NGS: Read Mapping to Reference Genome

Daniel Sobral / José Ferrão

Ministério da Saúde

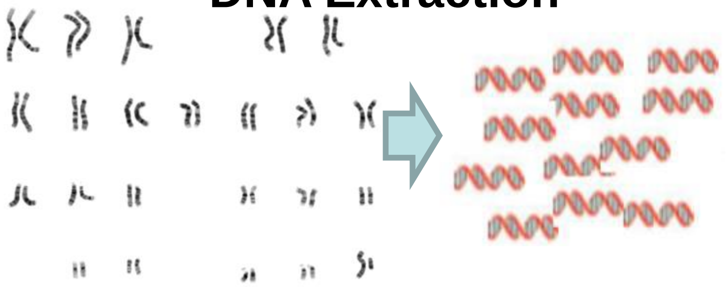
Instituto_Nacional de Saúde
Doutor Ricardo Jorge

Learning Objectives

- Overal understanding of the data preprocessing steps
  - Alignment of NGS reads (fastq) to a reference sequence
    - The SAM / BAM format
    - Secondary and supplementary alignments
  - Marking Duplicates
  - Base Quality Recalibration
- Visualizing results and assessing quality

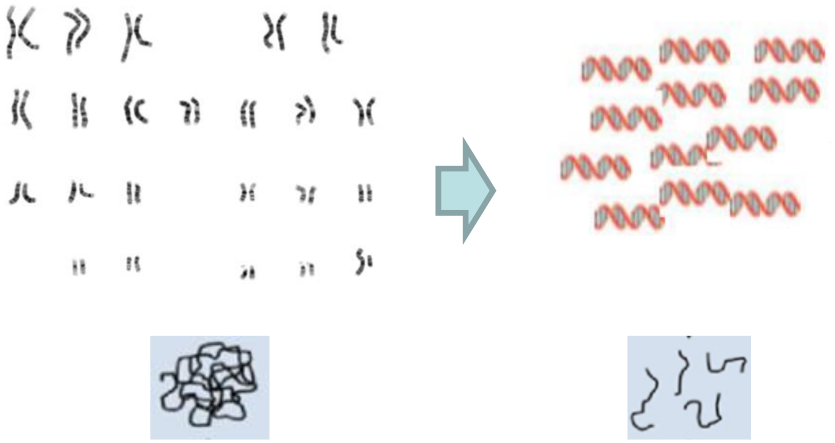
Data pre-processing for variant discovery

DNA Extraction

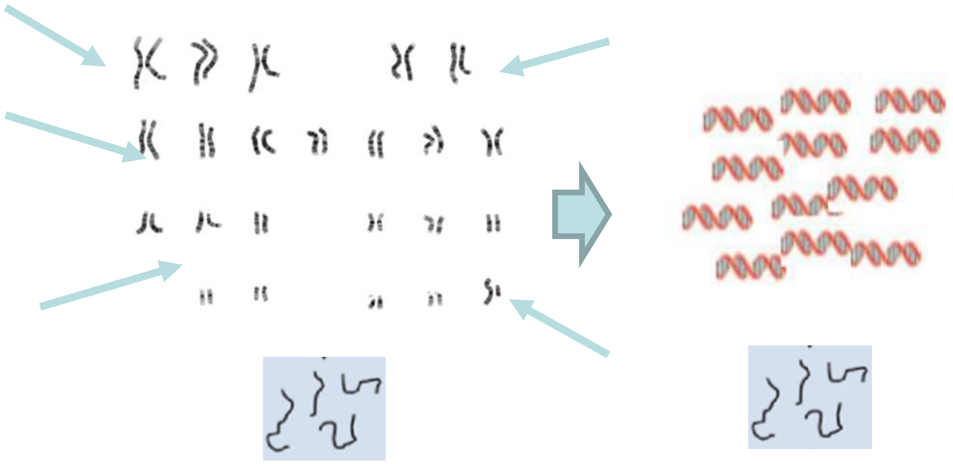


Data pre-processing for variant discovery

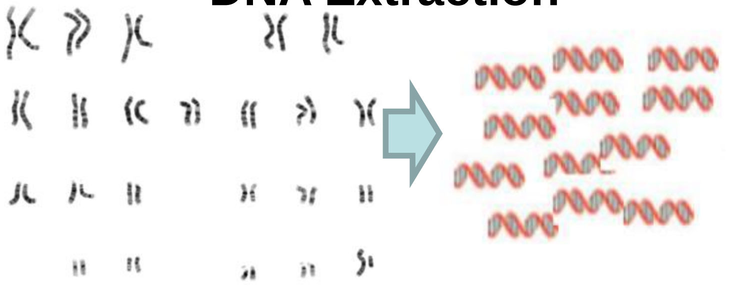
Whole Genome



Targeted (eg. WES)



DNA Extraction



Eg. TruSight One Enrichment Panel

How to calculate coverage

- WGS 30x coverage, 150bp read pairs
  - 30 x $3x10^9$ / (150x2) = $3x10^8$ = 300 million read pairs
- WES (~1-2% genome) 30x coverage, 150bp read pairs
  - 6 million read pairs (theoretical minimum, but usually more)

Genome (WGS) vs Exome (WES)

Advantages of WES over WGS

- WES only focus on potentially actionable regions
  - WES is more economic and less compute intensive

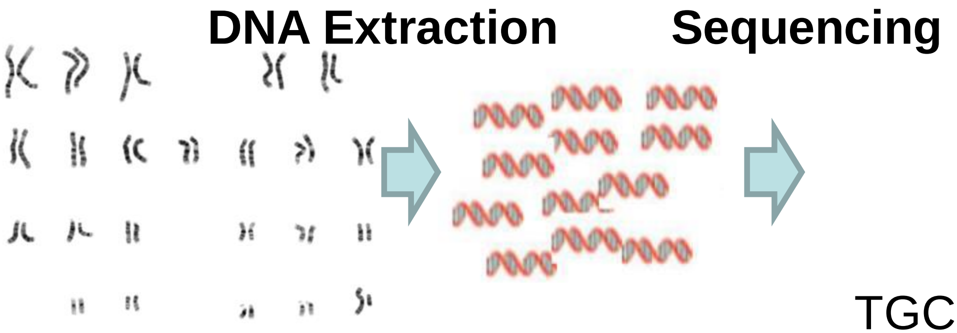
Disadvantages of WES

- WES can have primer-derived artifacts
  - eg. amplification bias, primer dimers, kit-dependent
- WES is very limited for structural variants

Data pre-processing for variant discovery

Raw Unmapped Reads

uBAM or FASTQ



DNA Extraction

Sequencing

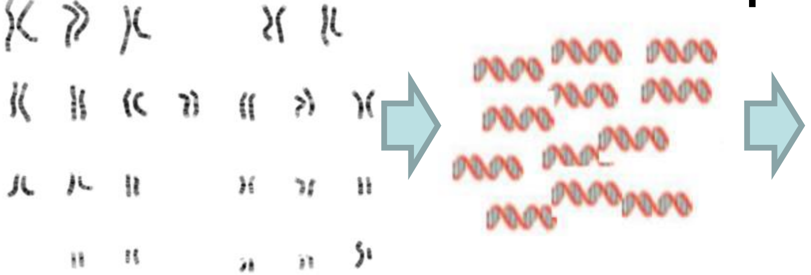
TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT **(“raw” fastQ)**

Data pre-processing for variant discovery

Raw Unmapped Reads

uBAM or FASTQ

DNA Extraction **Sequencing**



TGCTCAGTT TGAC ATGGAGT GTTGT **(“clean” fastQ)**

QC and pre-processing

TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT **(“raw” fastQ)**

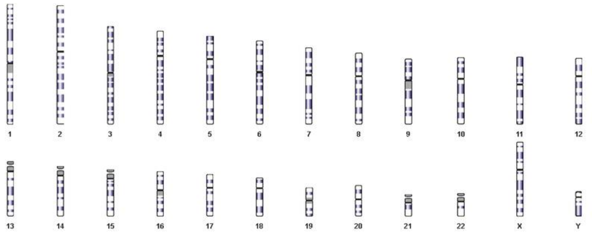
Data pre-processing for variant discovery

Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome



geneA **geneB** **geneC**

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

DNA Extraction **Sequencing**

TGCTCAGTT TGAC ATGGAGT GTTGT **("clean" fastQ)**

QC and pre-processing

TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT **("raw" fastQ)**

What genome reference to use?

- GRCh37 (hg19)
  - The “classical”, for which most resources were built (1000 genomes, etc…)
- **GRCh38 (hg38)**
  - **The recommended one. Most resources have been converted to this now**
- T2T-CHM13
  - Most complete genome, but will take time for resources to be available

Future: pangenome graphs

A lot of investment is going here, but several challenges remain

Mostly relevant for Structural Variants

What genome reference to use?

Things to consider:

- Mask PAR regions in Y chromosome
- Include decoys (eg. EBV sequences)
- Include alternative sequences such as HLA loci
    - Though these may need to be analized specifically

https://gatk.broadinstitute.org/hc/en-us/articles/360035890951-Human-genome-reference-builds-GRCh38-or-hg38-b37-hg19

https://gatk.broadinstitute.org/hc/en-us/articles/360035890951

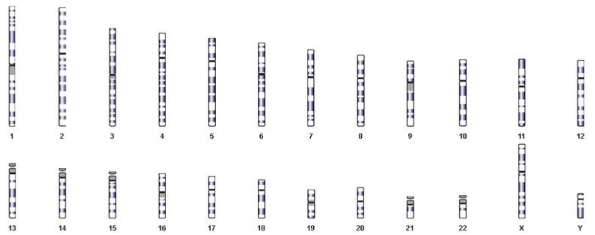
Data pre-processing for variant discovery

Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome



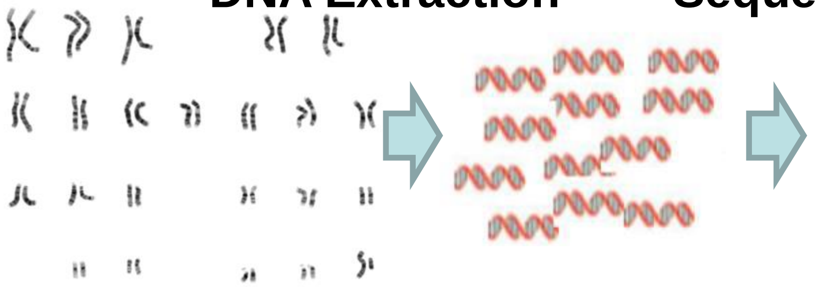
TGCTCAGTT

geneA geneB geneC

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

Alignment

DNA Extraction **Sequencing**



TGCTCAGTT TGAC ATGGAGT GTTGT **(“clean” fastQ)**

QC and pre-processing

TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT **(“raw” fastQ)**

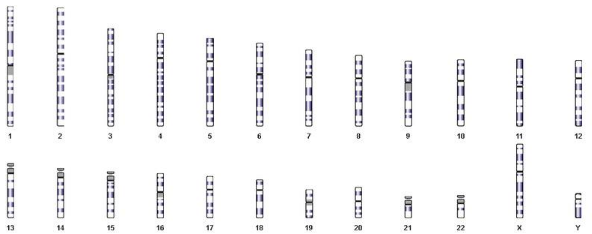
Data pre-processing for variant discovery

