Module 8: Variant Calling and Annotation (Part 1)

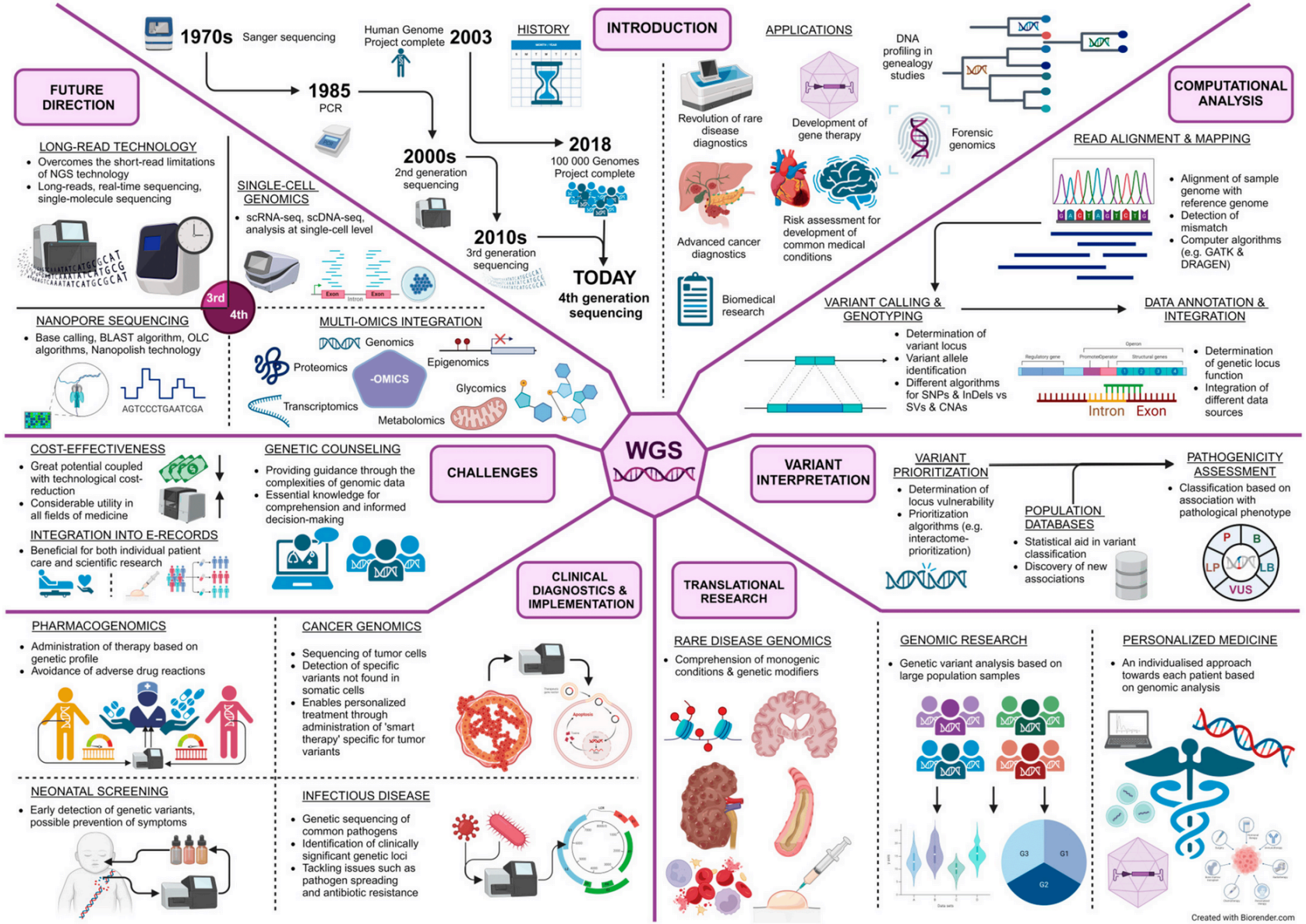
Anouncements

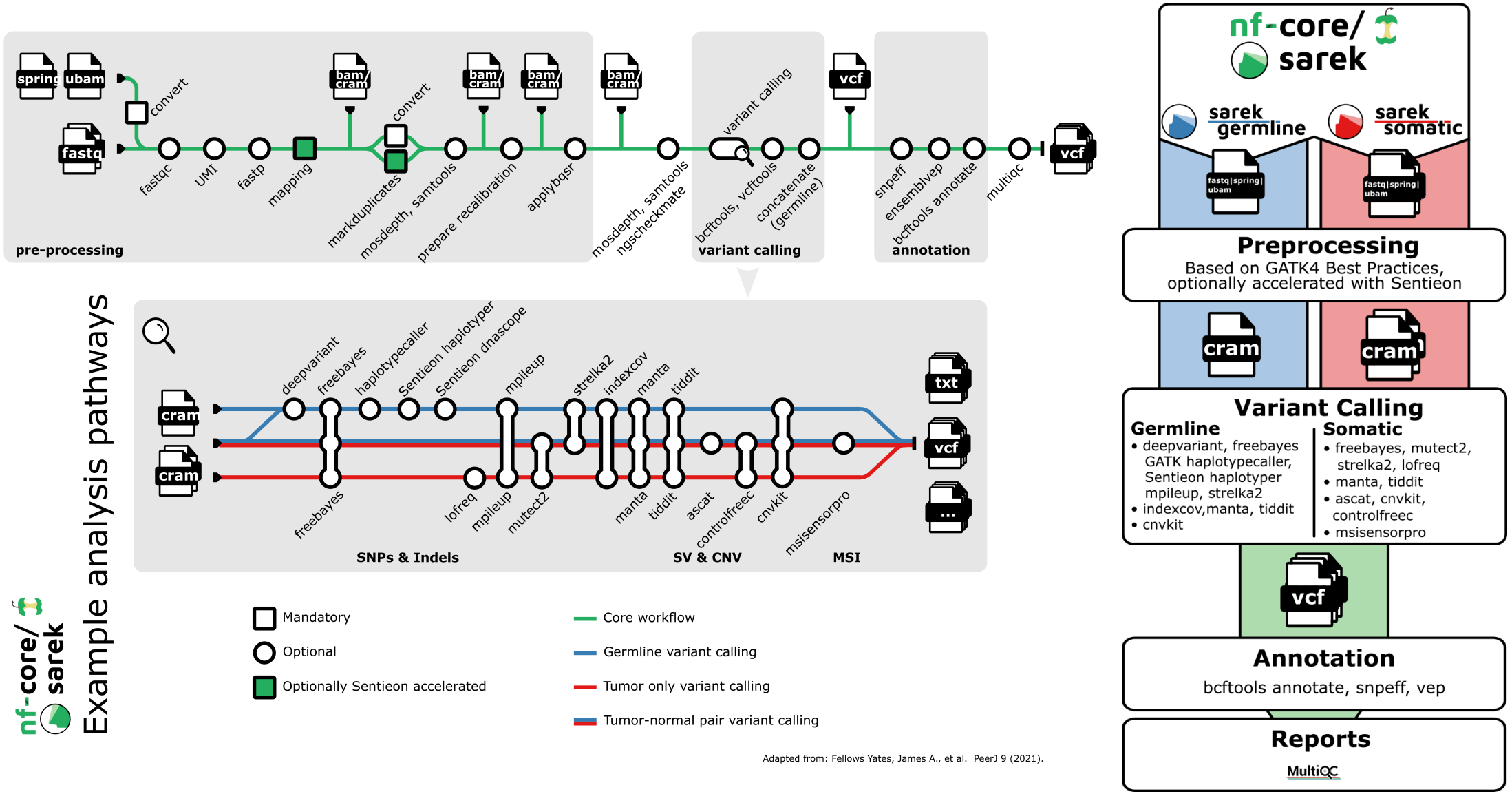
- **Group Assignment #2 (5%)**:due on **June 29**
- **Lab 4**: due on **July 6**
- Lab 2 marks

