Next generation 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

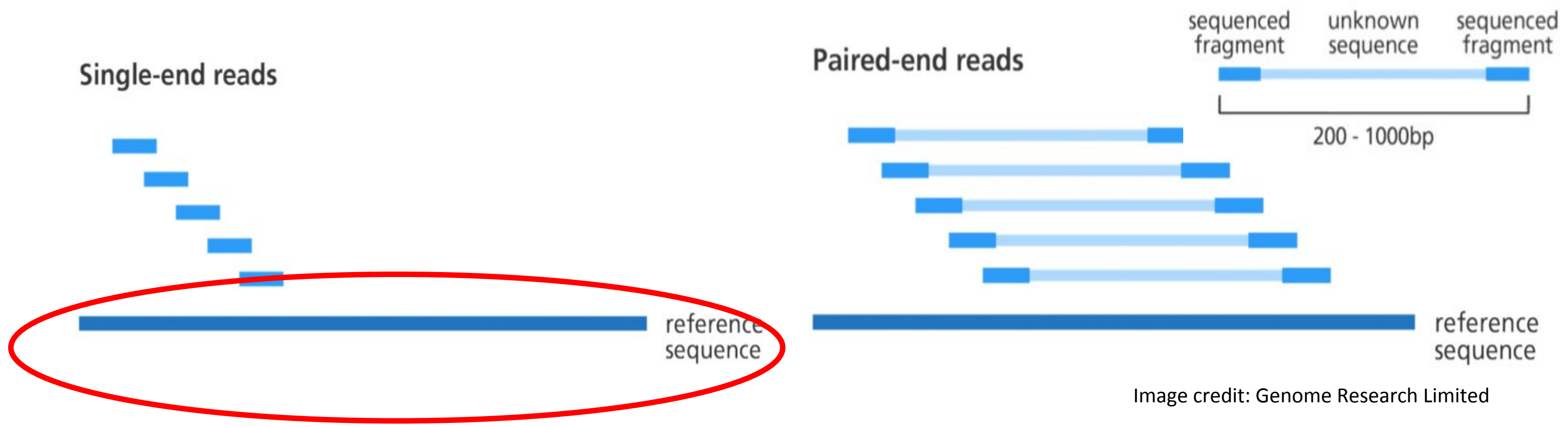


Image credit: Genome Research Limited

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?

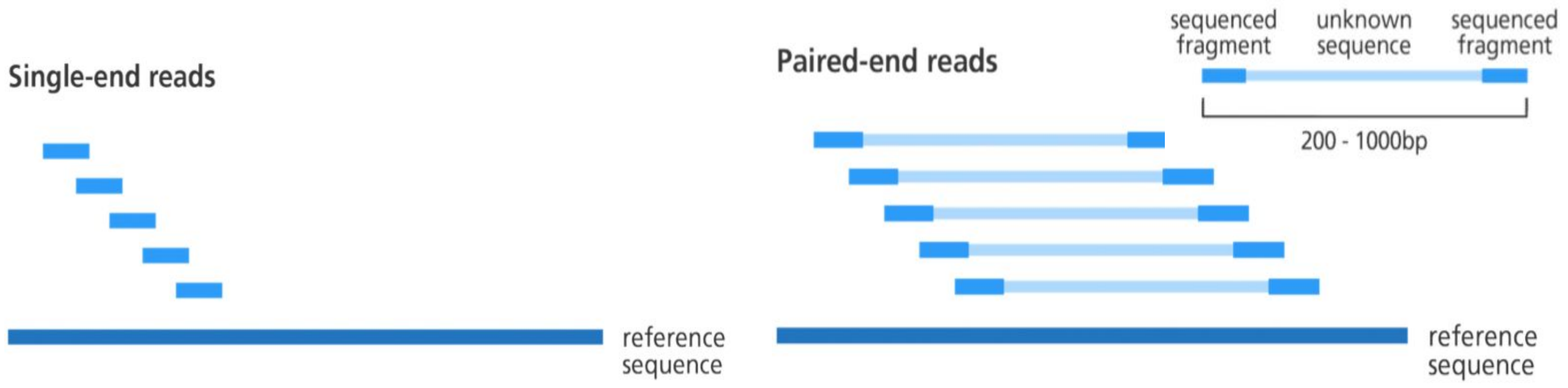
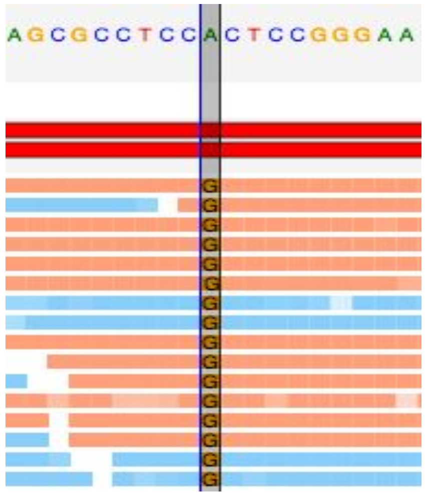
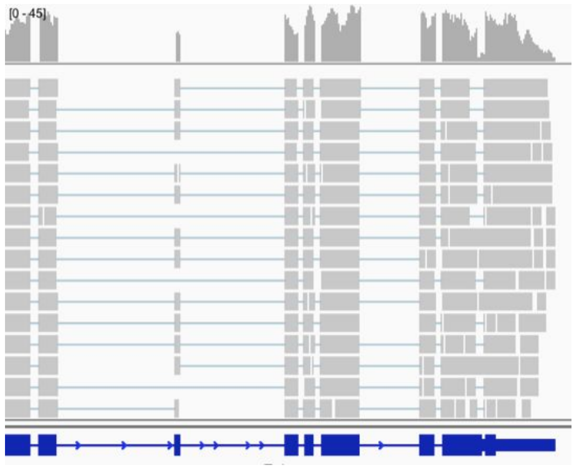


Image credit: Genome Research Limited

Align DNA: Identify variation



ALIGN RNA: Transcript abundance



Sequence Alignment

Sequence alignment in NGS is

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

Principles and approaches to sequence alignment 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
- Nucleotide blast, protein blast, blastx, tblastn, tblastx….

NGS: Nucleotide based alignment

- Very small evolutionary distances (human-human, or related strains of the reference genome)
- Allows for 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 a challenge that has existed since the inception of bioinformatics

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

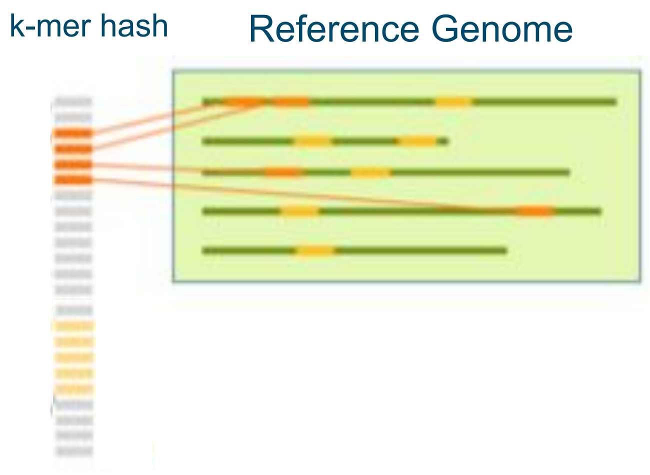
Hash of the reads: MAQ, ELAND, ZOOM and SHRiMP