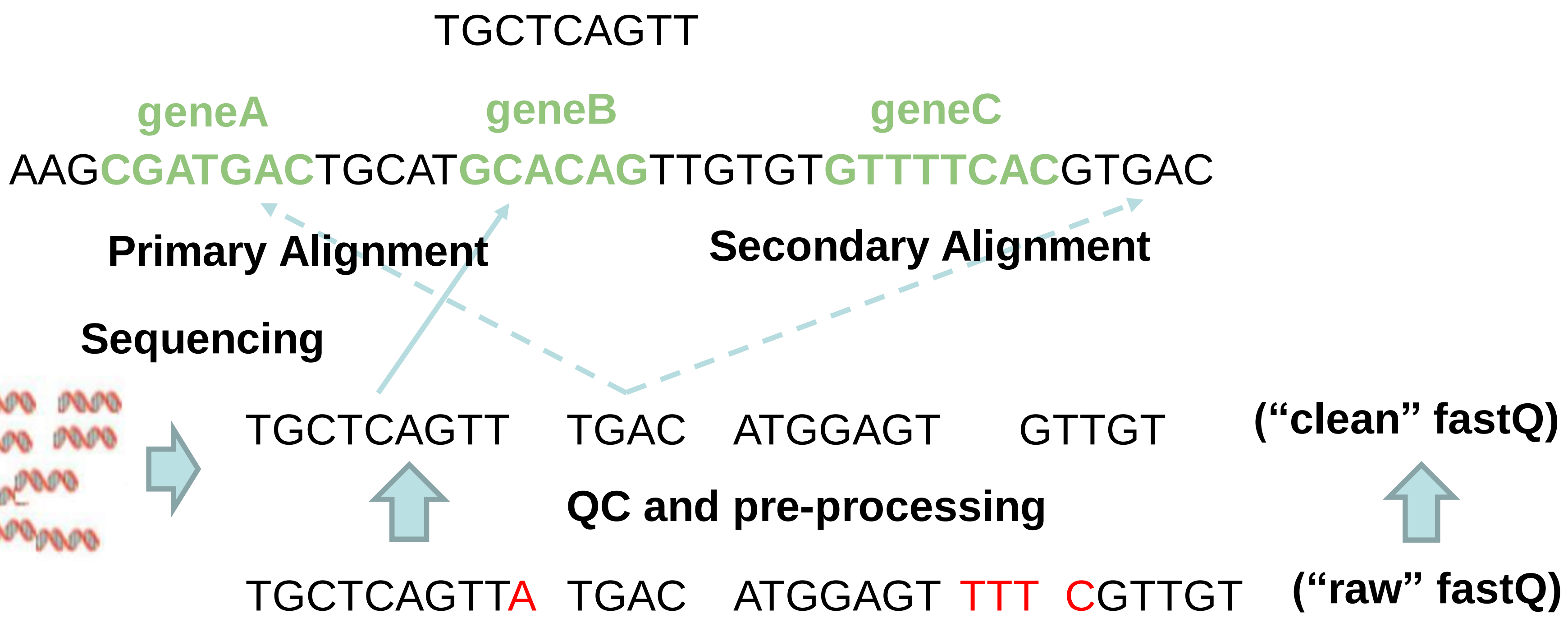
Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome





TGCTCAGTT

geneA **geneB** **geneC**

AAG**CGATGAC**TGCAT**GCACAG**TTGTGT**GTTTTCAC**GTGAC

Primary Alignment

Secondary Alignment

TGCTCAGTT TGAC ATGGAGT GTTGT

(“clean” fastQ)

QC and pre-processing

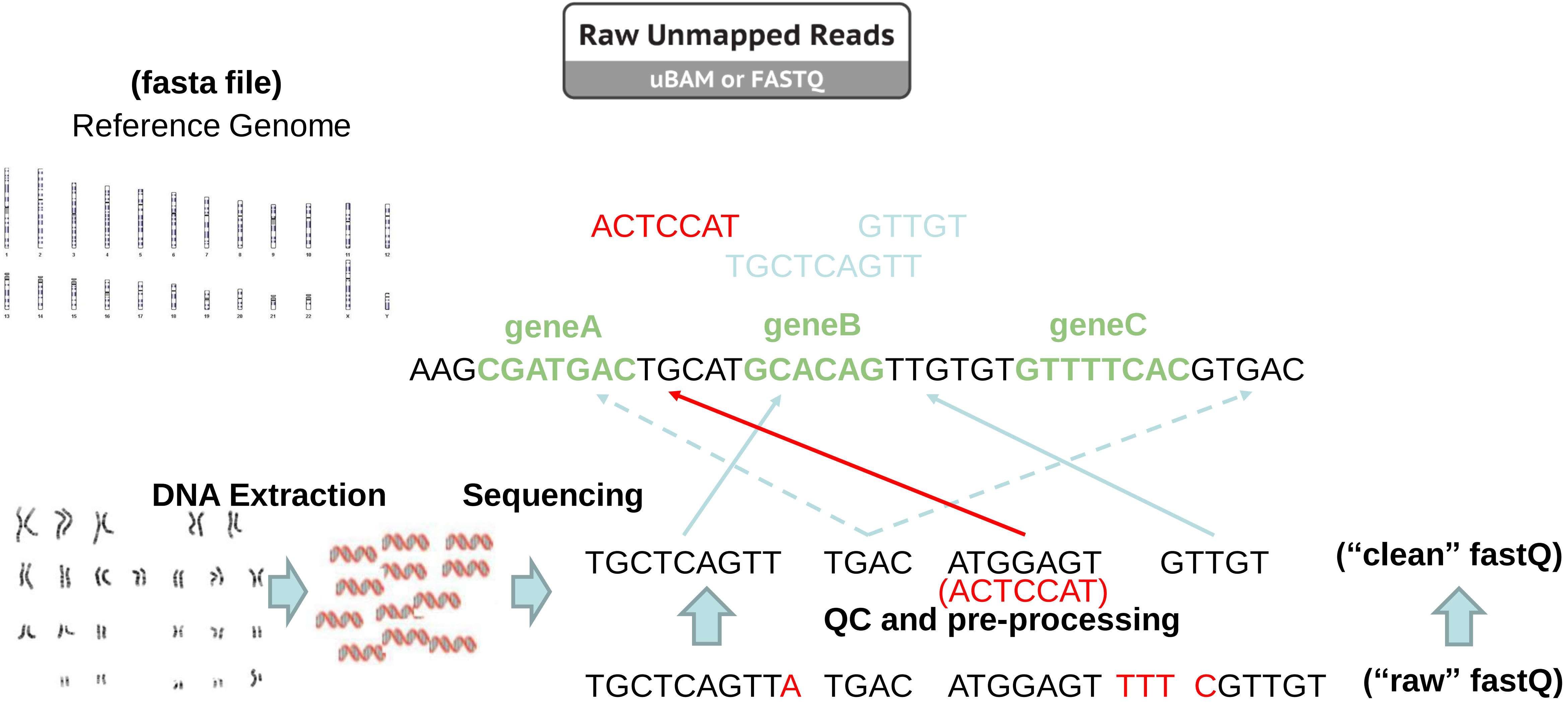
TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT

(“raw” fastQ)

DNA Extraction

Sequencing

Data pre-processing for variant discovery



Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome

ACTCCAT GTTGT

TGCTCAGTT

geneA geneB geneC

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

DNA Extraction

Sequencing

TGCTCAGTT TGAC ATGGAGT GTTGT

(ACTCCAT)

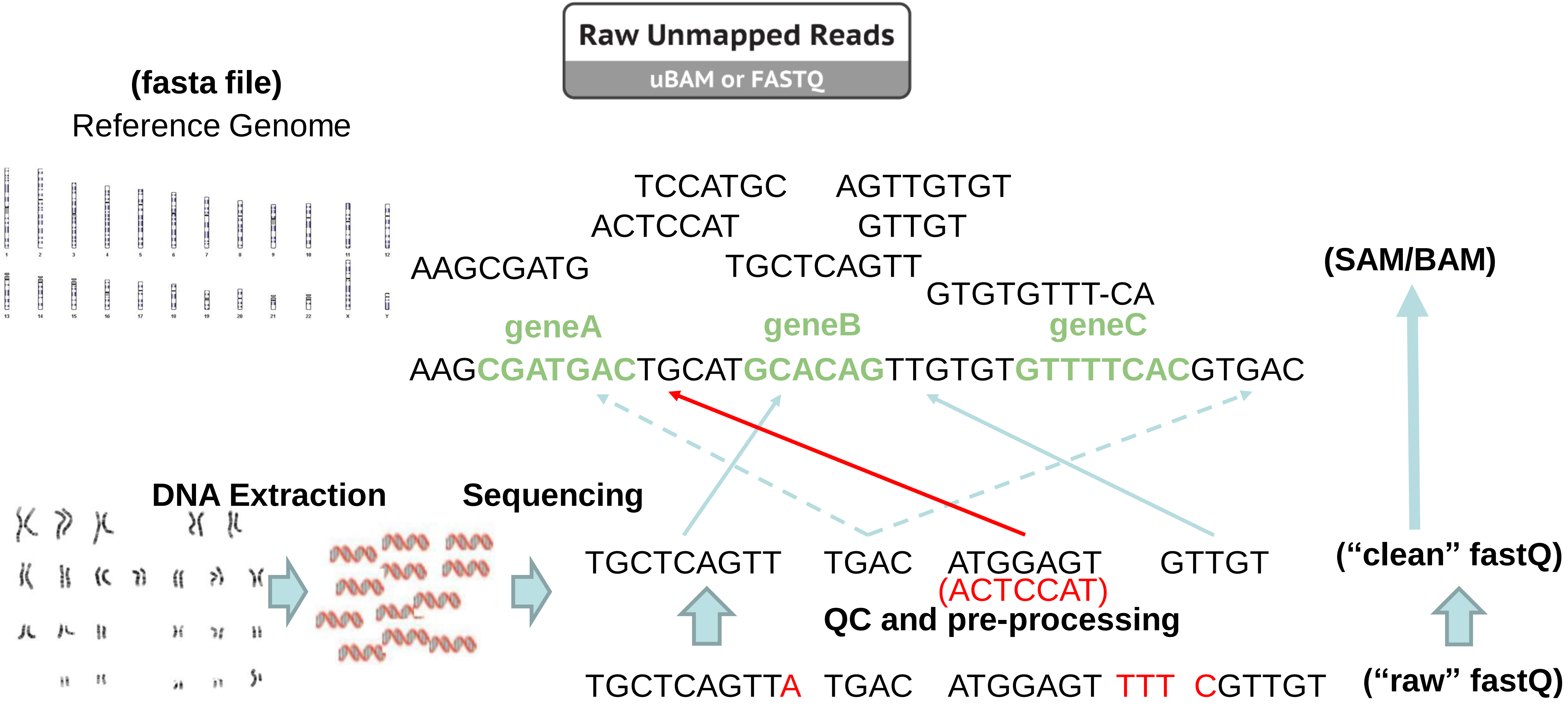
(“clean” fastQ)