Key Concepts

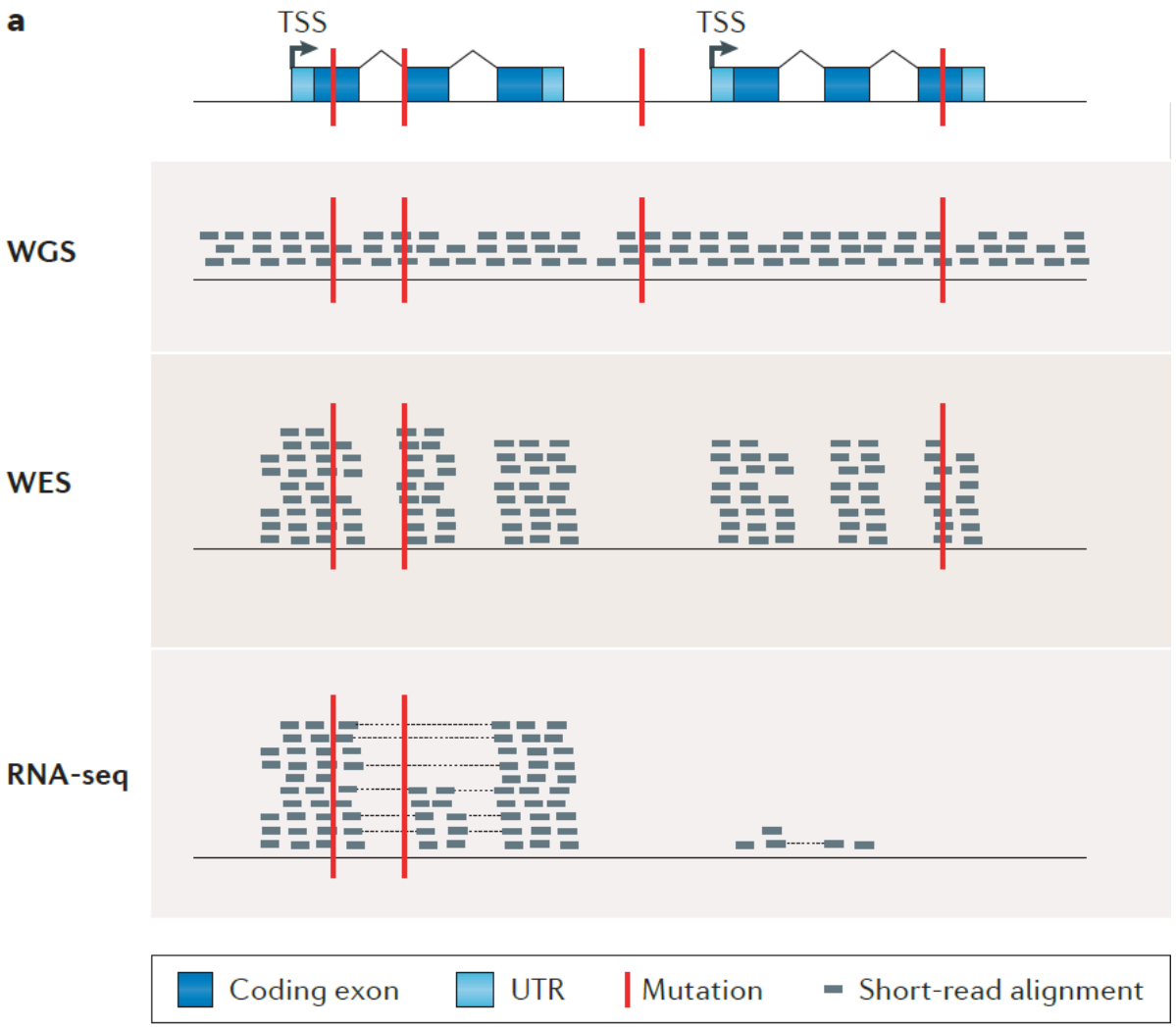
- Quality control
  - Read Quality
  - Alignment Quality
- Pre-processing steps
- Read mapping
- Variant calling

Where does variant calling and annotation fit in the big picture? [1]




nf-core/sarek
Example analysis pathways
pre-processing
variant calling
annotation
convert
fastqc
UMI
fastp
mapping
markduplicates
convert
mosdepth, samtools
prepare recalibration
applybqsr
mosdepth, samtools
ngscheckmate
variant calling
bcftools, vcftools
concatenate (germline)
snpeff
ensemblvep
bcftools annotate
multiqc
deepvariant
freebayes
haplotypecaller
Sentieon haplotyper
Sentieon dnascope
mpileup
strelka2
indexcov
manta
tiddit
freebayes
lofreq
mpileup
mutect2
manta
tiddit
ascat
controlfreec
cnvkit
msisensorpro
SNPs & Indels
SV & CNV
MSI
Mandatory
Optional
Optionally Sentieon accelerated
Core workflow
Germline variant calling
Tumor only variant calling
Tumor-normal pair variant calling
Adapted from: Fellows Yates, James A., et al. PeerJ 9 (2021).
nf-core/sarek
sarek germline
sarek somatic
Preprocessing
Based on GATK4 Best Practices, optionally accelerated with Sentieon
Variant Calling
Germline
deepvariant, freebayes GATK haplotypecaller, Sentieon haplotyper mpileup, strelka2
indexcov,manta, tiddit
cnvkit
Somatic
freebayes, mutect2, strelka2, lofreq
manta, tiddit
ascat, cnvkit, controlfreec
msisensorpro
Annotation
bcftools annotate, snpeff, vep
Reports
MultiQC

Sequence Data: Types of sequencing methods [2]



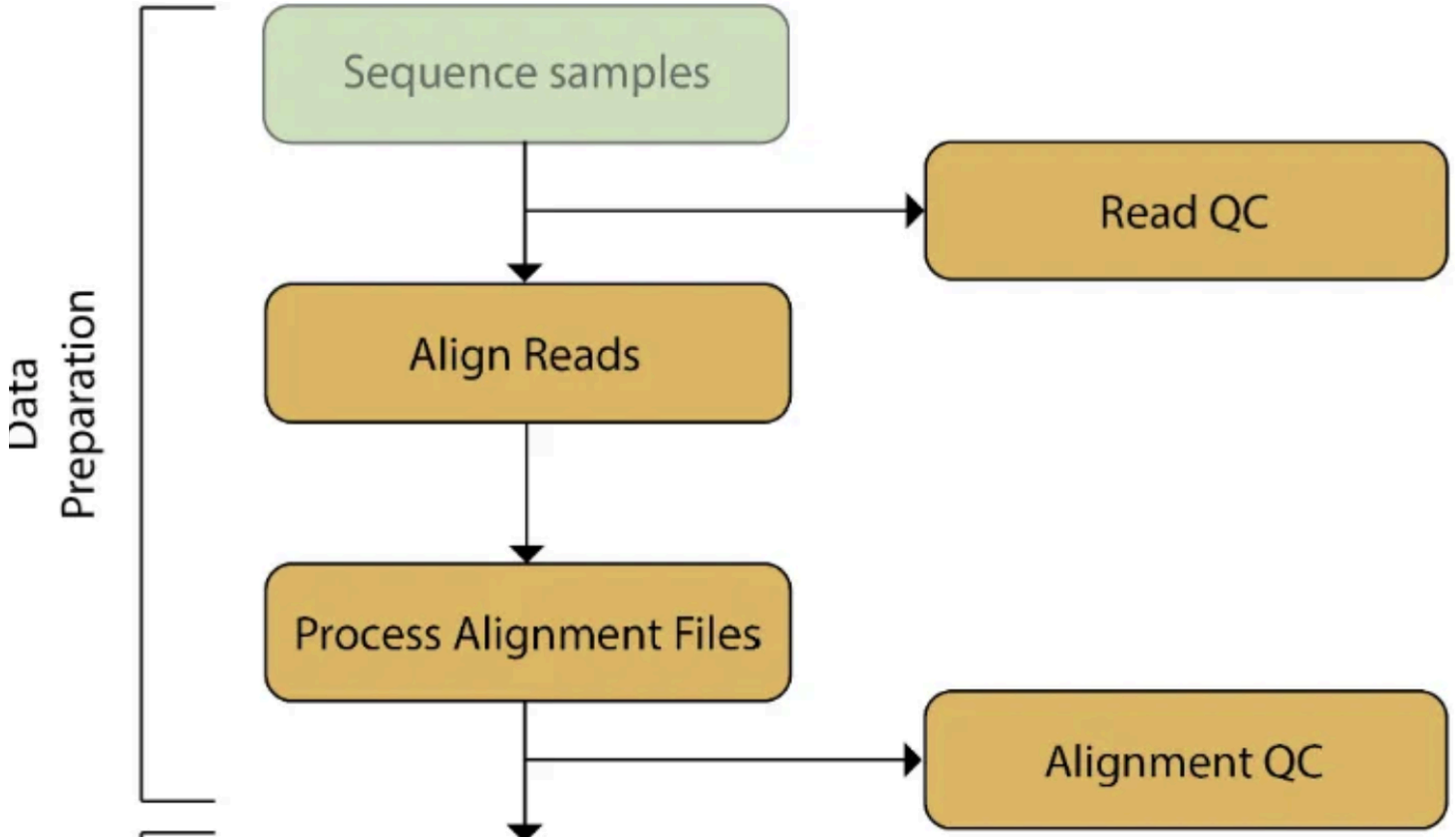
Sequence Data: Types of sequencing methods [3]