- Smaller but more variable memory requirements

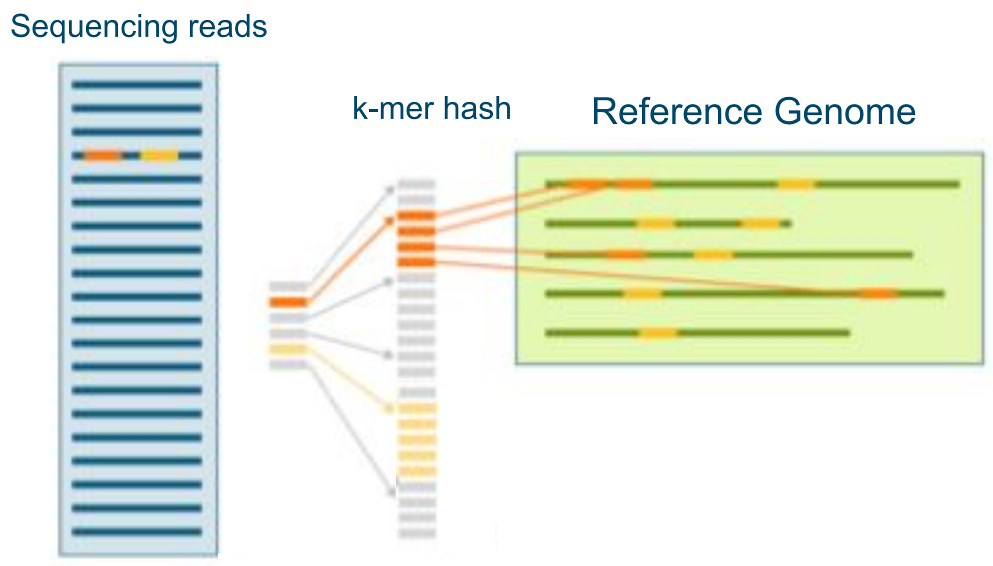
Hash the reference: SOAP, BFAST and MOSAIK

- Advantage: constant memory cost

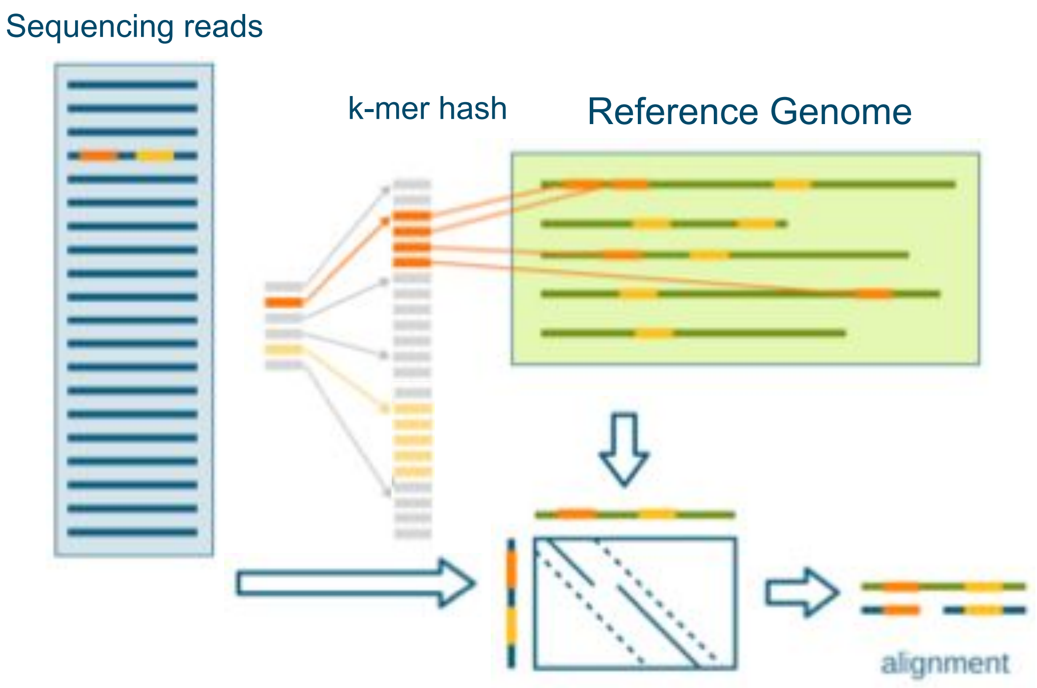
Hash Table Alignment



Hash Table Alignment



Hash Table Alignment



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

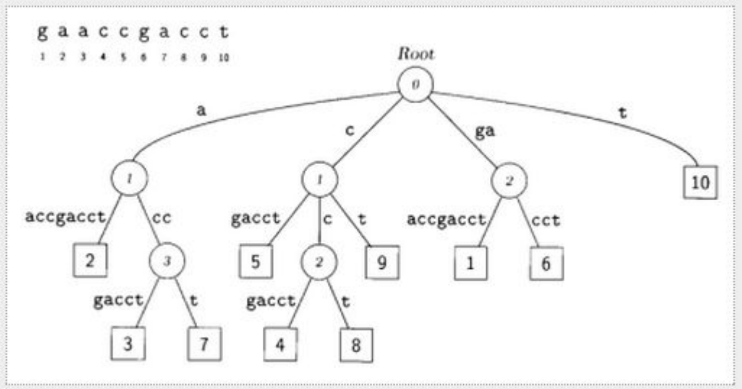


Figure 2
Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. In this example the longest repeated sequence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

Still require a final step to generate local alignment

Delcher et al (1999) *NAR*

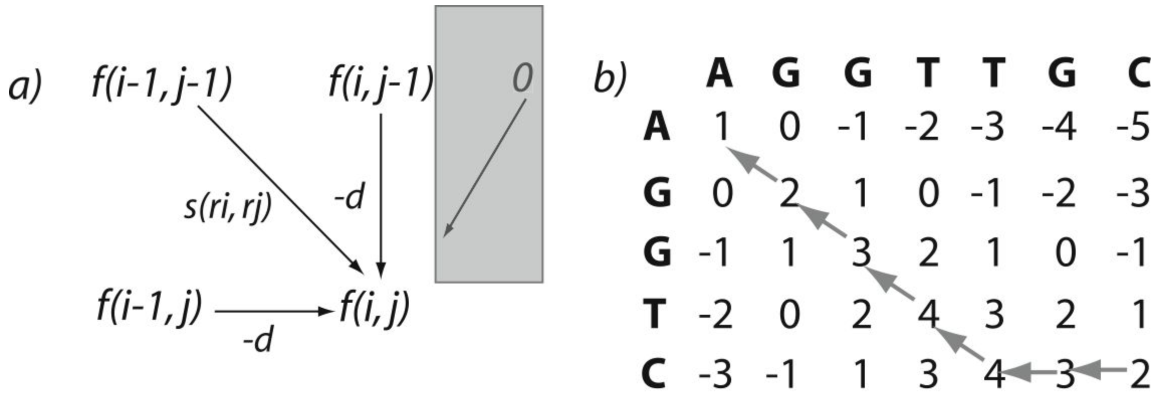
Local Alignment: Smith-Waterman Algorithm (1981)