QC and pre-processing

TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT

(“raw” fastQ)

Data pre-processing for variant discovery



Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome

TCCATGC AGTTGTGT

ACTCCAT GTTGT

AAGCGATG TGCTCAGTT

GTGTGTTT-CA

(SAM/BAM)

geneA **geneB** **geneC**

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

DNA Extraction

Sequencing

TGCTCAGTT TGAC ATGGAGT GTTGT

(ACTCCAT)

(“clean” fastQ)

QC and pre-processing

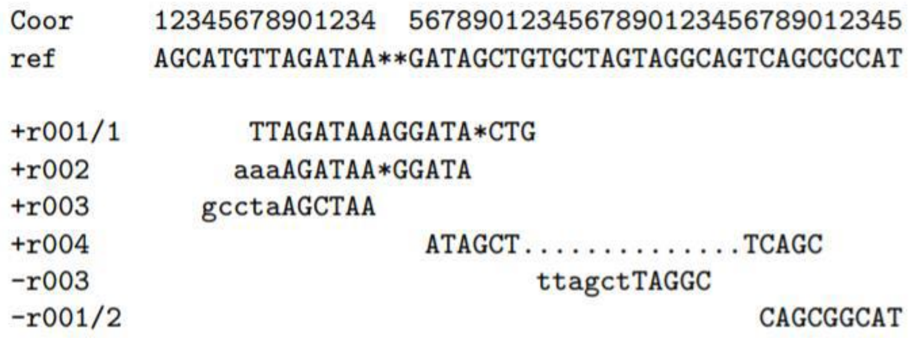
TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT

(“raw” fastQ)

SAM/BAM format

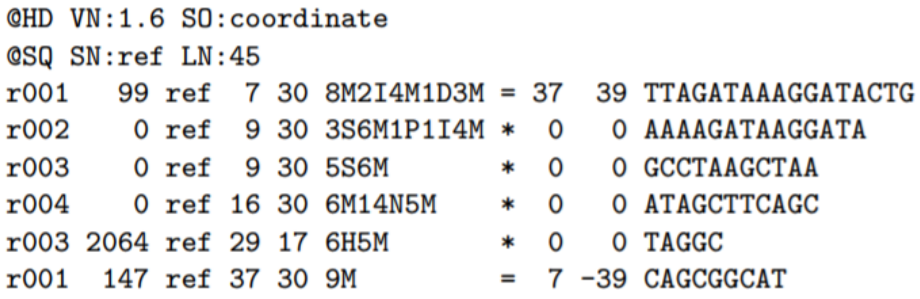
A file format to represent alignments

BAM -> binary form of SAM



```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                    ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                       CAGCGGCAT
```



```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT
```

https://samtools.github.io/hts-specs/SAMv1.pdf

How alignment is made in practice

BWA (Burrows-Wheeler Aligner) is the most popular tool for WGS/WES

- Align millions of short reads to a human-sized genome in minutes

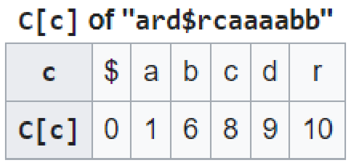
It is based on the FM-index of the Burrows-Wheeler Transform

"abracadabra$"

(genome)

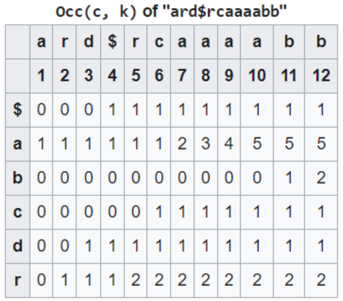
BWT

"ard$rcaaaabb"



C[c] of "ard$rcaaaabb"

c	$	a	b	c	d	r
C[c]	0	1	6	8	9	10



Occ(c, k) of "ard$rcaaaabb"

	a	r	d	$	r	c	a	a	a	a	b	b
	1	2	3	4	5	6	7	8	9	10	11	12
$	0	0	0	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	2	3	4	5	5	5
b	0	0	0	0	0	0	0	0	0	0	1	2
c	0	0	0	0	0	1	1	1	1	1	1	1
d	0	0	1	1	1	1	1	1	1	1	1	1
r	0	1	1	1	2	2	2	2	2	2	2	2

FM - index

https://en.wikipedia.org/wiki/FM-index

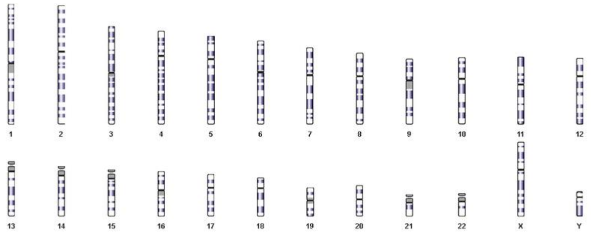
How alignment is made in practice

Genome BWT FM-índex needs to be created (only once)

```
$ bwa index genome.fasta
```

(genome.fasta file)

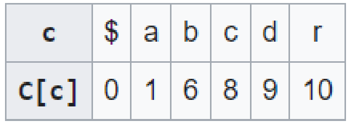
Reference Genome



3×10^9 bases

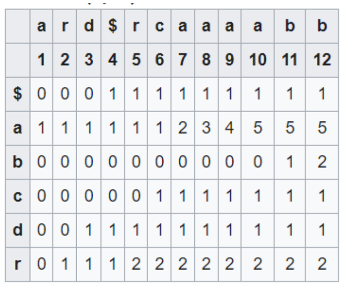
AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

BWT +



c	$	a	b	c	d	r
c[c]	0	1	6	8	9	10

+



	a	r	d	$	r	c	a	a	a	a	b	b
	1	2	3	4	5	6	7	8	9	10	11	12
$	0	0	0	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	2	3	4	5	5	5
b	0	0	0	0	0	0	0	0	0	0	1	2
c	0	0	0	0	0	1	1	1	1	1	1	1
d	0	0	1	1	1	1	1	1	1	1	1	1
r	0	1	1	1	2	2	2	2	2	2	2	2

FM - index

How alignment is made in practice

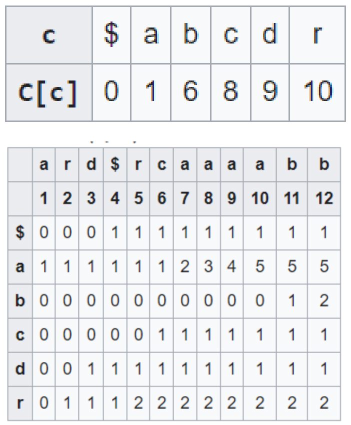
Basic command to generate alignments with BWA:

```
$ bwa mem genome.fasta reads_R1.fastq(.gz) reads_R2.fastq(.gz)>output.sam
```

reads_R1.fastq
TGCTCAGTT

reads_R2.fastq
ACGTCCGA

BWT



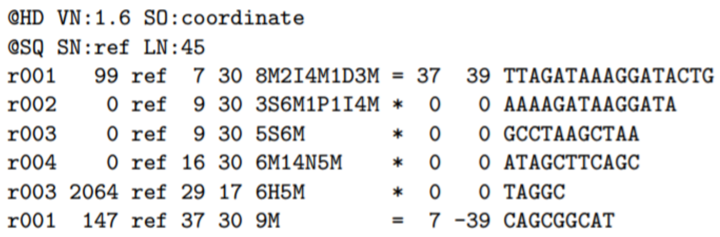
c	$	a	b	c	d	r
C[c]	0	1	6	8	9	10

	a	r	d	$	r	c	a	a	a	a	b	b
	1	2	3	4	5	6	7	8	9	10	11	12
$	0	0	0	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	2	3	4	5	5	5
b	0	0	0	0	0	0	0	0	0	0	1	2
c	0	0	0	0	0	1	1	1	1	1	1	1
d	0	0	1	1	1	1	1	1	1	1	1	1
r	0	1	1	1	2	2	2	2	2	2	2	2

Output in SAM format

TGCTCAGTT Chr2 position xxx

ACGTCCGA Chr2 position yyy



```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT
```

How alignment is made in practice

Paired R1

TGCTCAGTT

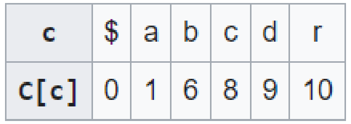
TGC TCA GTT

Paired R2

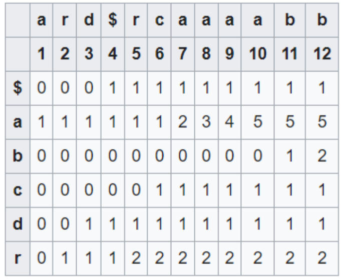
ACGTCCGA

ACG TCC

BWT



c	$	a	b	c	d	r
C[c]	0	1	6	8	9	10



	a	r	d	$	r	c	a	a	a	a	b	b
	1	2	3	4	5	6	7	8	9	10	11	12
$	0	0	0	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	2	3	4	5	5	5
b	0	0	0	0	0	0	0	0	0	0	1	2
c	0	0	0	0	0	1	1	1	1	1	1	1
d	0	0	1	1	1	1	1	1	1	1	1	1
r	0	1	1	1	2	2	2	2	2	2	2	2

TCC chr1
chr2
chr3

TCA chr2
chr3

GTT chr2
chr3
chr6

chr2
chr3
(multimapping)

ACG chr2
chr6

TCC chr1
chr2

chr2

Most likely:

chr2

How alignment is made in practice: Summary

- Special algorithms are used to have fast alignments
  - They are not guaranteed to be perfect but most of the time they are very good

- A read can map equally well to multiple regions (multimappings)
  - BWA reports **one primary alignment (randomly chosen) with mapping quality of 0**
  - Depending on the software, it can generate **secondary alignments**
  - Information of paired reads are used to disambiguate multimappings if possible

- Alignments are made piece-wise (a read is split in segments)
  - A read alignment can be split in a primary and **supplementary alignment(s)**
    - Eg. splicing in RNA-Seq; large deletions
  - Sometimes, only a part of the read is aligned (the rest is “masked”/hidden)
    - Particularly in repetitive areas this can lead to false alignments

Practical Exercise

You will only need to run alignment using ‘bwa mem’, like this:

```
$ bwa mem –t 1 –M -R ‘@RG\tID:sampleid\tPU:FlowCell.Lane.Sample\tSM:samplename\tPL:ILLUMINA\tLB:samplename\tCN:UTI’ genoma.fasta reads_R1.fastq.gz reads_R2.fastq.gz  > output.sam
```

-M: Mark split reads as secondary (relevant for detecting duplicates)

-R: Read Group (to associated specific metadata to each read)

ID: Identifier; PU: Platform Code (FlowCell.Lane.Sample); SM: Sample; PN: LB: Library; CN: Center Name (Sequencing Team) ; other fields possible (see SAM specification)