Table 4 Comparison of massive sequencing techniques

Technique	Properties	Limitations
Exome sequencing	– Detects genetic variations in all the protein-coding regions of the genome. – Detects nucleotide variations and small insertions and deletions. – Exome size relatively small (1% of the genome size).	– Genetic variations in nonprotein-coding regions, including gene expression regulatory regions, are not detected.
Genome sequencing	– Detects all the genetic variations, including protein-coding and regulatory regions. – Detects nucleotide variations and genome reorganizations such as deletions, duplications, inversions, or translocations.	– The large size of the human genome makes genome sequencing expensive and the analyses of the data generated long and complex.
RNA sequencing	– Detects genetic variations in protein-coding regions. – RNA expression levels can be determined. – Detects RNA splicing variants. – The size of the transcriptome is much smaller than that of the genome.	– The analysis is restricted to the genes expressed in the tissue or cell type analyzed. – Genetic variations in untranscribed regions are not detected.
Selected-DNA sequencing	– Detects genetic variants in a set of predetermined genes. – A relatively small amount of sequencing and data analysis is required. – Can be easily applied to a large number of patients.	– Only the preselected DNA regions are analyzed.

Abbreviations: RNA, ribonucleic acid; DNA, deoxyribonucleic acid.

Quality Control



- Reduces false positives by filtering out low-quality bases, adapters, and contaminants.

Quality Control - Read QC tools

1. **FastQC** : Generates summary metrics (per-base quality, GC content, adapter content, sequence duplication).
    - **Do we need trimming or other preprocessing?**
2. **fastp**: Performs both QC and automated read trimming/correction in a single step.
    - Removes adapters, filters out low-quality reads, corrects mismatches in overlapping paired-end reads.

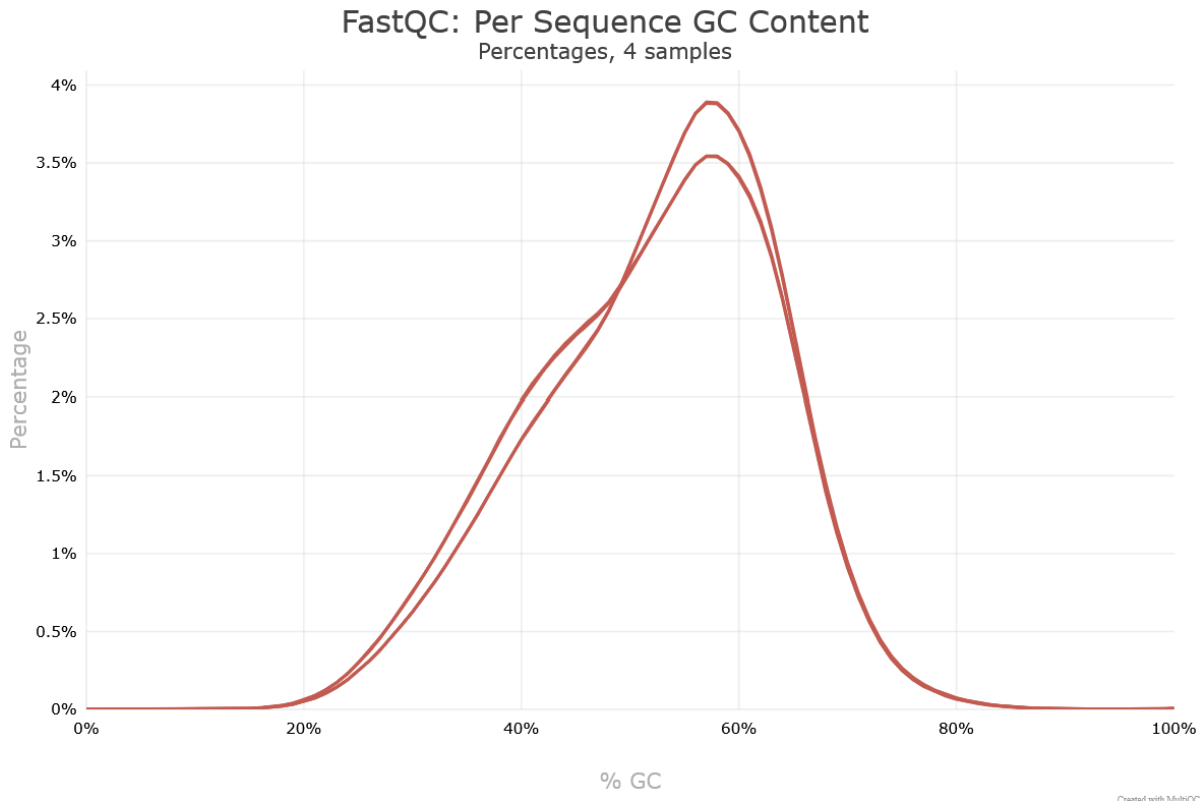
Quality Control - Read QC Evaluation Metrics