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



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

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

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

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

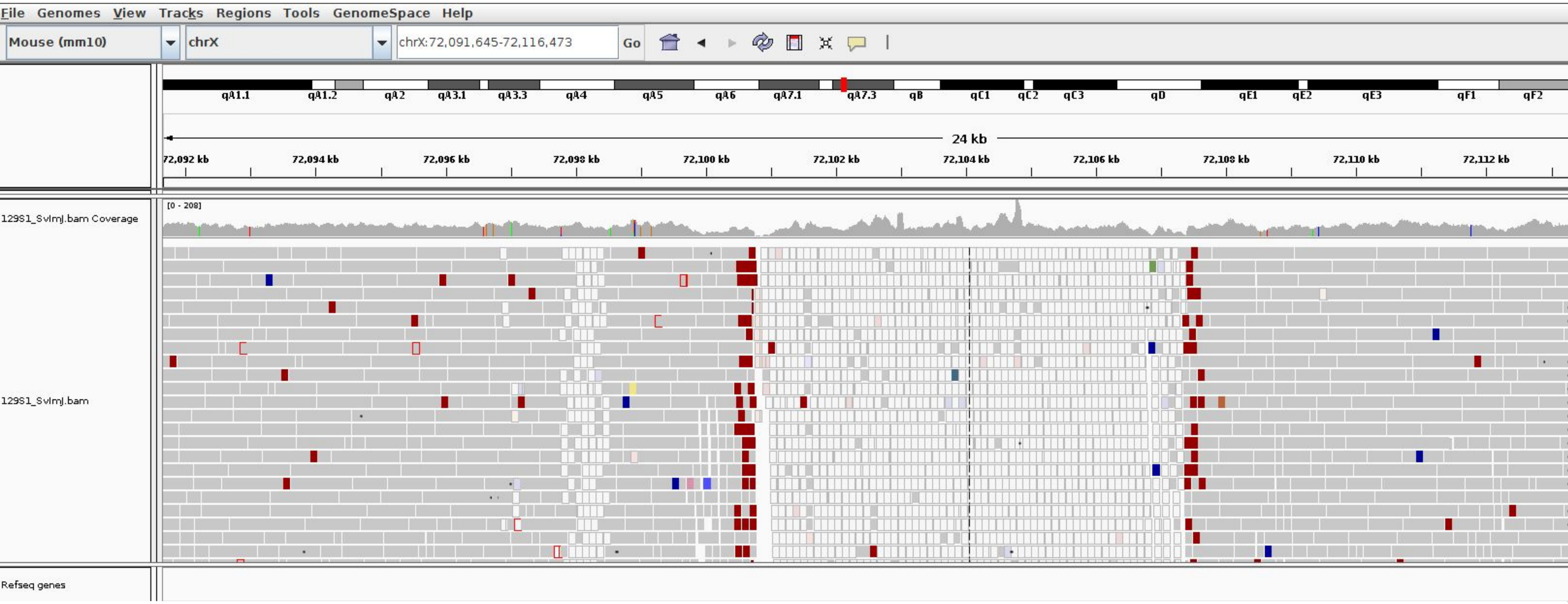
Typically represented as a phred score (log scale)

- Q10 = 1 in 10 incorrect
- Q20 = 1 in 100 incorrect
- Q0 used to indicate no confidence in the alignment of the read to this location

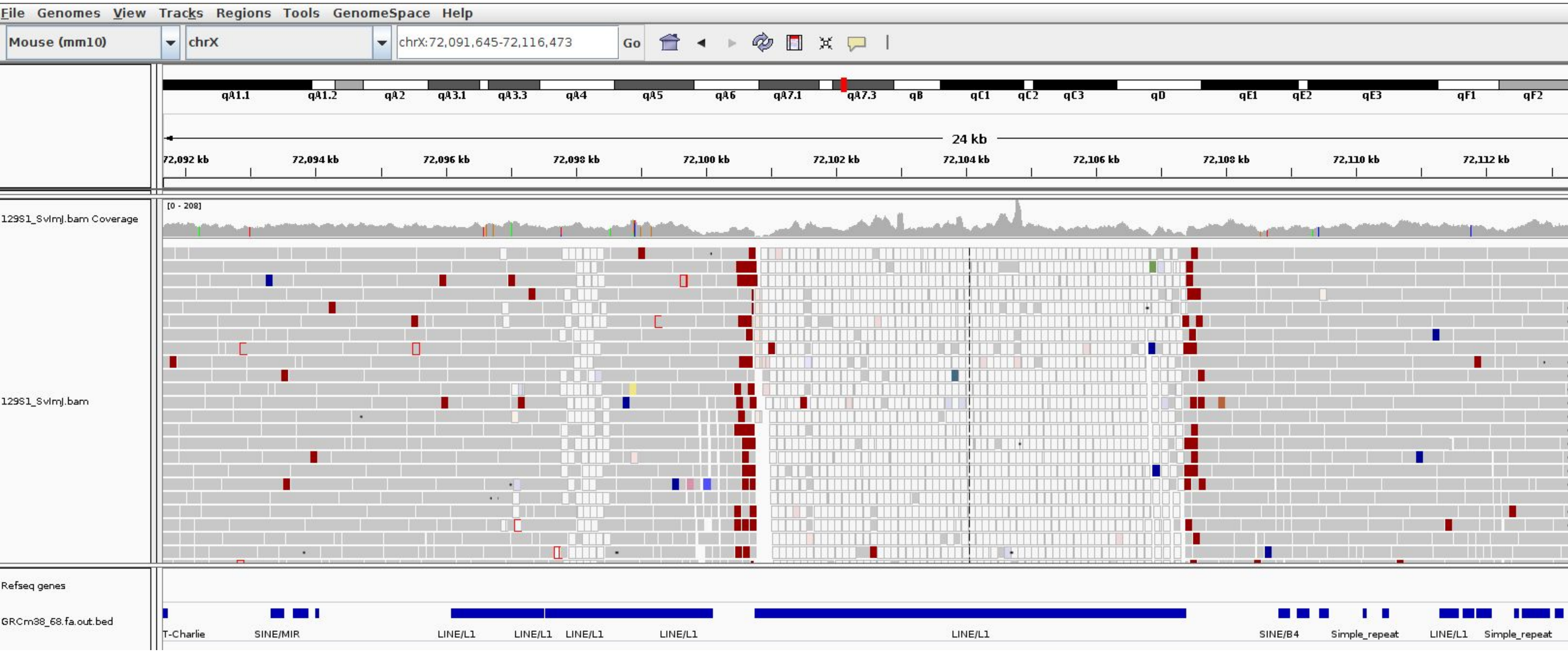
Paired-end sequencing is useful

- 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

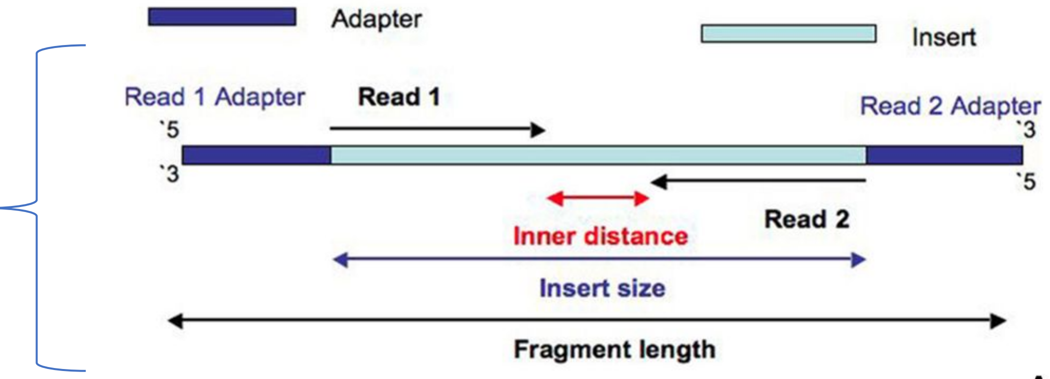

File Genomes View Tracks Regions Tools GenomeSpace Help
Mouse (mm10)
chrX
chrX:72,091,645-72,116,473
Go
qA1.1 qA1.2 qA2 qA3.1 qA3.3 qA4 qA5 qA6 qA7.1 qA7.3 qB qC1 qC2 qC3 qD qE1 qE2 qE3 qF1 qF2
24 kb
72,092 kb 72,094 kb 72,096 kb 72,098 kb 72,100 kb 72,102 kb 72,104 kb 72,106 kb 72,108 kb 72,110 kb 72,112 kb
[0 - 208]
129S1_SvImJ.bam Coverage
129S1_SvImJ.bam
Refseq genes