-t: number of “threads” (CPU processors) to use

Practical Exercise

‘bwa mem’ generates a SAM file. To generate a more compact BAM file that is used to visualize and in variant calling, we need to process the SAM:

- Sort SAM by coordinates using ‘samtools sort’

```
$ samtools sort file.sam –o output_sorted.sam
```

- Convert SAM to BAM using ‘samtools view’

```
$ samtools view –Sb file.sam –o output.bam
```

-S: input file is SAM; -b: output file is bam

- Index the BAM so it can used by other programs using ‘samtools index’

```
$ samtools sort file.bam –o output_sorted.bam
```

(note: this has nothing to do with ‘bwa índex’)

Practical Exercise

- There are three samples, use data from any/all:
  - – Align selected reads against a specific chromosome
    - What is the identifier of the first aligned read? Where does it align?
  - – Sort the aligned reads
    - What is the identifier of the first aligned read? Where does it align?
  - – Convert to a bam and index the bam

Commands to run are in the github

Interlude

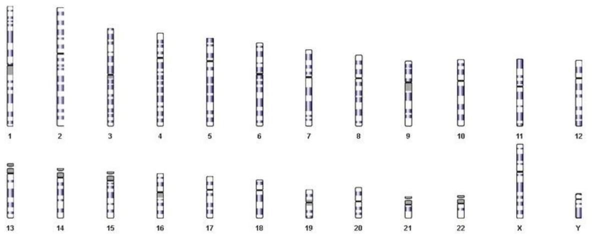
Data pre-processing for variant discovery

Raw Unmapped Reads

uBAM or FASTQ

(fasta file)

Reference Genome



TCCATGC AGTTGTGT

ACTCCAT GTTGT

AAGCGATG TGCTCAGTT

GTGTGTTT-CA

Alignments

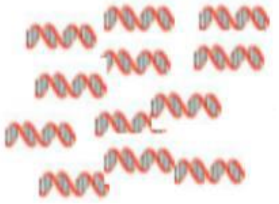
(SAM/BAM)

geneA **geneB** **geneC**

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

DNA Extraction

Sequencing



TGCTCAGTT TGAC ATGGAGT GTTGT

(ACTCCAT)

(“clean” fastQ)

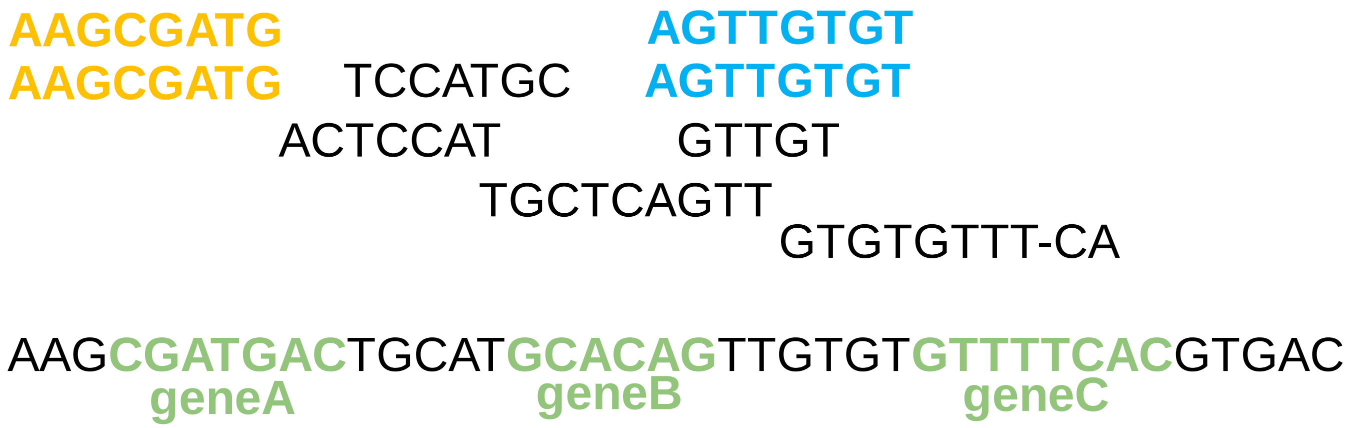
QC and pre-processing

TGCTCAGTTA TGAC ATGGAGT TTT CGTTGT

(“raw” fastQ)

Data pre-processing for variant discovery
Duplicated Reads

- Duplicate reads (same fragment) can appear
  - In library preparation during amplification (eg. WES)
  - In the amplification process while sequencing (optical duplicates)
  - They can generate "false" mutations, or at a false frequency

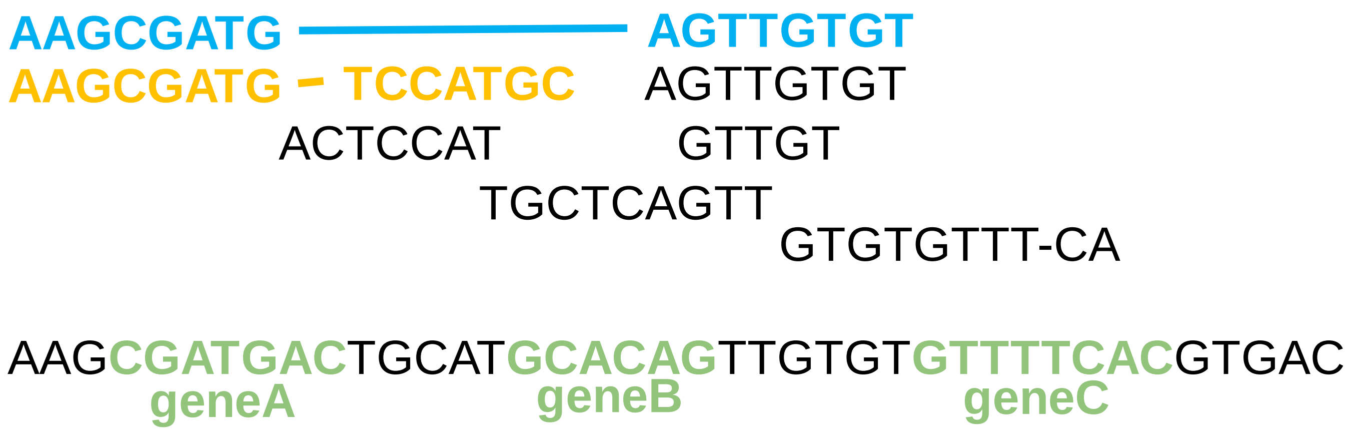


```
AAGCGATG                         AGTTGTGT
AAGCGATG    TCCATGC     AGTTGTGT
                  ACTCCAT             GTTGT
                            TGCTCAGTT
                                         GTGTGTTT-CA

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC
       geneA                  geneB                geneC
```

Data pre-processing for variant discovery
Duplicated Reads

Read pairs need to be taken in consideration



AAGCGATG AGTTGTGT

AAGCGATG - TCCATGC AGTTGTGT

ACTCCAT GTTGT

TGCTCAGTT

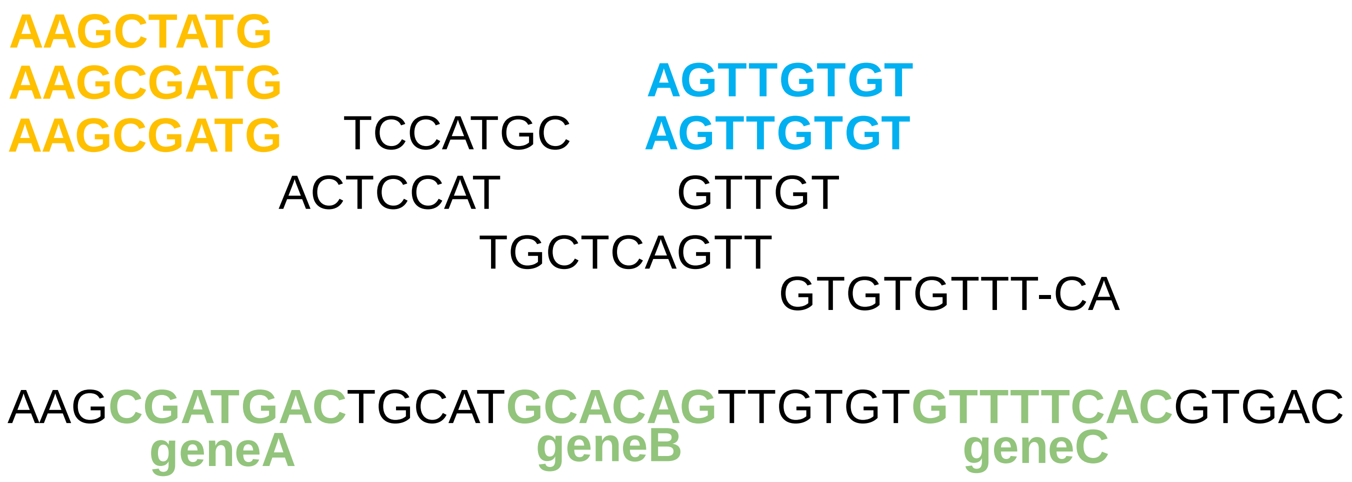
GTGTGTTT-CA

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC

geneA geneB geneC

Data pre-processing for variant discovery
Duplicated Reads

Duplicates are identified by position (alignment and/or flow-cell), not by sequence