Almost identical to Lab 2: RNA Sequencing

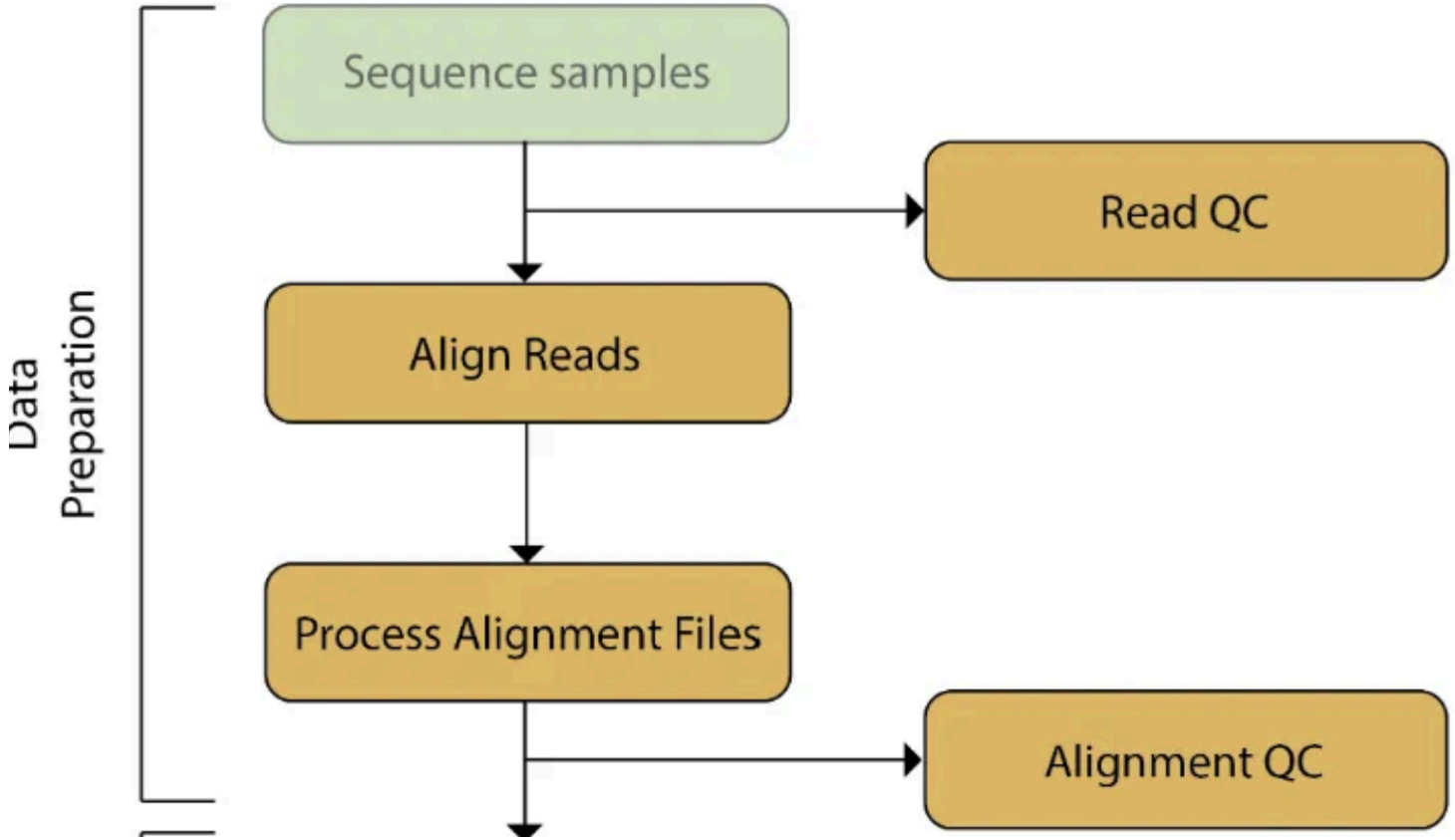
- Sequence Quality (PHRED score)
- GC content
- Insert size distribution
- Read Duplicates
- Overrepresented sequences

Quality Control - Read QC Evaluation Metrics

Exception for WES: The average GC content of the human genome is **~41%**, but GC-content is **higher** in **genic** regions than **intergenic** regions; so **a slightly higher % GC is normal**.



Read Alignment



- After pre-processing (trimming, quality filtering), reads need to be aligned to a reference genome.
- **Alignment tools**: `bwa`, `bwa-mem2`, `dragmap`

Why should we be selective about the tools we use? [4]

- **Time and computing cost**. Average costs per patient on AWS batch for `nf-core/sarek`.

Version	Samples	Avg. costs [$]	Runtime	CPU hours
2.7.2	1 normal	68.04	46h8m	1118.4
3.1.1	1 normal	20.82	12h4m	342.5
3.1.1	1 paired	66.83	31h47m	1324.3

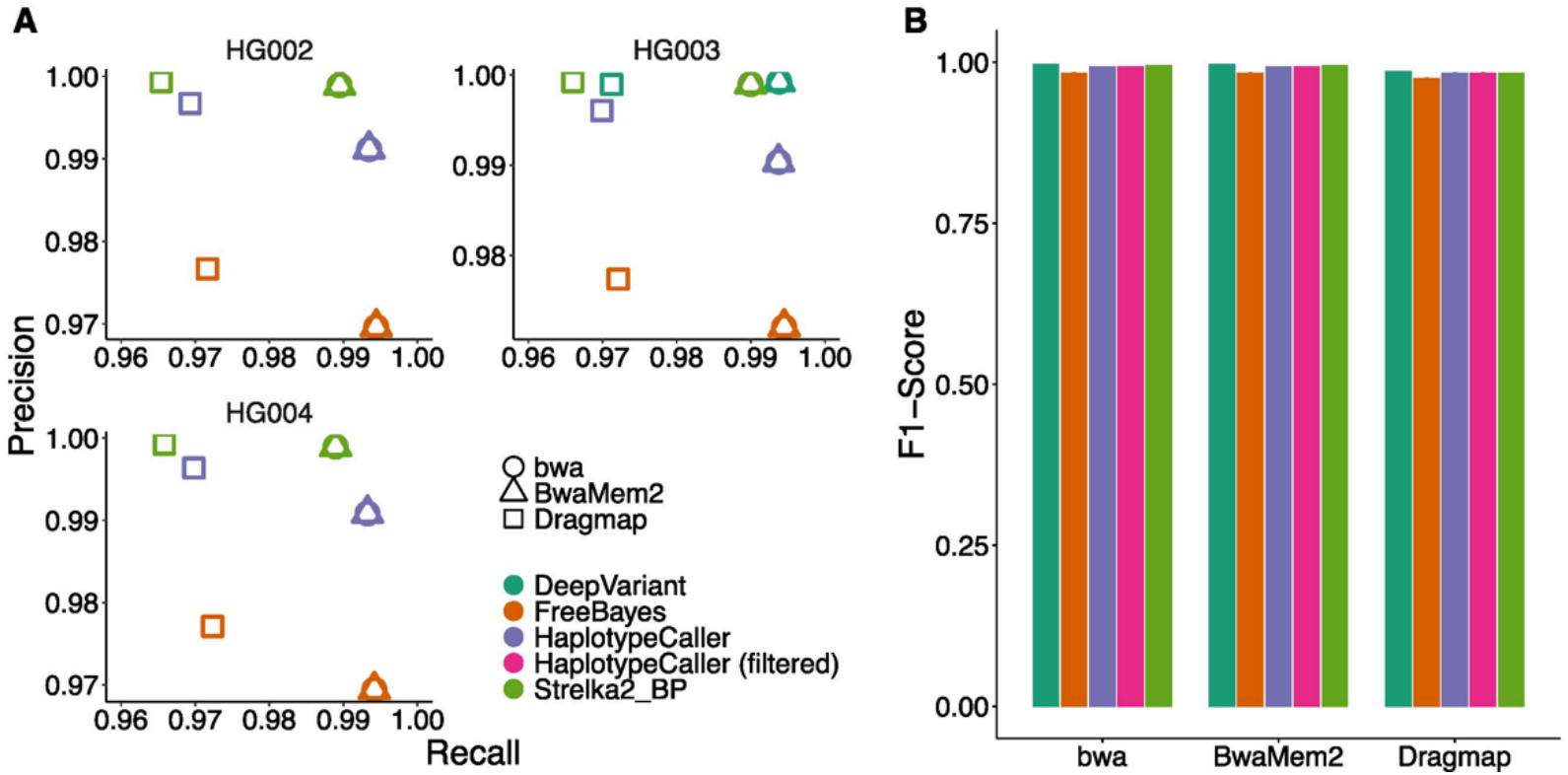
Why should we be selective about the tools we use?

Not all tools are equal. The optimal tool depends on:

1. what information do you what to get out of using it,
2. the underlying assumptions of the algorithms used by the tools, and
3. the nature of the data that you are processing.