Mapping Qualities

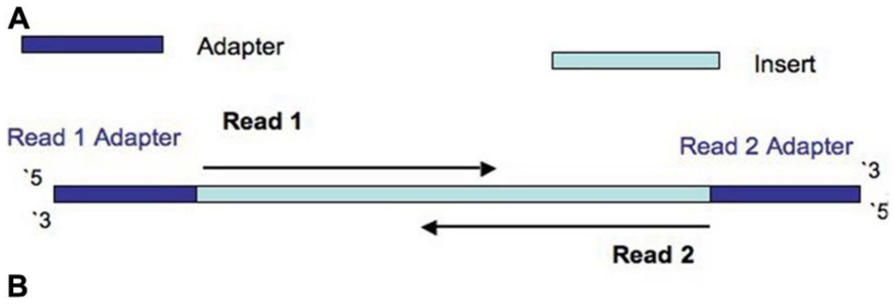

File Genomes View Tracks Regions Tools GenomeSpace Help
Mouse (mm10)
chrX
chrX:72,091,645-72,116,473
Go
qA1.1 qA1.2 qA2 qA3.1 qA3.3 qA4 qA5 qA6 qA7.1 qA7.3 qB qC1 qC2 qC3 qD qE1 qE2 qE3 qF1 qF2
24 kb
72,092 kb 72,094 kb 72,096 kb 72,098 kb 72,100 kb 72,102 kb 72,104 kb 72,106 kb 72,108 kb 72,110 kb 72,112 kb
129S1_SvImJ.bam Coverage
[0 - 208]
129S1_SvImJ.bam
Refseq genes
GRCm38_68.fa.out.bed
T-Charlie SINE/MIR LINE/L1 LINE/L1 LINE/L1 LINE/L1 LINE/L1 SINE/B4 Simple_repeat LINE/L1 Simple_repeat

Some Terminology

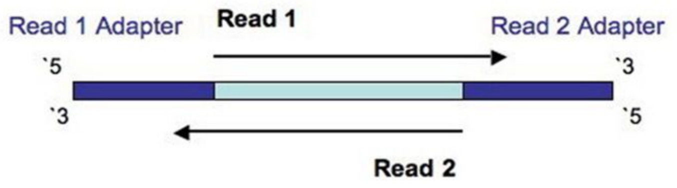
Insert size > length(read 1 + read 2)



Insert size < length(read 1 + read 2)



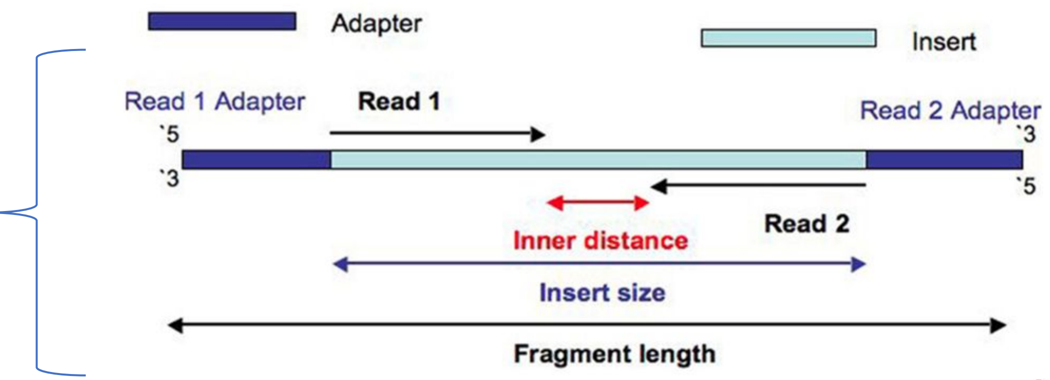
Insert size < length(read 1);
Insert size < length(read 2)



Turner, 2014. PMID:24523726

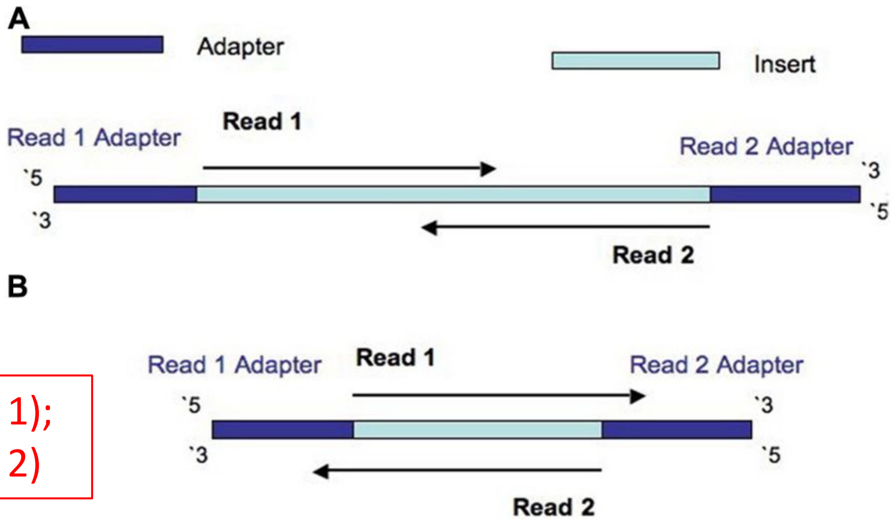
Some Terminology

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

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

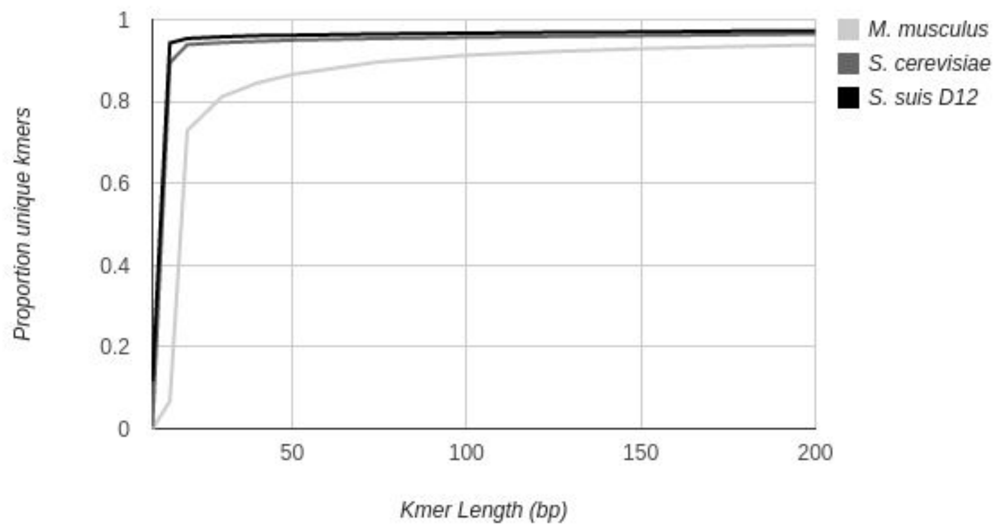
Alignment around indels

- 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
- 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