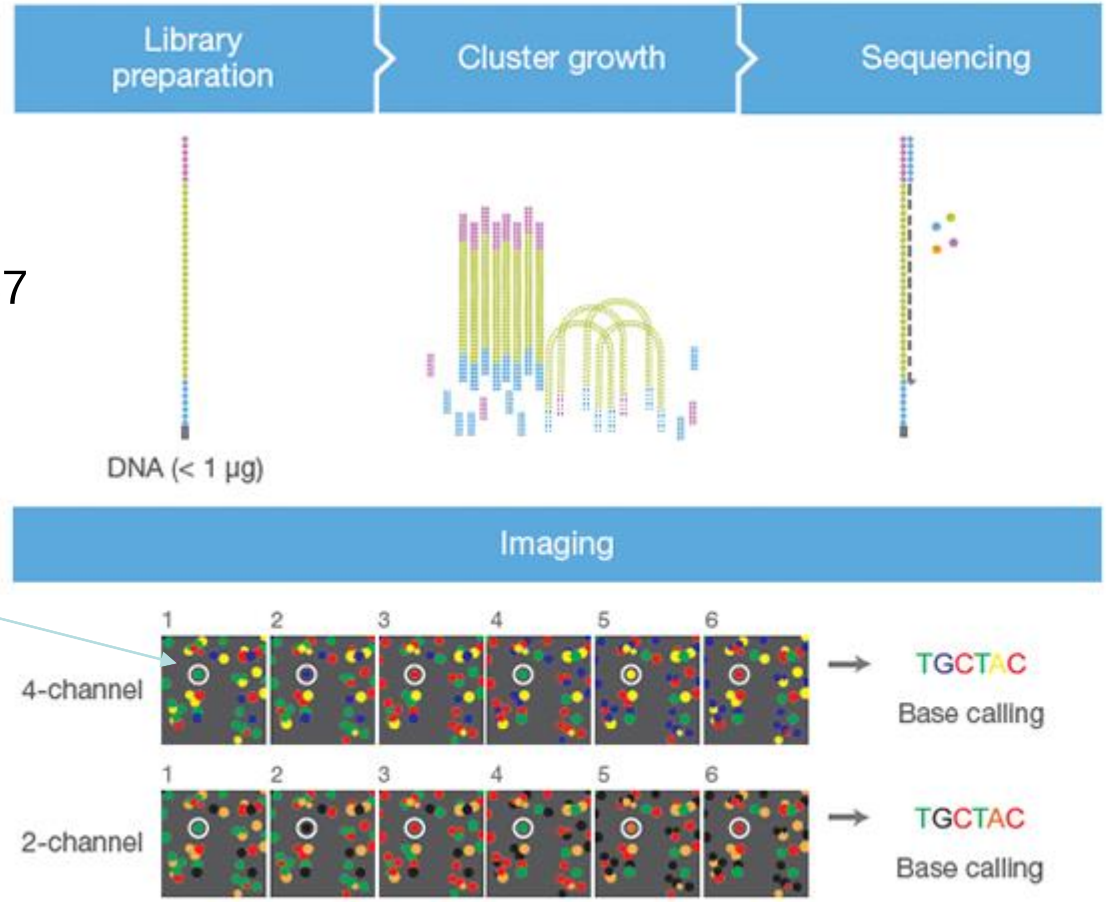
```
AAGCTATG
AAGCGATG                              AGTTGTGT
AAGCGATG  TCCATGC                     AGTTGTGT
        ACTCCAT                         GTTGT
                  TGCTCAGTT
                                         GTGTGTTT-CA

AAGCGATGACTGCATGCACAGTTGTGTGTTTTCACGTGAC
   geneA        geneB          geneC
```

Data pre-processing for variant discovery
Duplicated Reads: optical duplicates

@MN00723:33:000H3MCVT:1:11102:7591:1087 1:N:0:47

Machine Flow Cell Lane Position



Optical duplicates are duplicate reads that are very close in the flow cell

Data pre-processing for variant discovery
Duplicated Reads

- **The recommended practice is to ignore duplicates**
  - Only consider one of the duplicates for variant calling
    - Usually the one with the best quality
  - This may remove good information (eg. with high coverage, targeted)
  - Duplicates are marked and later ignored (or not)
  - Benefits of marking duplicates not always obvious
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4965708/
    - Eg. one can chose to only ignore reads marker as optical duplicates

Data pre-processing for variant discovery
Duplicated Reads

We will run this process using ‘gatk MarkDuplicates’

It takes a BAM file and annotates duplicated reads:

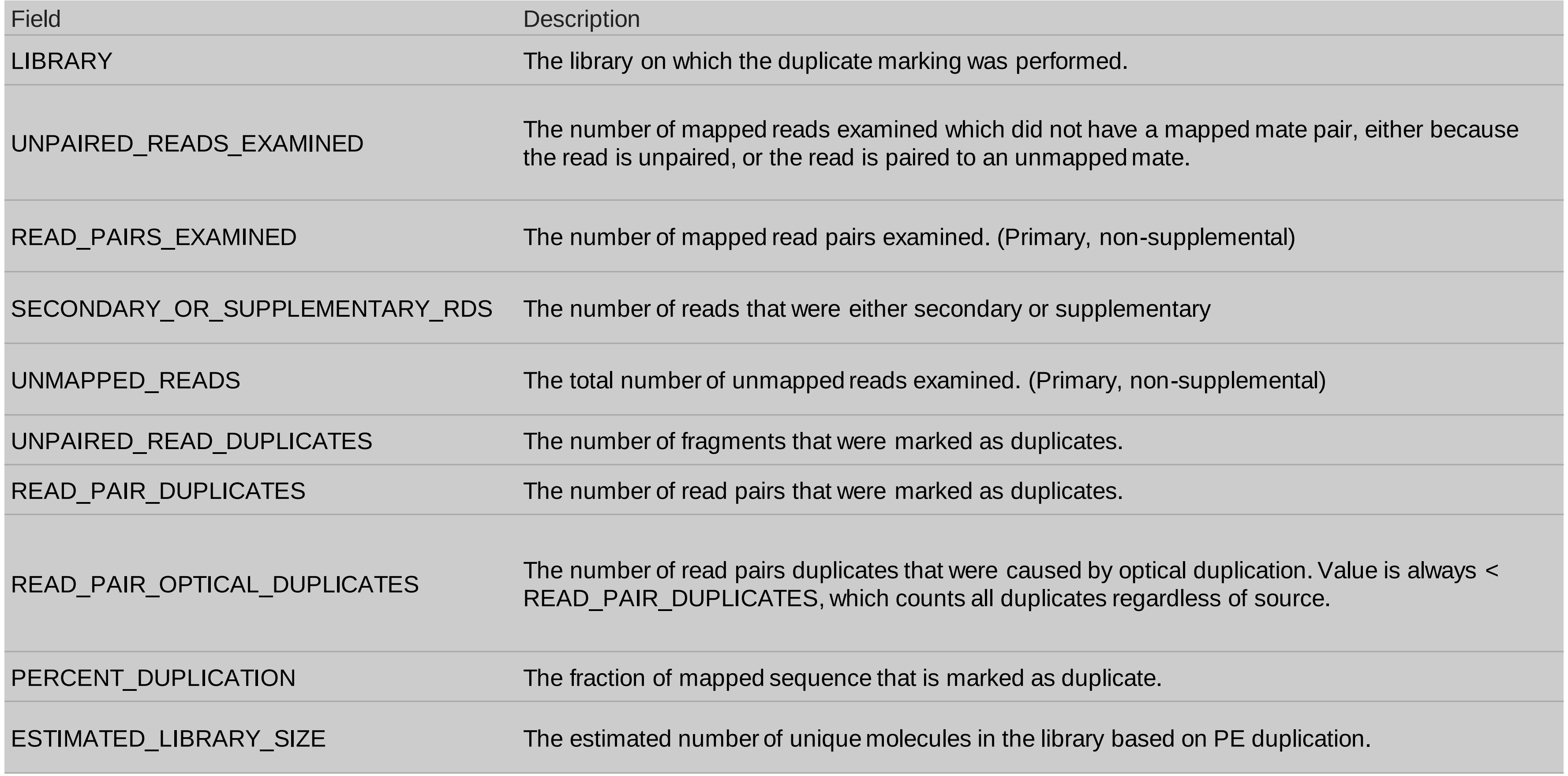
```
> gatk MarkDuplicates -I=sample.bam -O=sample_marked.bam -M=sample_marked_metrics.txt
```

It generates another BAM file with the same number of alignments, but alignments determined as duplicated have an annotation that can be used.

It generates a file with metrics from the analysis (described in next slide)

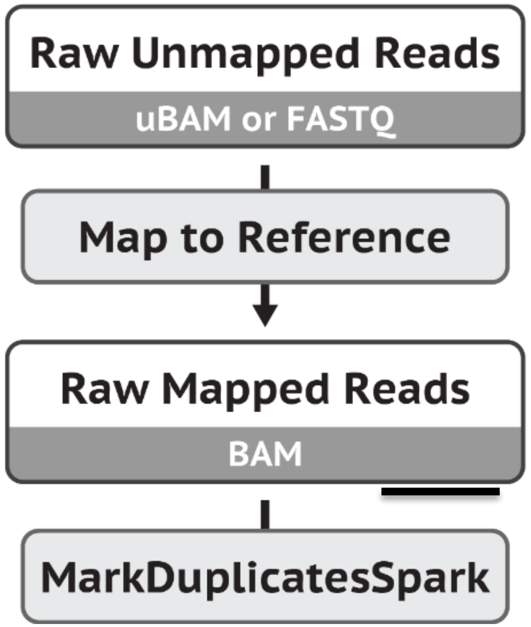
Data pre-processing for variant discovery
Duplicated Reads

Metrics



Field	Description
LIBRARY	The library on which the duplicate marking was performed.
UNPAIRED_READS_EXAMINED	The number of mapped reads examined which did not have a mapped mate pair, either because the read is unpaired, or the read is paired to an unmapped mate.
READ_PAIRS_EXAMINED	The number of mapped read pairs examined. (Primary, non-supplemental)
SECONDARY_OR_SUPPLEMENTARY_RDS	The number of reads that were either secondary or supplementary
UNMAPPED_READS	The total number of unmapped reads examined. (Primary, non-supplemental)
UNPAIRED_READ_DUPLICATES	The number of fragments that were marked as duplicates.
READ_PAIR_DUPLICATES	The number of read pairs that were marked as duplicates.
READ_PAIR_OPTICAL_DUPLICATES	The number of read pairs duplicates that were caused by optical duplication. Value is always < READ_PAIR_DUPLICATES, which counts all duplicates regardless of source.
PERCENT_DUPLICATION	The fraction of mapped sequence that is marked as duplicate.
ESTIMATED_LIBRARY_SIZE	The estimated number of unique molecules in the library based on PE duplication.

Data pre-processing for variant discovery



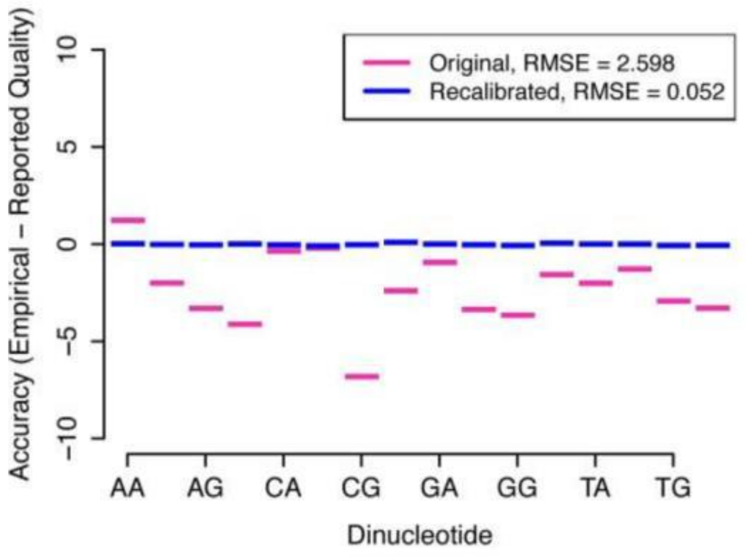
Now we have a BAM with potential duplicated alignments marked

Data pre-processing for variant discovery
Base Quality Recalibration

Base Quality not very precise

Depends on several factors:

- Sample Quality (DNA)
- Nucleotide context
- Machine and cycle of sequencing
- Type of variant (SNP or Indel)