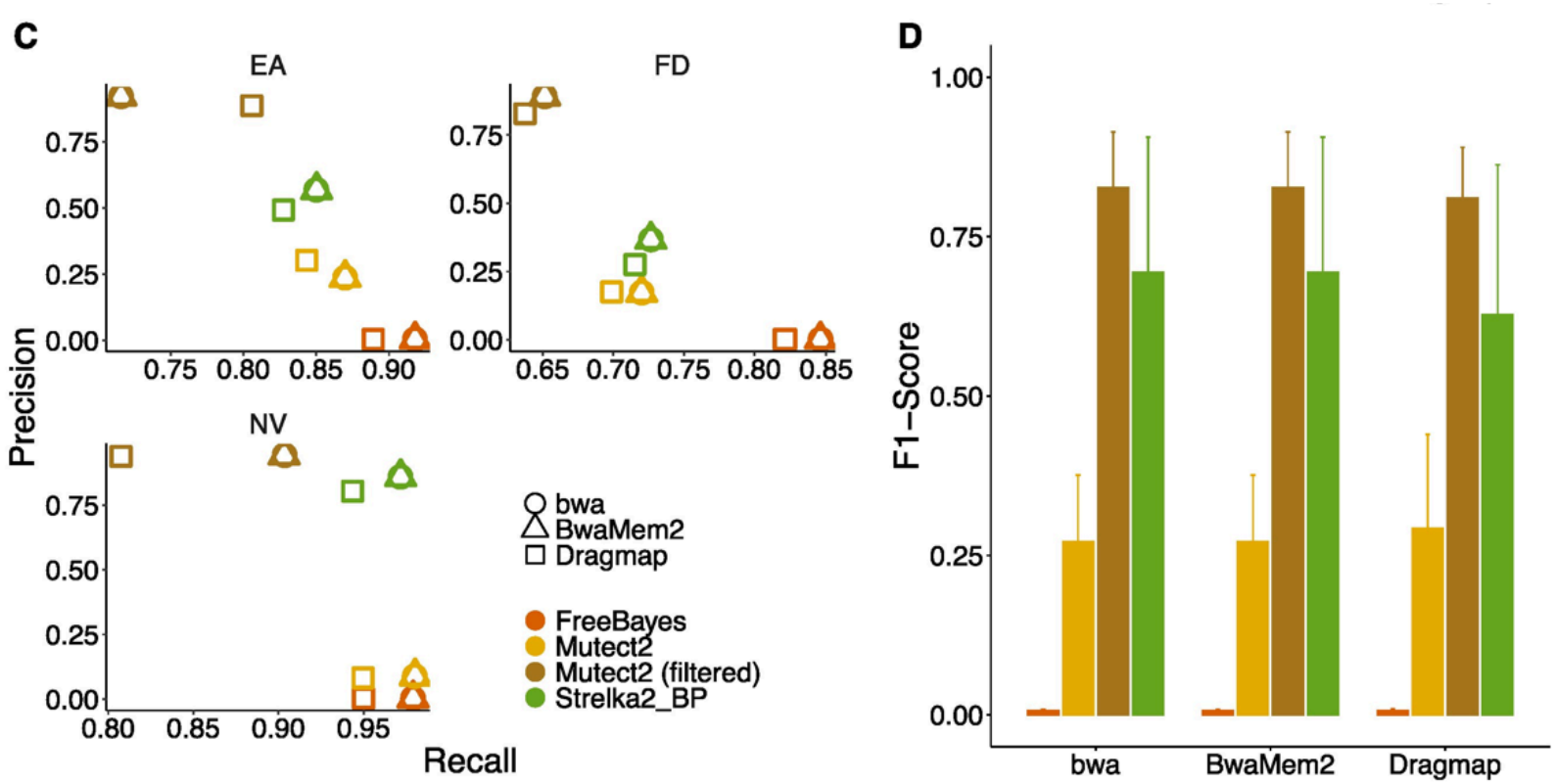
Read Alignment: Tool Performance for Germline WGS [4]

- Negligible difference in variant calling precision, recall, and F1 metrics between the different alignment tools.

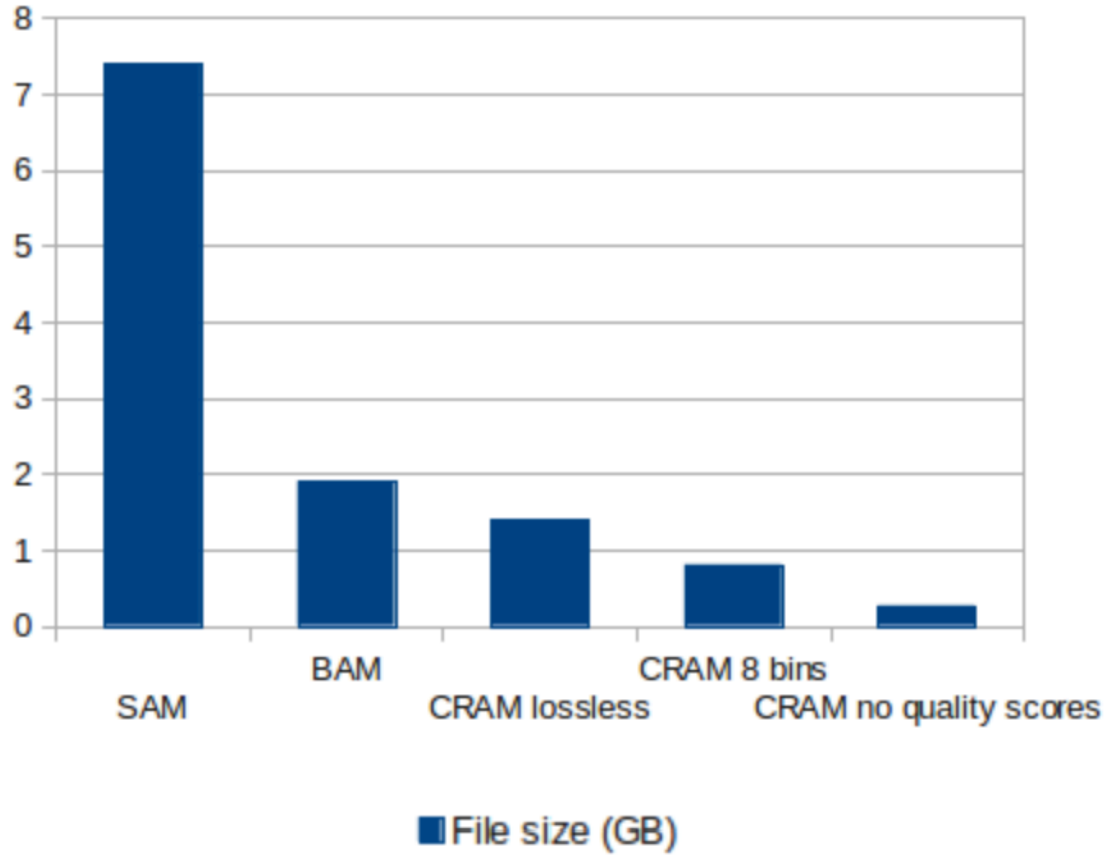


Read Alignment: Tool Performance for Somatic WES (Tumor-Normal Pairs) [4]

- No difference in variant calling precision, recall, and F1 metrics between the different alignment tools.



SAM/BAM/CRAM Output Files



- **SAM**: Sequence Alignment Map containing alignment information
- **BAM**: Binary-encoded (compressed) version of SAM;
- **CRAM**: Compressed version of BAM
- BAM/CRAM files are usually 'indexed' for fast retrieval of alignments
  - A `.bai` file will be found beside the `.bam` file

Source

What's Inside a SAM/BAM/CRAM?