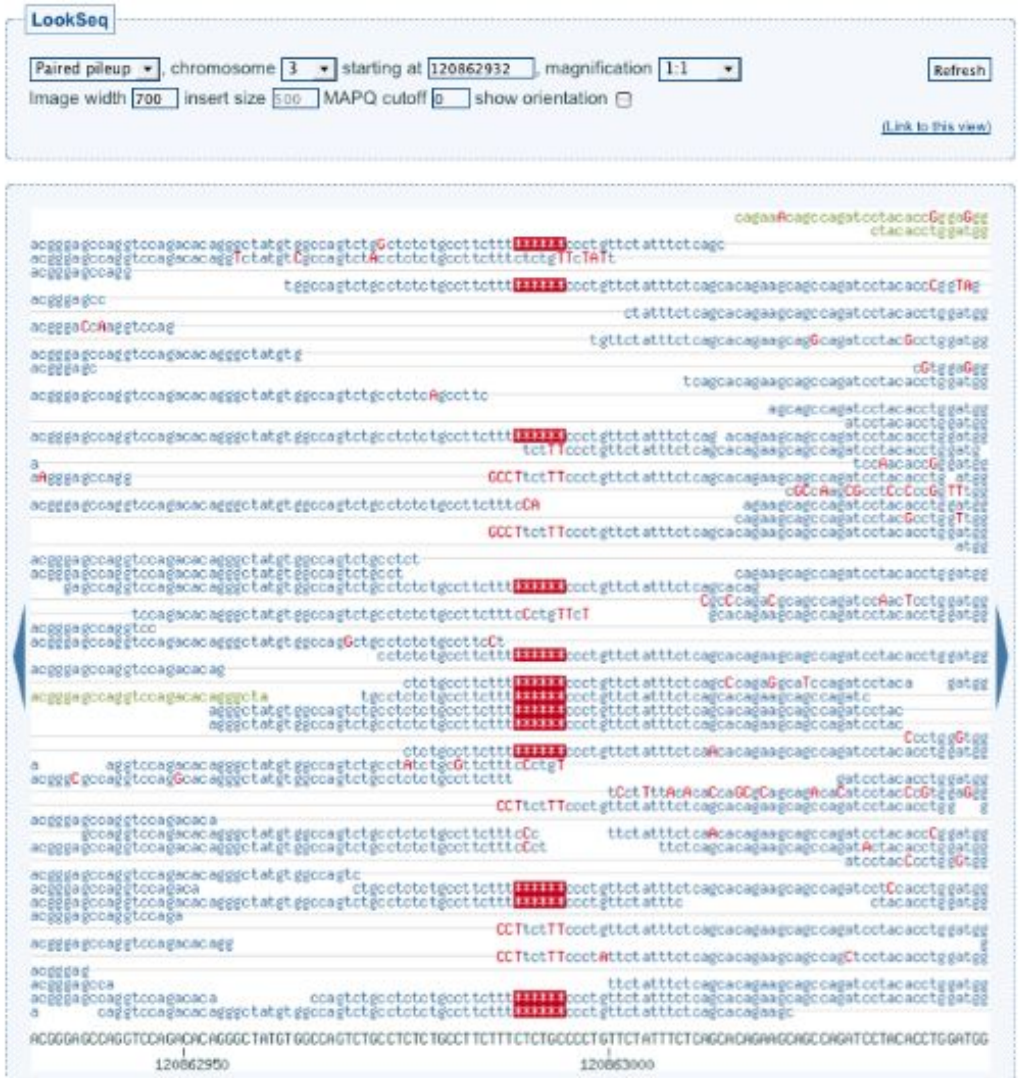
- Seed and extend based aligners can have an upper limit on the number of consecutive SNPs in seed region of read (e.g. Maq - max of 2 mismatches in first 28bp of read)
- BWT based aligners work best at low divergence

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



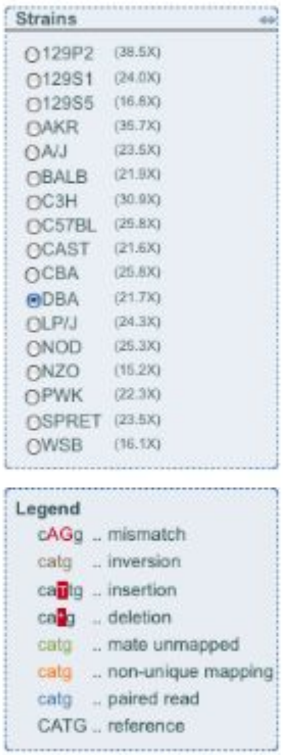
LookSeq

Paired pileup, chromosome 3 starting at 120862932, magnification 1:1 Refresh

Image width 700 insert size 500 MAPQ cutoff 0 show orientation ☐

(Link to this view)

```
ACGGGAGCCAGGTCCAGACACAGGGCTATGTGGCCAGTCTGCCTCTCTGCCTTCTTTCTCTGCCCCTGTTCTATTTCTCAGCACAGAAGCAGCCAGATCCTACACCTGGATGG
                120862950                                          120863000
```



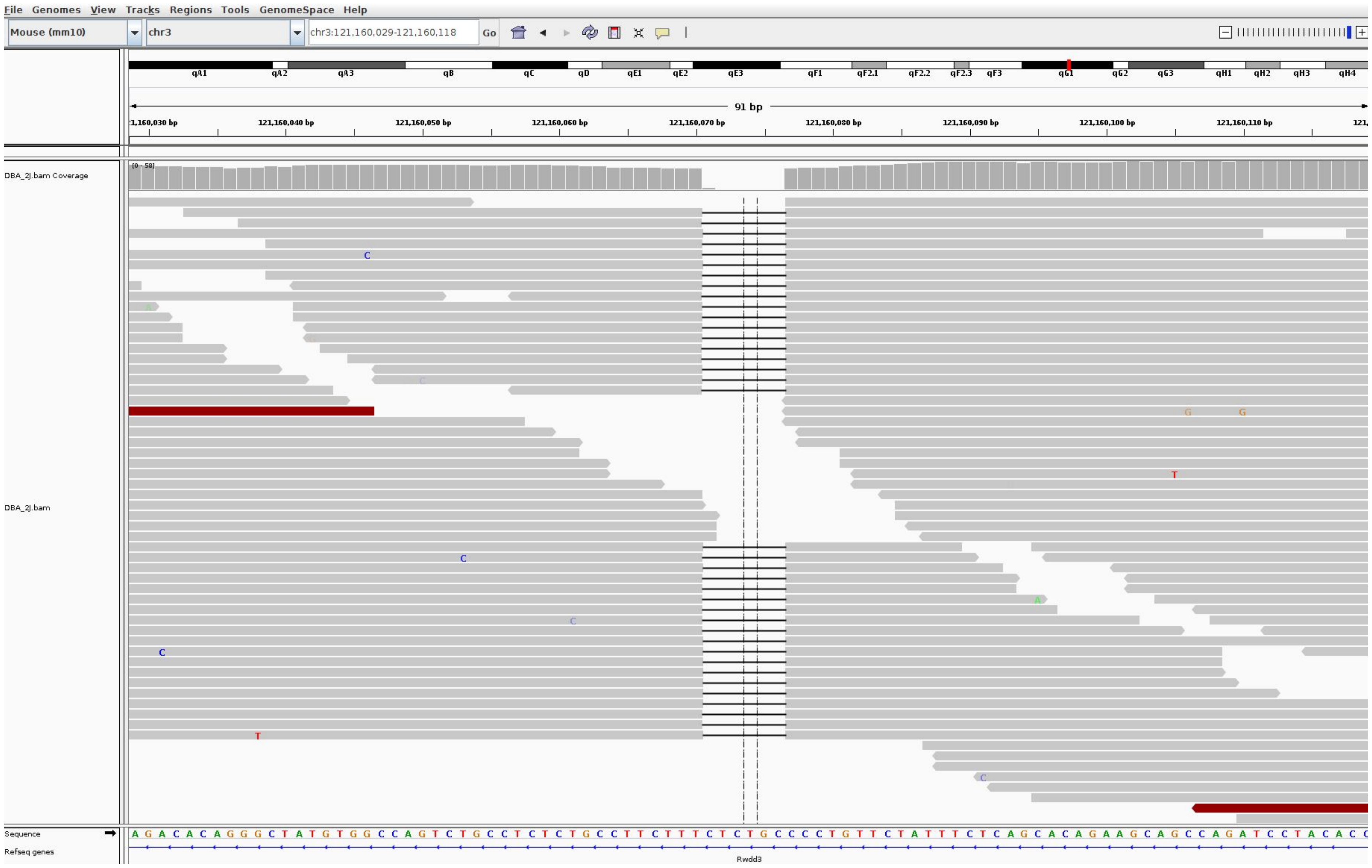
Strains

○129P2	(38.5X)
○129S1	(24.0X)
○129S5	(16.6X)
○AKR	(35.7X)
○A/J	(23.5X)
○BALB	(21.9X)
○C3H	(30.9X)
○C57BL	(25.8X)
○CAST	(21.6X)
○CBA	(25.8X)
●DBA	(21.7X)
○LP/J	(24.3X)
○NOD	(25.3X)
○NZO	(15.2X)
○PWK	(22.3X)
○SPRET	(23.5X)
○WSB	(16.1X)