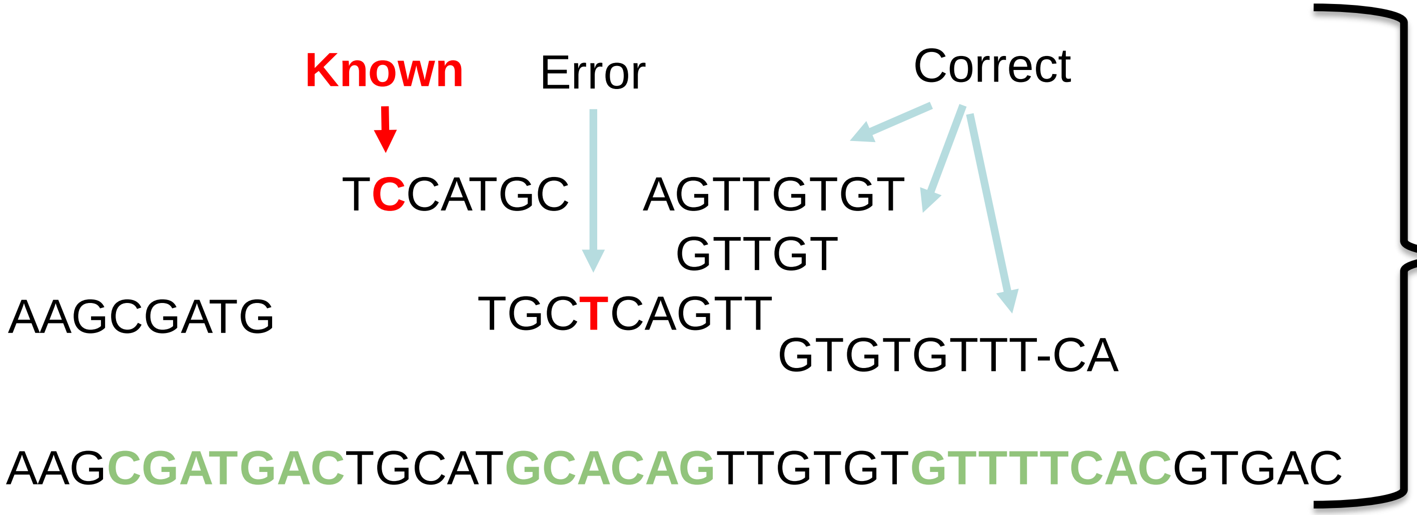


The machine oftens does not estimate correctly the base quality score

Data pre-processing for variant discovery
Base Quality Recalibration

Use list of known variants to estimate correct quality values

- All bases different from reference **not in the provided list** of known variants are considered to be errors



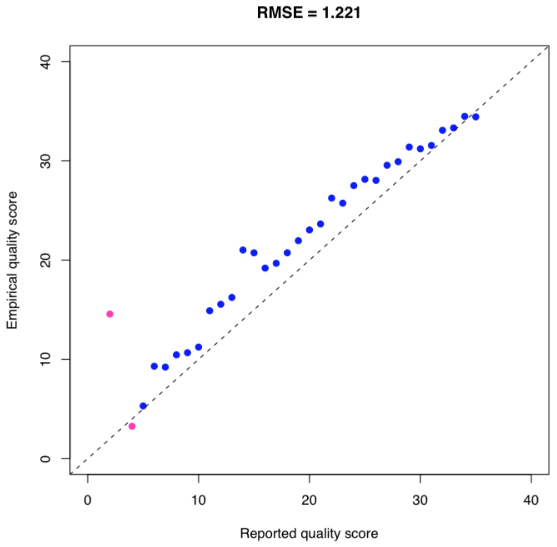
Count occurrences and compare Q value marked by the machine with the observed % errors

Data pre-processing for variant discovery
Base Quality Recalibration

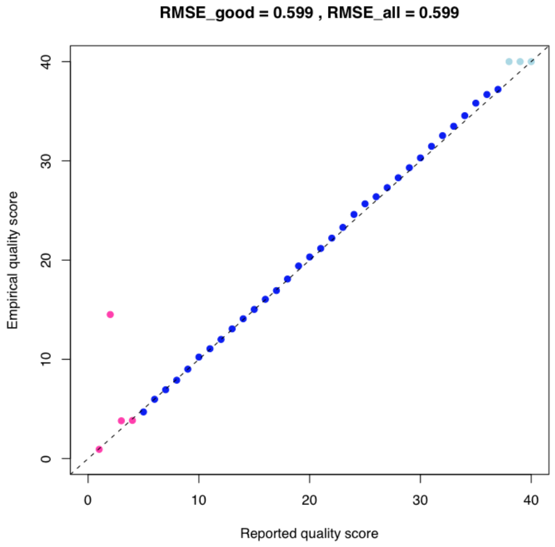
Base Quality Recalibration:

The covariates being used here:

- ReadGroupCovariate
- QualityScoreCovariate
- ContextCovariate
- CycleCovariate



Original Data



After GATK Recalibration

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3083463/

https://www.youtube.com/watch?v=L4D1dwES9s8

https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR-

Data pre-processing for variant discovery
Base Quality Recalibration

We will run this process using ‘gatk BaseRecalibrator / ApplyBQSR’

First, we need to generate a table with estimated base qualities split by covariate

```
> gatk BaseRecalibrator -I sample_marked.bam -R genome.fasta --known-sites known_snps.vcf
--known-sites known_indels.vcf [ --intervals target_positions.bed --interval-padding 100 ] -O
sample_marked_baserecalibrator_report.txt
```

In the case of WES we need to pass the relevant regions (--intervals and --interval-padding)

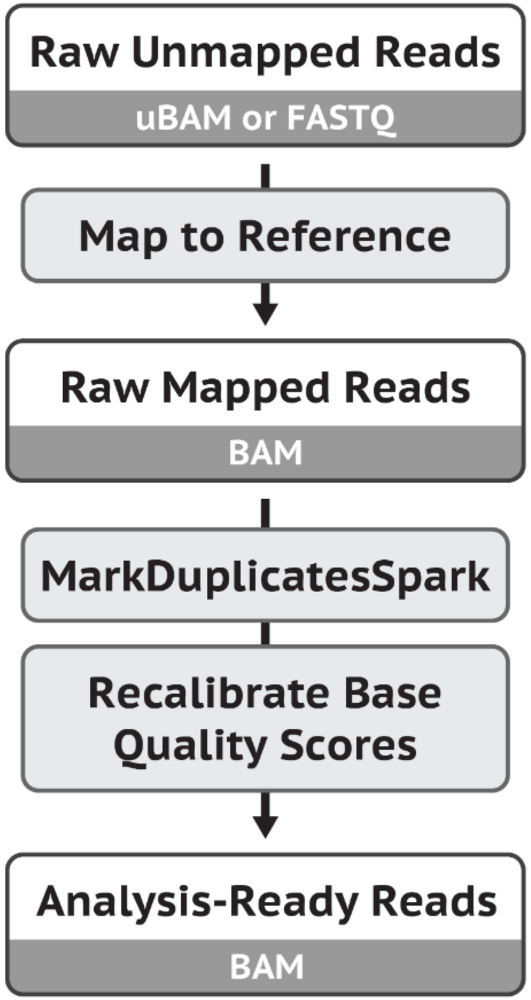
Next, we need to apply the inferred re-estimated qualities to the BAM file:

```
>gatk ApplyBQSR -I sample_marked.bam -R genome.fasta --bqsr-recal-file
sample_marked_baserecalibrator_report.txt –O sample_marked_baserecalibrator.bam
```

This will generate a final BAM, with marked duplicates and calibrated base qualities

Data pre-processing for variant discovery

GATK
Best
Practices



These steps can be used with other variant callers

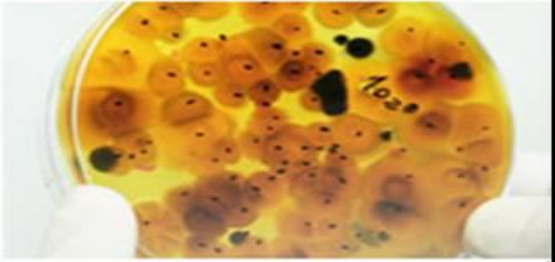
https://gatk.broadinstitute.org/hc/en-us/articles/360035535912-Data-pre-processing-for-variant-discovery

Practical Exercise

- There are three samples, use data from any/all:
  - – Mark duplicates using “gatk MarkDuplicates”
    - Describe the contents of the Metrics File
  - – Apply Base Recalibration: gatk BaseRecalibrator / ApplyBQSR

Commands to run are in the github

Interlude



NGS: Quality Assessment and Visualization of Read Mappings

Quality Assessment of Read Mappings

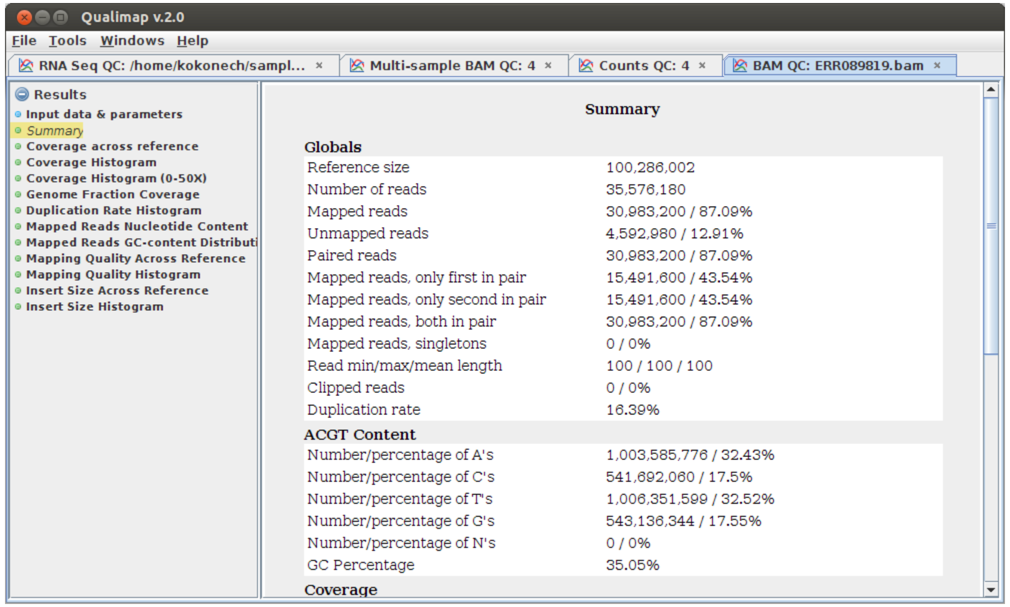
Measures to consider:

- Mapped Reads / Unmapped Reads
  - Usually >80% for WGS; >90% for WES
- Total Mapped Reads / Coverage
  - Presence of Duplicates (depends on coverage):
    - Note that this is different from the duplicates inferred by FastQC
- Homogeneity in the coverage
- Distribution of Fragment Size (paired-end)
- Coverage of Targeted Regions (in case of eg. WES)