1. **Header section**: provides general information about the alignment strategy
2. **Alignment Section**: provides details for each read alignment

```
@HD VN:1.5 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       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Header section

Alignment section

What's Inside a SAM/BAM/CRAM?

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
★ 2	FLAG	Int	$[0,2^{16}-1]$	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	$[0,2^{29}-1]$	1-based leftmost mapping POSition
5	MAPQ	Int	$[0,2^{8}-1]$	MAPping Quality
★ 6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next segment
8	PNEXT	Int	$[0,2^{29}-1]$	Position of the mate/next segment
9	TLEN	Int	$[-2^{29}+1,2^{29}-1]$	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

What's Inside a SAM/BAM/CRAM?

- Flag is a number representing the condition of a read alignment.
- https://samformat.pages.dev/sam-format-flag

#	Decimal	Description of first read
1	1	Read paired
2	2	Read mapped in proper pair
3	4	Read unmapped
4	8	Mate unmapped
5	16	Read reverse strand
6	32	Mate reverse strand
7	64	First in pair
8	128	Second in pair
9	256	Not primary alignment
10	512	Read fails platform/vendor quality checks
11	1024	Read is PCR or optical duplicate
12	2048	Supplementary alignment
Sum	147	

Decimal	Description of second read
1	Read paired
2	Read mapped in proper pair
4	Read unmapped
8	Mate unmapped
16	Read reverse strand
32	Mate reverse strand
64	First in pair
128	Second in pair
256	Not primary alignment
512	Read fails platform/vendor quality checks
1024	Read is PCR or optical duplicate
2048	Supplementary alignment
99	

Common flags*

One of the reads is unmapped:
73, 133, 89, 121, 165, 181, 101, 117, 153, 185, 69, 137

Both reads are unmapped:
77, 141

Mapped within the insert size and in correct orientation:
99, 147, 83, 163

Mapped within the insert size but in wrong orientation:
67, 131, 115, 179

Mapped uniquely, but with wrong insert size:
81, 161, 97, 145, 65, 129, 113, 177

* Collected from here

What's Inside a SAM/BAM/CRAM? [5]

- CIGAR strings explain how the query (or "read") sequence aligns to the reference genome.