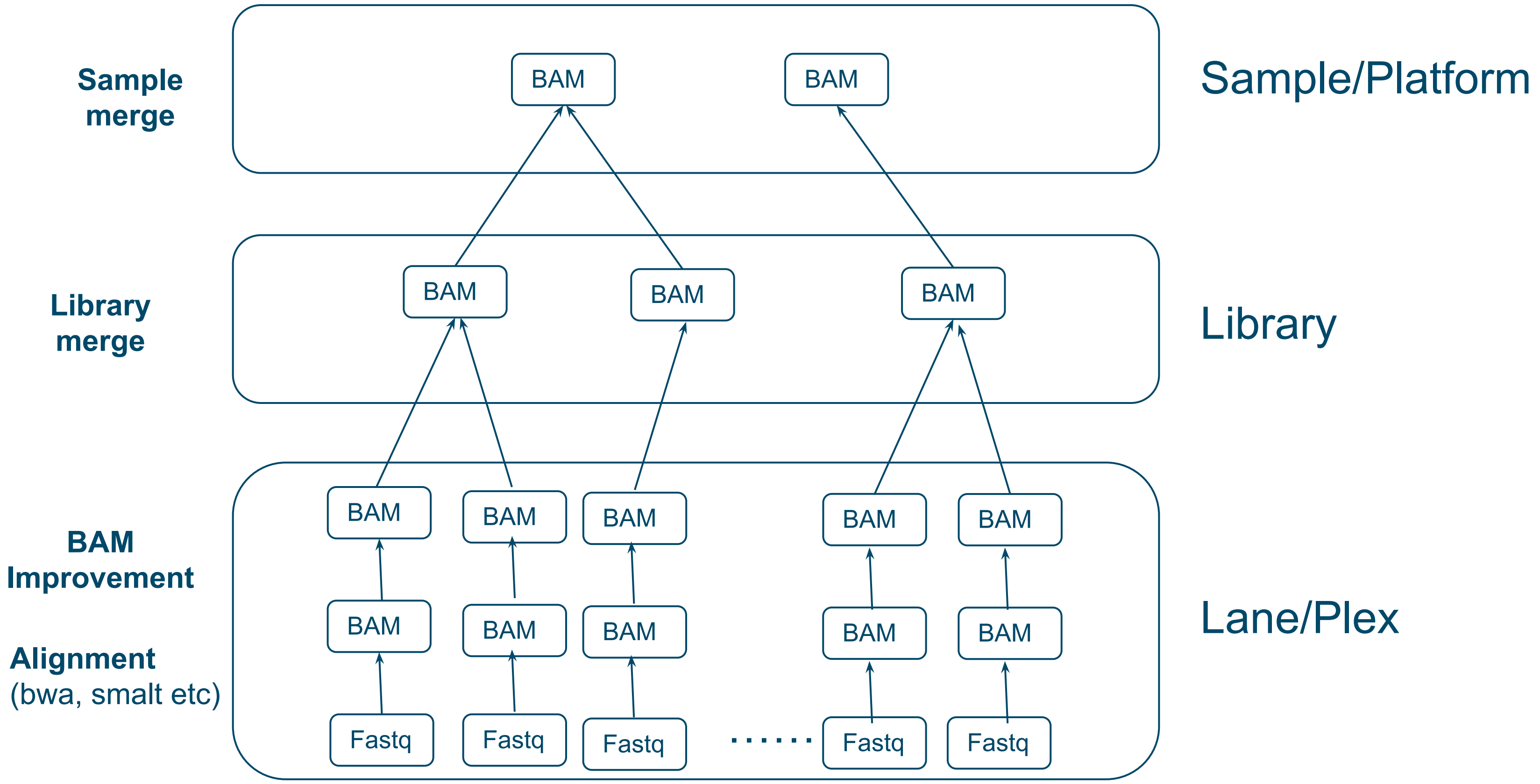
Legend

cAGg	.. mismatch
catg	.. inversion
catttg	.. insertion
ca▇g	.. deletion
catg	.. mate unmapped
catg	.. non-unique mapping
catg	.. paired read
CATG	.. reference

Same Indel - 2014


File Genomes View Tracks Regions Tools GenomeSpace Help
Mouse (mm10)
chr3
chr3:121,160,029-121,160,118
Go
qA1 qA2 qA3 qB qC qD qE1 qE2 qE3 qF1 qF2.1 qF2.2 qF2.3 qF3 qG1 qG2 qG3 qH1 qH2 qH3 qH4
91 bp
121,160,030 bp
121,160,040 bp
121,160,050 bp
121,160,060 bp
121,160,070 bp
121,160,080 bp
121,160,090 bp
121,160,100 bp
121,160,110 bp
DBA_2J.bam Coverage
[0 - 58]
DBA_2J.bam
Sequence
A G A C A C A G G G C T A T G T G G C C A G T C T G C C T C T C T G C C T T C T T T C T C T G C C C C T G T T C T A T T T C T C A G C A C A G A A G C A G C C A G A T C C T A C A C C
Refseq genes
Rwdd3

Data Production Workflow



Base Quality 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

Illumina sequencing

- Control lane or spiked control used to generate a quality calibration table
- If no control – then use pre-computed calibration tables

Quality recalibration

- 1000 genomes project sequencing carried out on multiple platforms at multiple different sequencing centres
- Are the quality values comparable across centres/platforms given they have all been calibrated using different methods?

Base Quality Recalibration

Original recalibration algorithm

- Align subsample of reads from a lane to human reference
- Exclude all known dbSNP+1000G pilot SNP sites
  - Assume all other mismatches are sequencing errors
- Compute a new calibration table bases on mismatch rates per position on the read

Pre-calibration sequence reports Q25 base calls

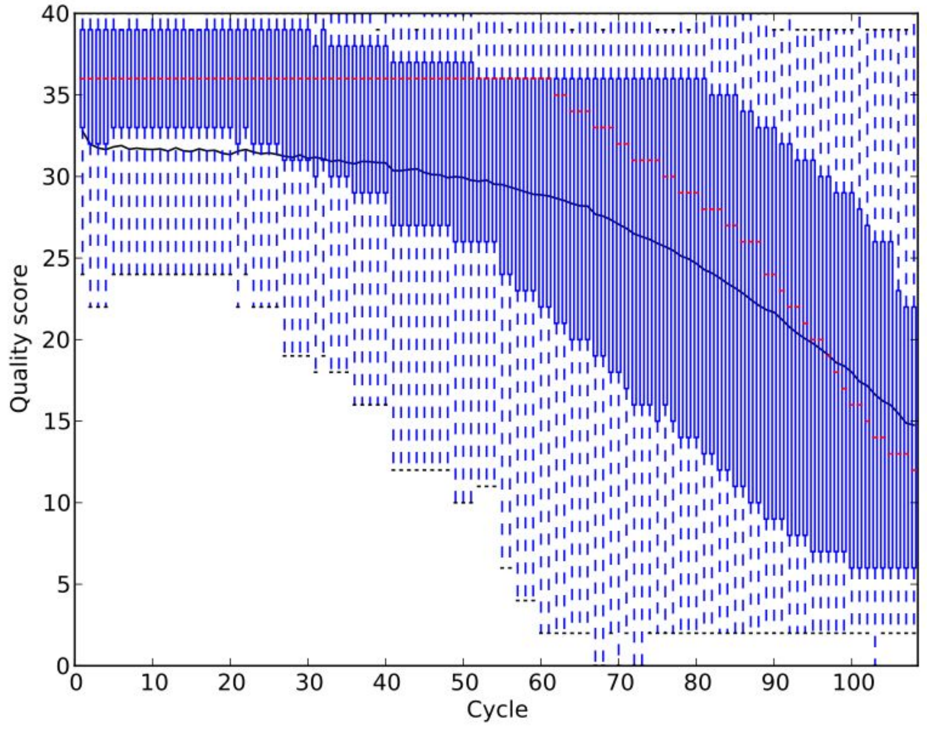
- After alignment - it may be that these bases actually mismatch the reference at a 1 in 100 rate, so are actually Q20