Quality Assessment of Read Mappings

- Qualimap: similar to FastQC, but for BAM files

http://qualimap.conesalab.org/


Qualimap v.2.0
File Tools Windows Help
RNA Seq QC: /home/kokonech/sampl... ×
Multi-sample BAM QC: 4 ×
Counts QC: 4 ×
BAM QC: ERR089819.bam ×
Results
Input data & parameters
Summary
Coverage across reference
Coverage Histogram
Coverage Histogram (0-50X)
Genome Fraction Coverage
Duplication Rate Histogram
Mapped Reads Nucleotide Content
Mapped Reads GC-content Distributi
Mapping Quality Across Reference
Mapping Quality Histogram
Insert Size Across Reference
Insert Size Histogram

Summary

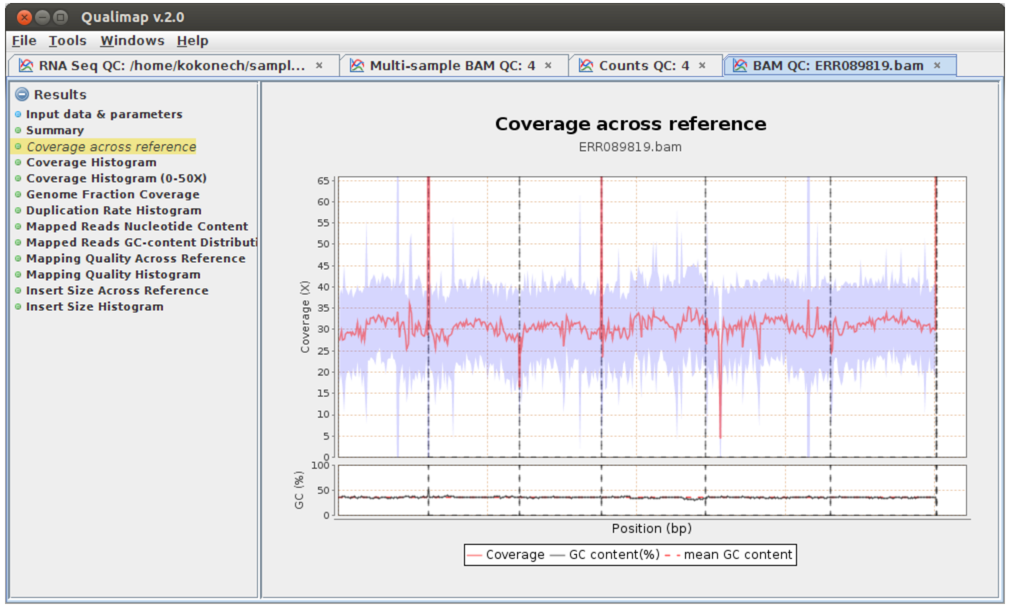
Globals

Reference size	100,286,002
Number of reads	35,576,180
Mapped reads	30,983,200 / 87.09%
Unmapped reads	4,592,980 / 12.91%
Paired reads	30,983,200 / 87.09%
Mapped reads, only first in pair	15,491,600 / 43.54%
Mapped reads, only second in pair	15,491,600 / 43.54%
Mapped reads, both in pair	30,983,200 / 87.09%
Mapped reads, singletons	0 / 0%
Read min/max/mean length	100 / 100 / 100
Clipped reads	0 / 0%
Duplication rate	16.39%

ACGT Content

Number/percentage of A's	1,003,585,776 / 32.43%
Number/percentage of C's	541,692,060 / 17.5%
Number/percentage of T's	1,006,351,599 / 32.52%
Number/percentage of G's	543,136,344 / 17.55%
Number/percentage of N's	0 / 0%
GC Percentage	35.05%

Coverage



Practical example

Qualimap is a tool that can easily be run in Windows:

We will use ‘qualimap bamqc’:

```
>qualimap bamqc –bam sample_marked_recalibrated.bam –outdir . –outfile
sample_marked_recalibrated_qualimap.pdf –outformat PDF [ -gff
targets_position.bed ]
```

In the case of WES you need to pass a file with relevant intervals

It will generate a PDF (or HTML) with results, like FASTQC

Practical Exercise

Run qualimap bamqc for any/all of the samples

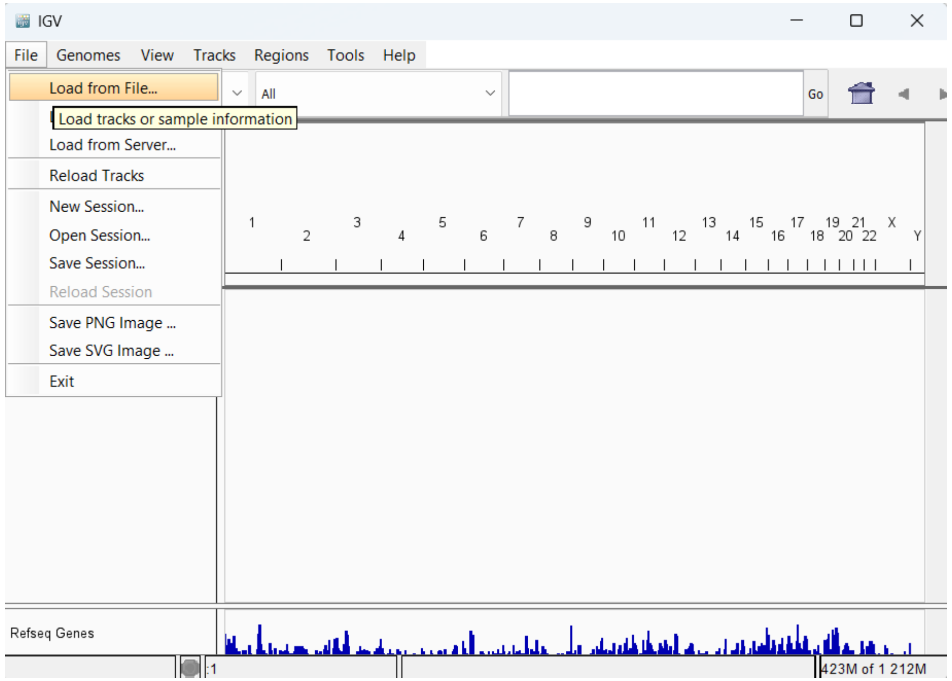
- Compare with the metrics from GATK markduplicates
- What is the distribution of fragment length?
- If you have time, compare reports of BAMs from each of the steps

Commands to run are in the github

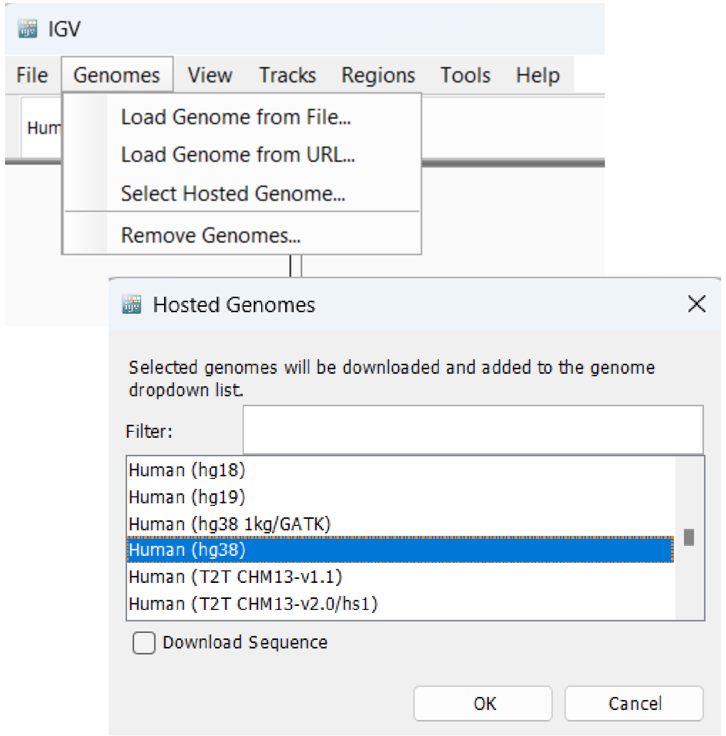
Visualization of Read Mappings with IGV

Load Data in IGV:

- It can be BAM, but also VCF



If genome is not loaded:



Selected genomes will be downloaded and added to the genome dropdown list.

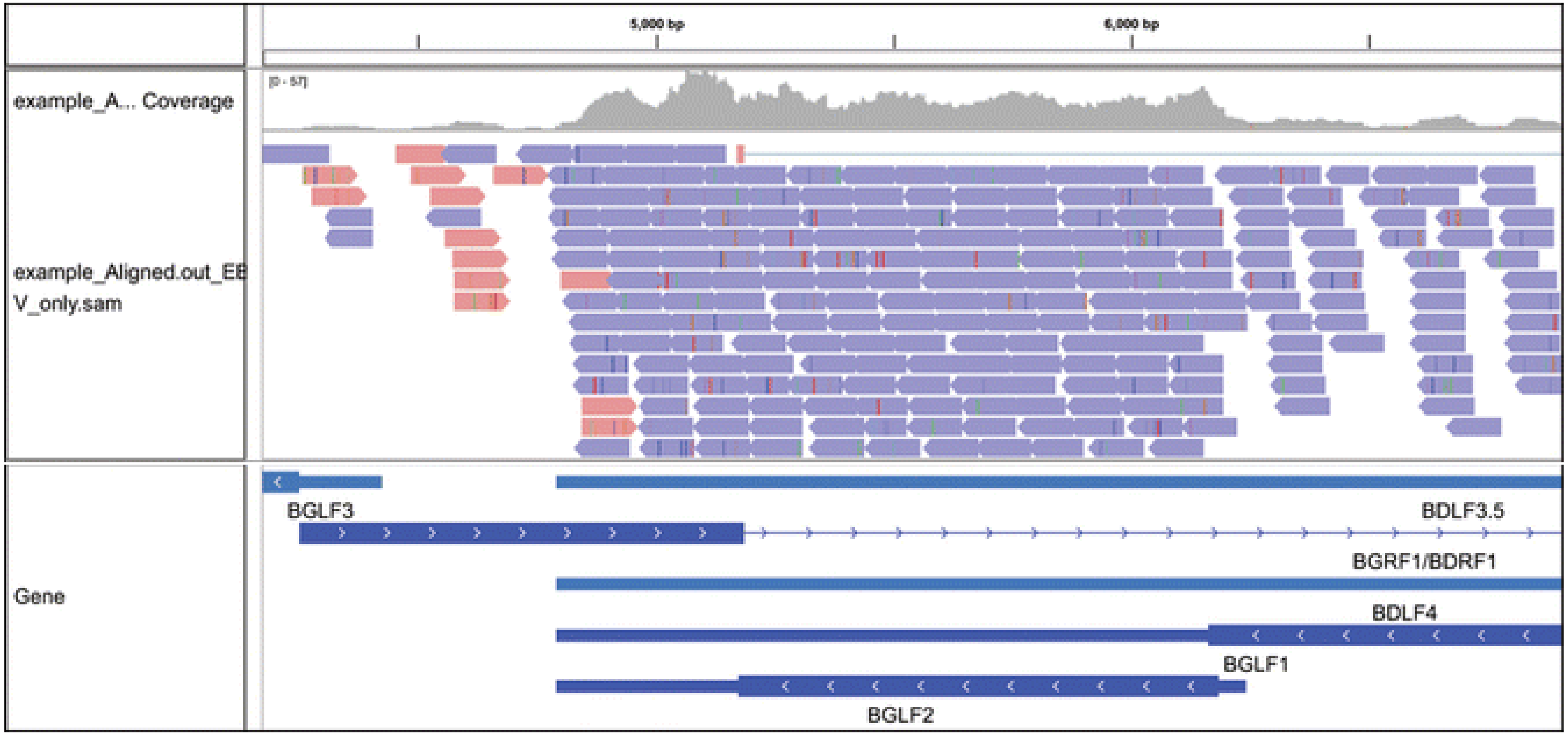
Filter:

Human (hg18)
Human (hg19)
Human (hg38 1kg/GATK)
Human (hg38)
Human (T2T CHM13-v1.1)
Human (T2T CHM13-v2.0/hs1)

□ Download Sequence

Visualization of Read Mappings with IGV

Integrative Genomics Viewer



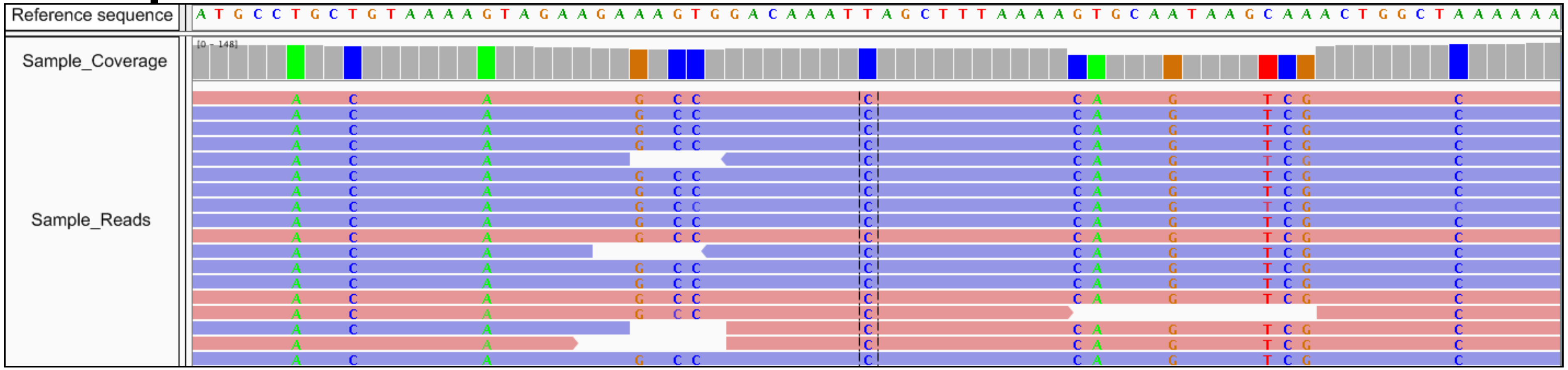
Coverage

Alignments

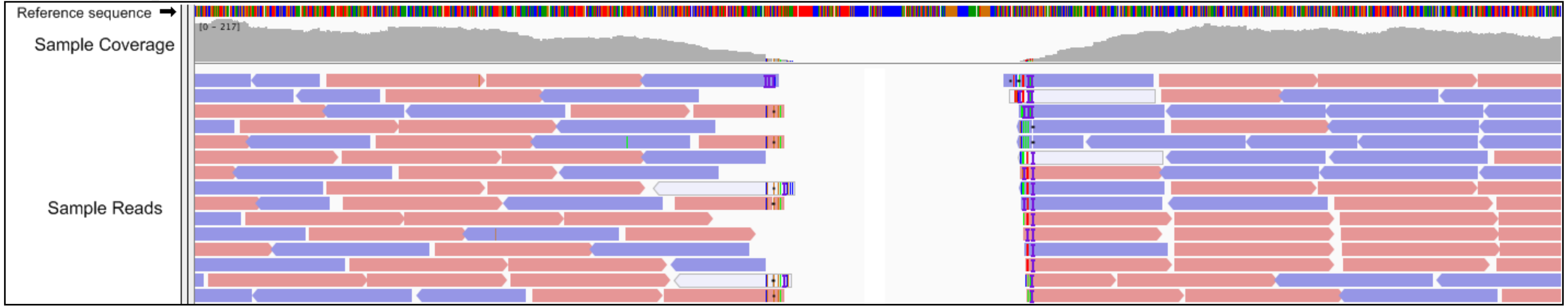
Annotation (genes)

Visualization of Read Mappings with IGV

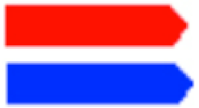
Example mutations


Reference sequence
A T G C C T G C T G T A A A A G T A G A A G A A A G T G G A C A A A T T A G C T T T A A A A G T G C A A T A A G C A A A C T G G C T A A A A A A
Sample_Coverage
[0 - 148]
Sample_Reads

Example deletion

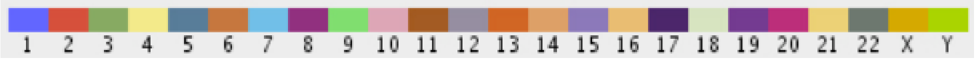


- IGV provides colors to signal unusual situations
  - Besides mutations, information from paired-end is also there



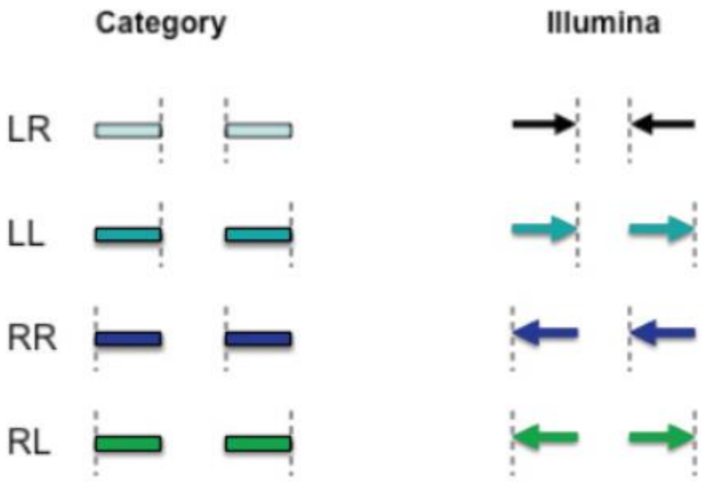
Bigger than expected

Smaller than expected



Pairs on different chromosomes

Insert Size Lengths



Normal

Unusual

Unusual

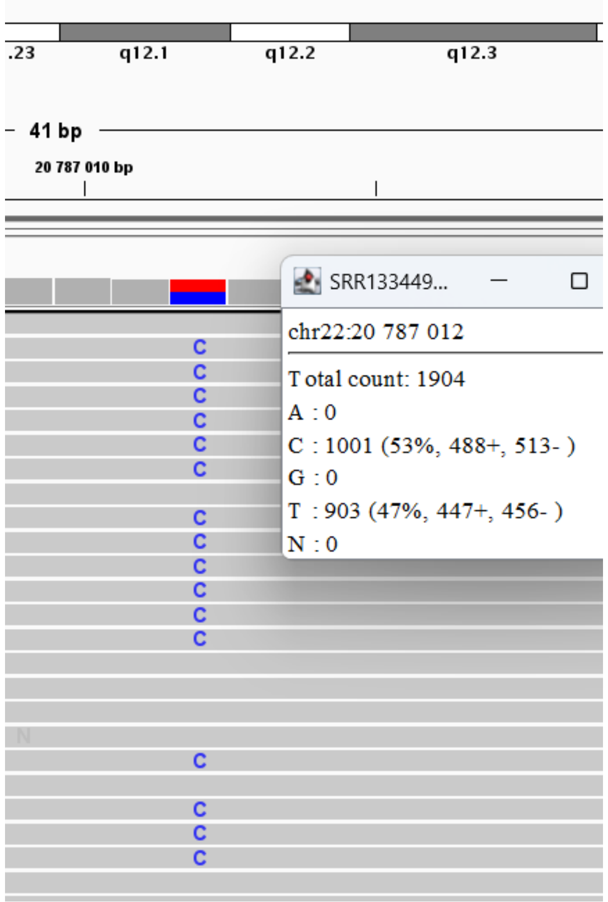
Unusual

Pair orientation

https://software.broadinstitute.org/software/igv/

Visualization of Read Mappings

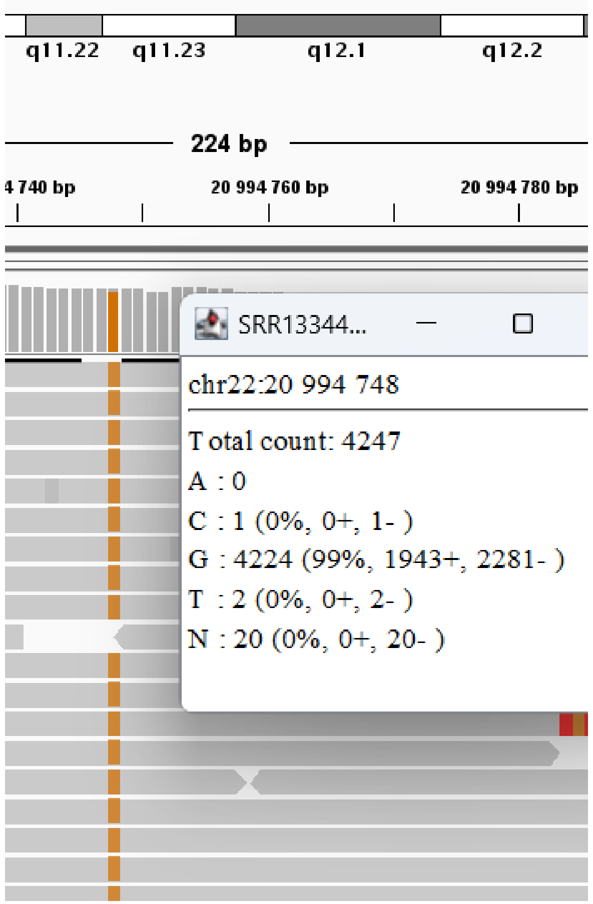
- Example of mutations, showing evidence



SRR133449...

chr22:20 787 012

Total count: 1904
A : 0
C : 1001 (53%, 488+, 513-)
G : 0
T : 903 (47%, 447+, 456-)
N : 0



SRR13344...

chr22:20 994 748

Total count: 4247
A : 0
C : 1 (0%, 0+, 1-)
G : 4224 (99%, 1943+, 2281-)
T : 2 (0%, 0+, 2-)
N : 20 (0%, 0+, 20-)

Practical Exercise

- Look at the alignments in IGV in specific positions:
  - – Sample 1: 2:171868791
  - – Sample 2: 6:121216049
  - – Sample 3: X:129746638

Gather evidence that you think is relevant for the mutation:

- Number of reads, strand of reads, quality etc…