```
Ref   CTGGCCATTATCTC--GGTGGTAGGACATGGCATGCCC
Read        aaATGTCGCGGTG.TAGGAggatcc
```

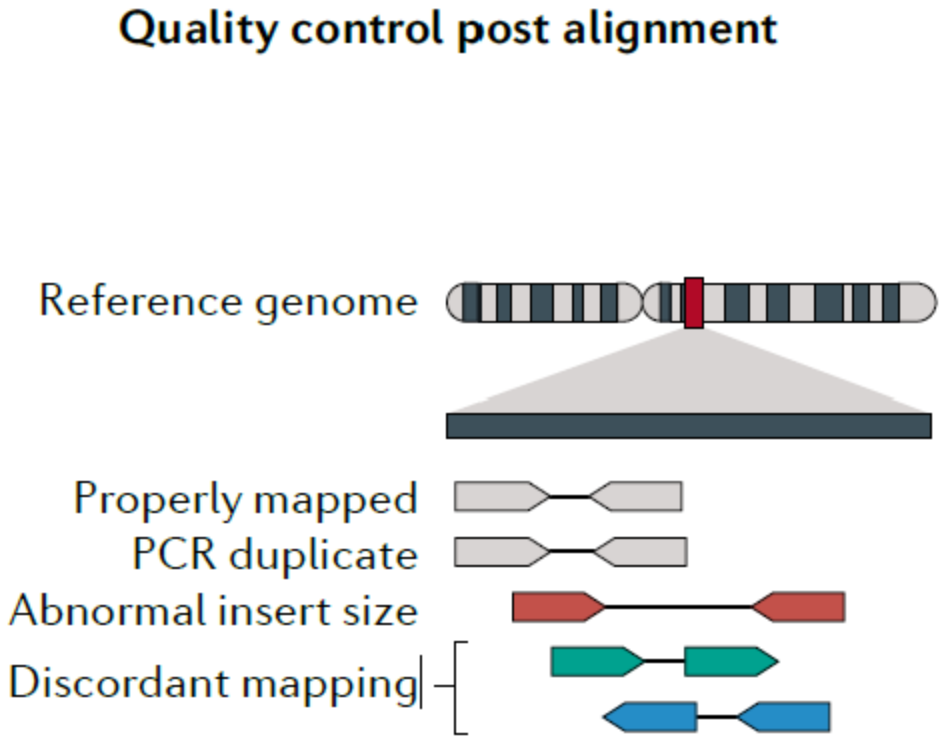
2S|5M|2I|4M|1D|5M|6S

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
* H	5	hard clipping (clipped sequences NOT present in SEQ)
* P	6	padding (silent deletion from padded reference)
* =	7	sequence match
* X	8	sequence mismatch

"N" indicates splicing event in RNAseq BAMs

*Rarer / newer

CIGAR = "Concise Idiosyncratic Gapped Alignment Report"



Cortés-Ciriano et al. 2022

Post Alignment QC Metrics - Mapping Quality

1. Mapping Rate
2. Coverage Depth
3. Duplication Levels
4. Insert Size Distribution

Mapping Quality (MAPQ)

- **A score that indicates how likely it is that a read is misaligned**
  `MAPQ = -10log10(p)`, where p is the misalignment probability
- Ranges from 0 to 60 in many aligners (though some can exceed 60).
- **Higher** values indicate **greater** confidence that the read is aligned correctly and uniquely.

Why do we care?

- Reads with low MAPQ are more likely to be mapped to the wrong location or align equally well to multiple regions.
- Many pipelines filter out alignments below a certain MAPQ threshold (e.g., 20 or 30) to improve confidence in variant calling or downstream analyses.

Mapping Rate

- Percentage of reads successfully aligned to the reference.

Why do we care?

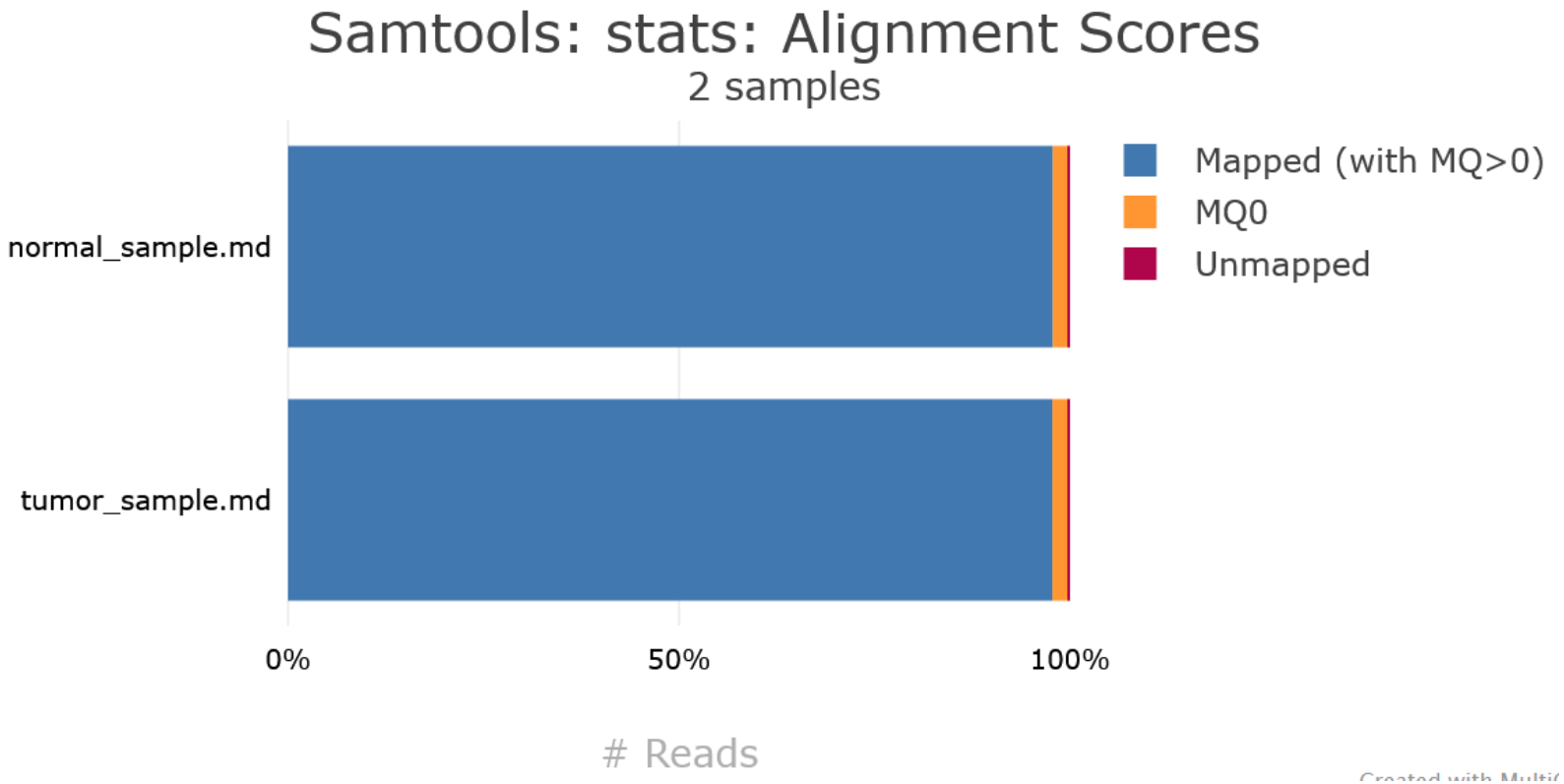
- Low mapping rate may indicate contamination, poor-quality reads, or reference-genome mismatch affecting variant-calling accuracy.

Tool：`samtools flagstat`

- Provides quick statistics on the number of mapped reads, properly paired reads, duplicates, etc.

Mapping Rate

What to look for: A high proportion of properly paired reads (for paired-end) and minimal unmapped reads.



Coverage Depth

- Average number of reads covering each base or region.

Why do we care?

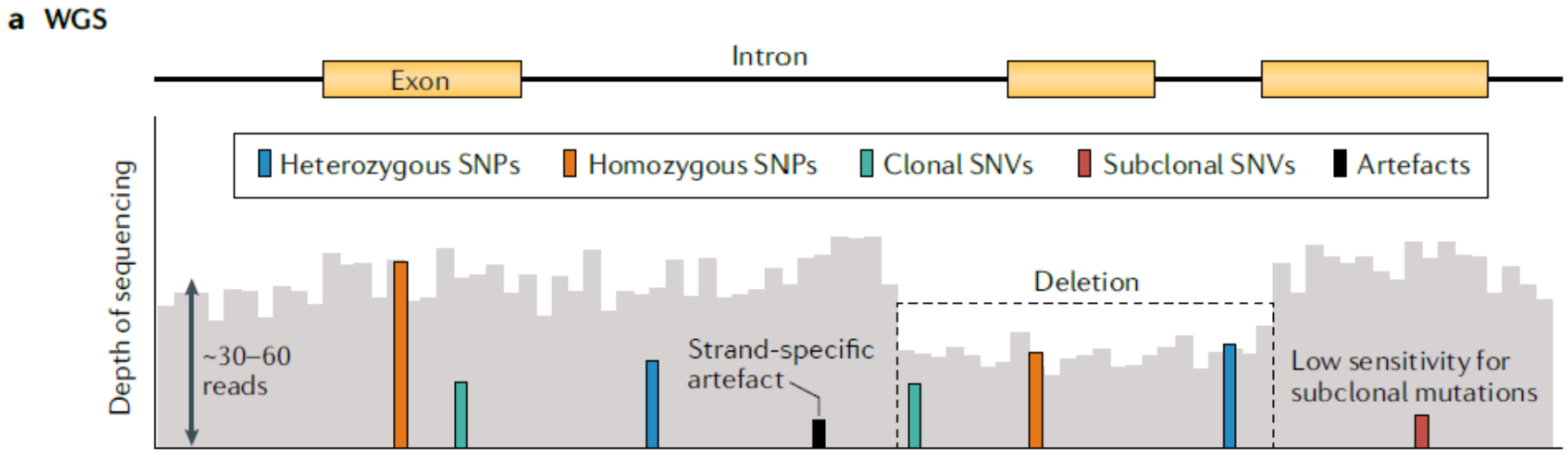
- We want regions of interest (e.g., exons, target panels) to have sufficient reads for variant detection. **Increasing sequencing depth** can increase the power to detect variants and **reduce the false-discovery rate** for variant calling.[6]
- "Coverage gaps" might call for re-sequencing or caution in biological interpretation.

Tool: `mosdepth`