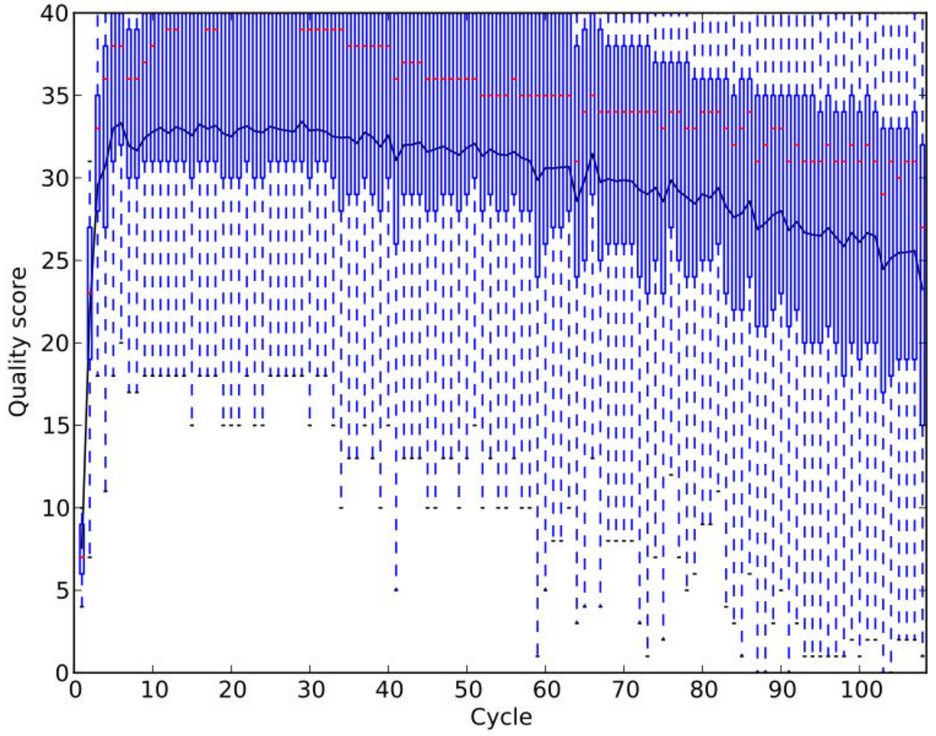
Recent improvements – GATK package

- Reported/original quality score
- The position within the read
- The preceding and current nucleotide (sequencing chemistry effect) observed by the sequencing machine
- Probability of mismatching the reference genome

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

Base Quality Recalibration Effects





N.B. Always replot quality values when trying BQSR on a new set of samples or species

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

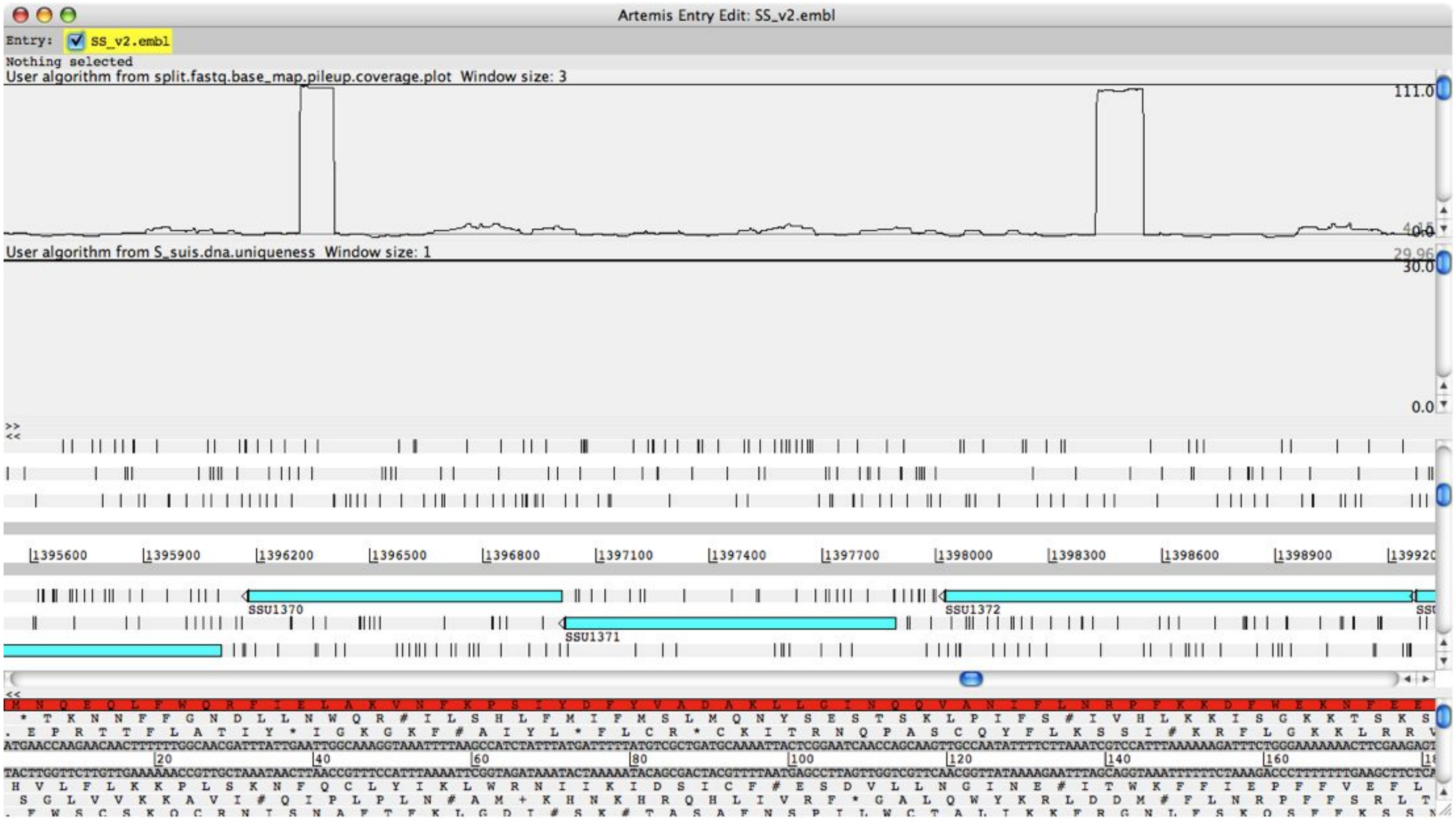
Generally low number of duplicates in good libraries (<5%)

- 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 or samtools rmdupse
  - GATK: MarkDuplicates

