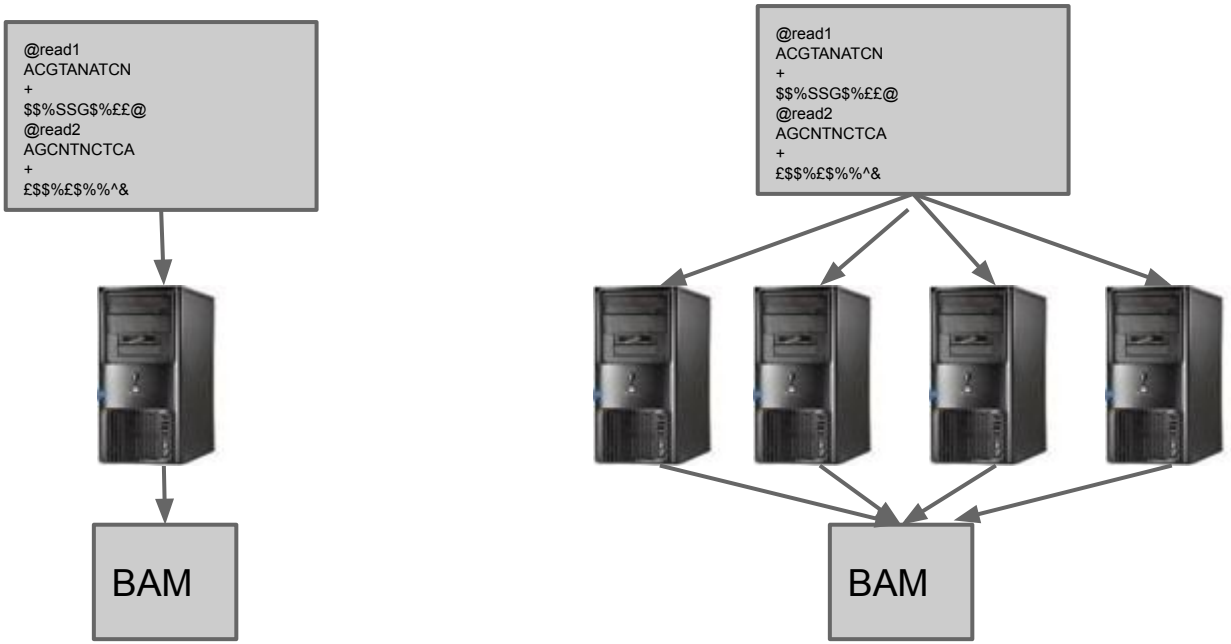
Alignment - Scaling Up

~800M reads/160 Gbp per NextSeq run

- Aligning a single lane of reads can take a long time on a single computer

Parallel computing

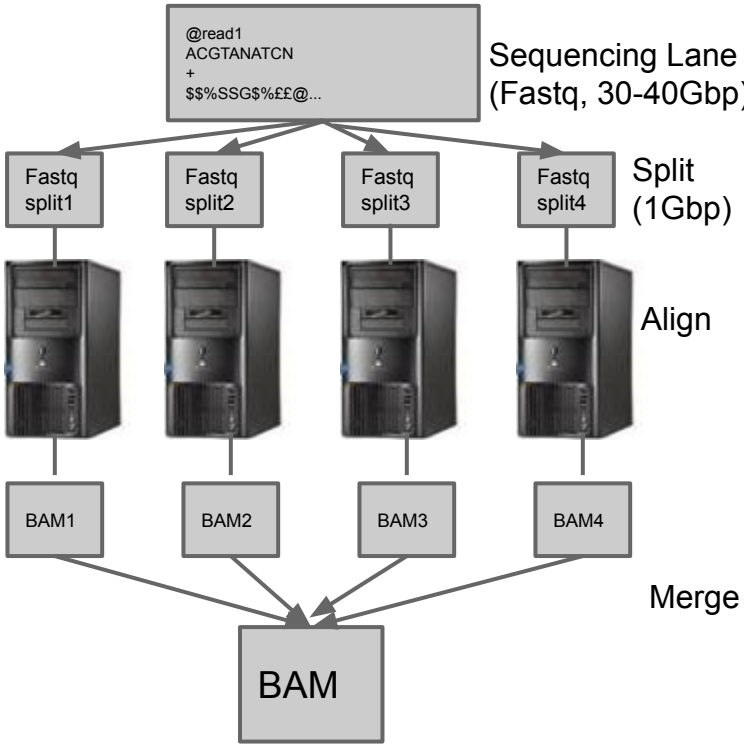
- A form of computation in which many calculations are carried out simultaneously



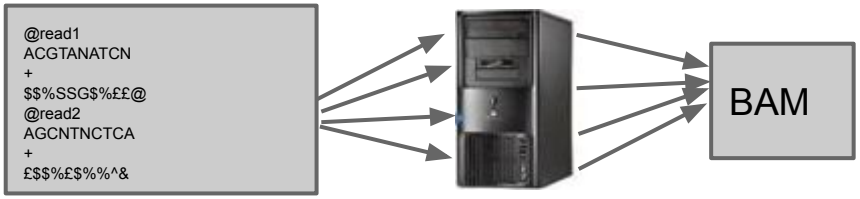
Alignment - Scaling Up

Two main approaches to speeding up read alignment

- Simple parallelism by splitting the data
  - Split lane into 1Gbp chunks and align independently on different processors
    - BWA ~8 hours per 1Gbp chunk
  - Merge chunk BAM files back into single lane BAM
    - 'samtools merge' command



- 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**)



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 components 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