Can result in false SNP calls

- Duplicates manifest themselves as high read depth support

Library Duplicates



Artemis Entry Edit: SS_v2.embl

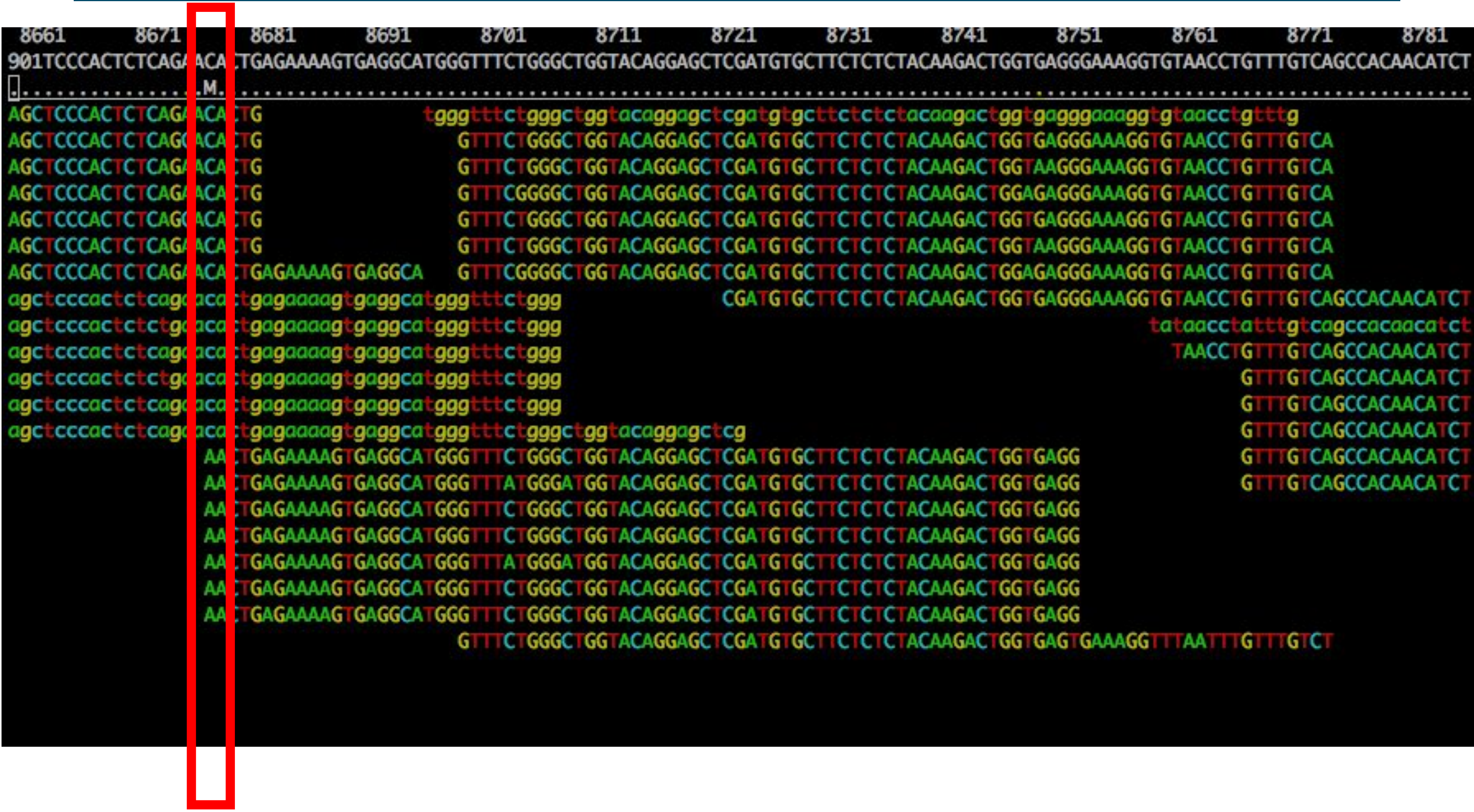
Entry: SS_v2.embl

Nothing selected

User algorithm from split.fastq.base_map.pileup.coverage.plot Window size: 3

User algorithm from S_suis.dna.uniqueness Window size: 1

Duplicates and False SNPs



```
 8661      8671      8681      8691      8701      8711      8721      8731      8741      8751      8761      8771      8781
901TCCCACTCTCAGAACACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
.................M.............................................................................................................
AGCTCCCACTCTCAGAACACTG              tgggtttctgggctggtacaggagctcgatgtgcttctctctacaagactggtgagggaaaggtgtaacctgtttg
AGCTCCCACTCTCAGGACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGGACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTGAGAAAAGTGAGGCA   GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
agctcccactctcagaacactgagaaaagtgaggcatgggtttctggg              CGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
agctcccactctctgaacactgagaaaagtgaggcatgggtttctggg                                                   tataacctatttgtcagccacaacatct
agctcccactctcagaacactgagaaaagtgaggcatgggtttctggg                                                     TAACCTGTTTGTCAGCCACAACATCT
agctcccactctctgaacactgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcagaacactgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcagaacactgagaaaagtgaggcatgggtttctgggctggtacaggagctcg                                           GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                                       GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGTGAAAGGTTTAATTTGTTTGTCT
```

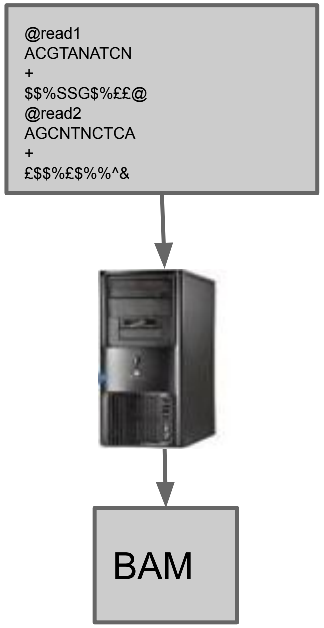
Alignment - Scaling Up

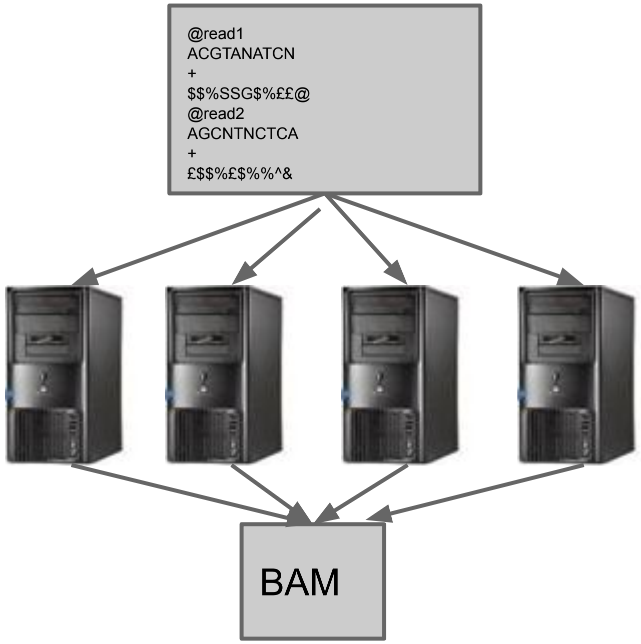
50-60 Gbp per HiSeq lane

- 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



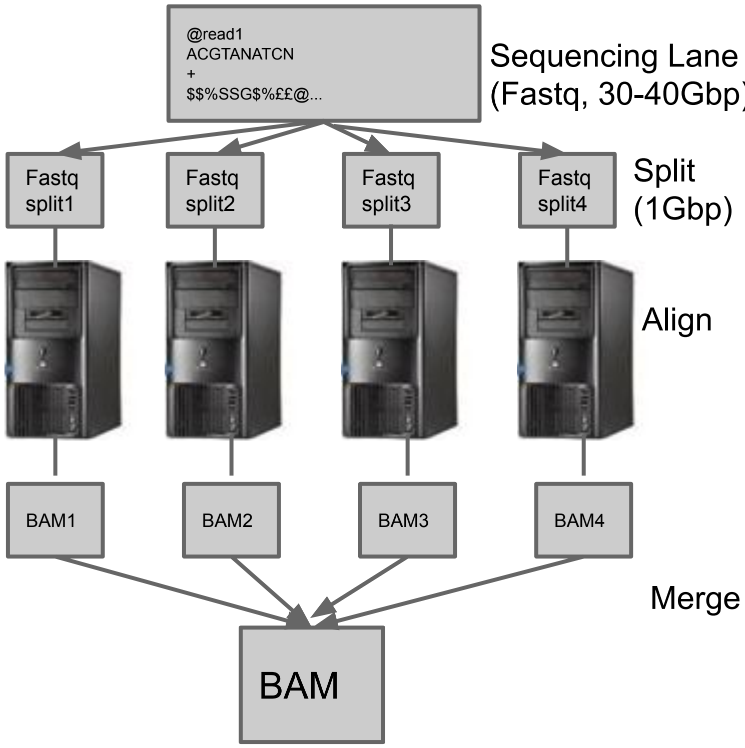


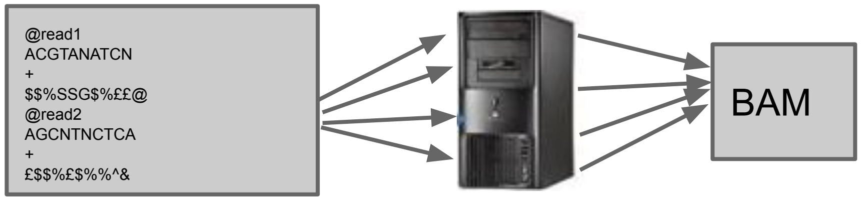
Alignment - Scaling Up

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 -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