- Calculates coverage depth across genome or targeted regions.

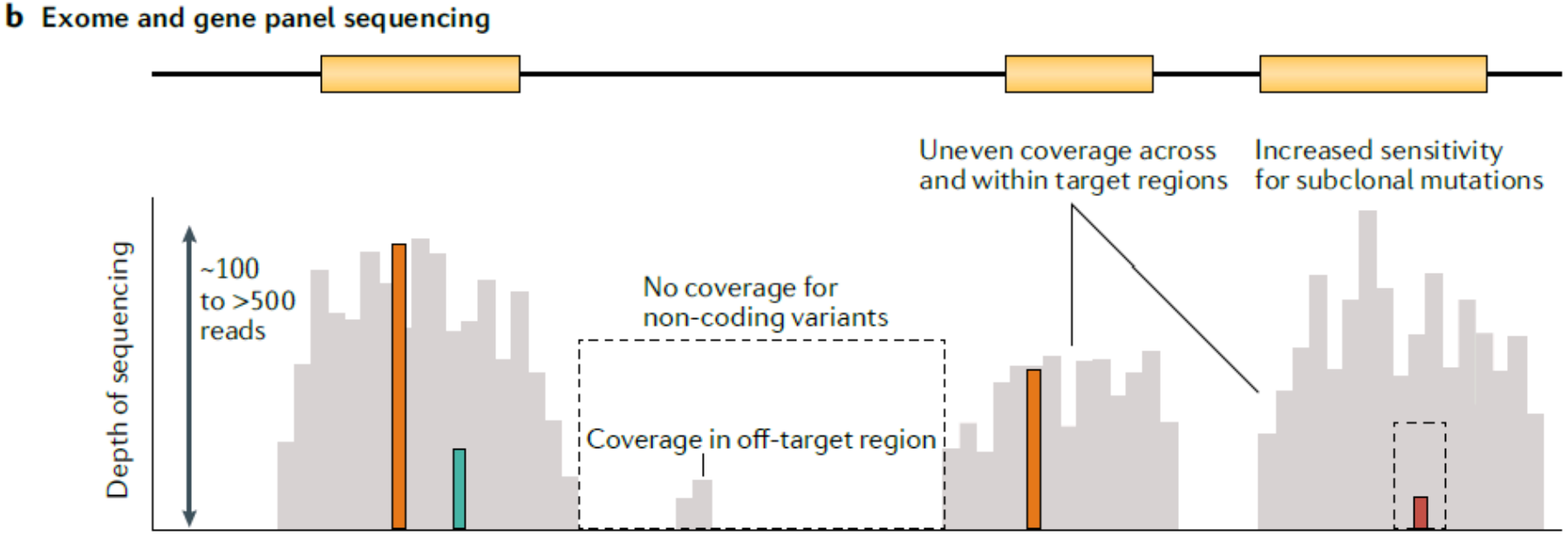
Coverage Depth - (WGS) What to look for [6]

- WGS provides **nearly uniform depth of coverage** (~30–60×) across the genome.
- Subclonal SNVs may be missed when the number of reads containing the mutation is too small or comparable with the number of reads containing artefacts.



Coverage Depth - (WES/TGS) What to look for [6]

- **It is normal for exome sequence coverage to have gaps in untranslated regions.**
- Exome sequencing (~100–200×) and gene panel sequencing of cancer-related genes (~500–1,000×) have higher depth of sequencing



Read Duplication

- Fraction of reads that appear to be duplicates (often from PCR amplification).

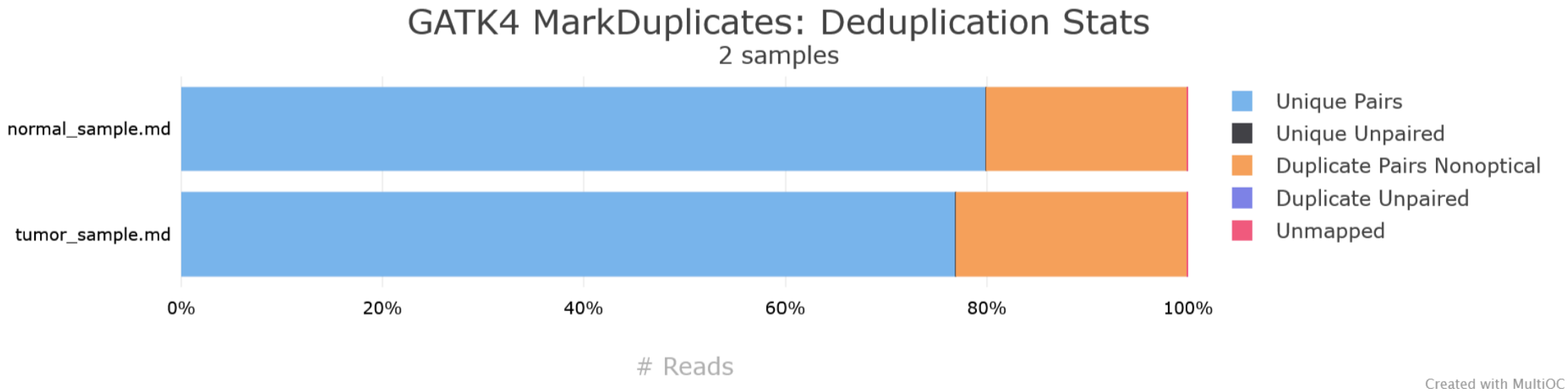
Why do we care?

- Removing duplicates improves true coverage to ensure variant calling accuracy.
- A high duplication rate be caused by over-amplification in library prep or technical bias.

Tool: `Picard MarkDuplicates` flags or removes duplicates

Read Duplication

What to look for: The ideal read duplicate % is **<30%**.[7]



Insert Size Distribution (Paired-End)

- Measures the fragment length between forward and reverse read pairs.

Why do we care?

- Abnormal insert sizes could indicate contamination, structural variants, or library prep inconsistencies.

Tool: `samtools stats` provides detailed alignment statistics, including insert size and base composition.

What to look for: Do the observed insert sizes match library prep method?

Variant Filtering: Base recalibration

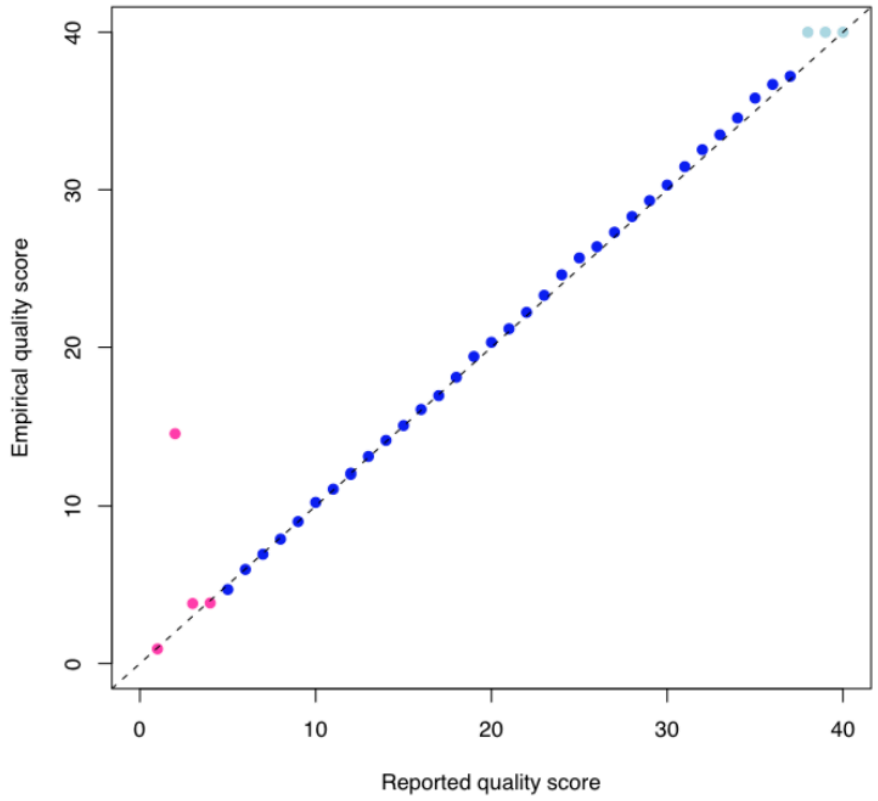
Base quality scores: per-base PHRED score indicating how confident the sequencing machine was at calling the correct base each time

- Short variant calling algorithms rely heavily on the quality score as a hard filter

What could go wrong?
The scores produced by the machines are subject to various sources of systematic (non-random) technical error, leading to over- or under-estimated base quality scores in the data.

- **Over-estimated quality scores can lead to false postive detection of variants.**
- **Under-estimated scores can mask low-frequency variants.**

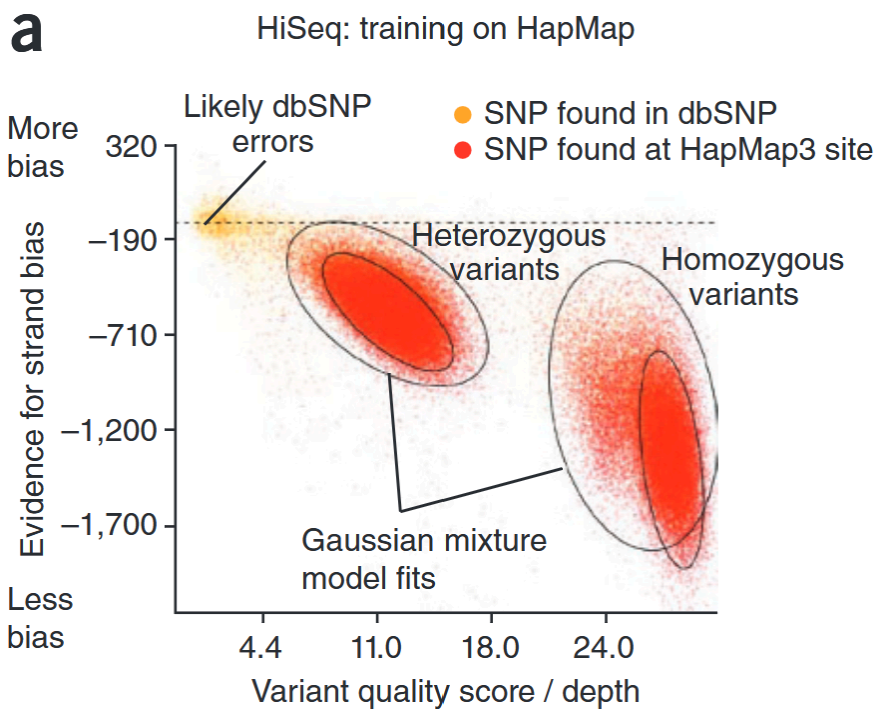
Variant Filtering: Base Quality Score Recalibration (BQSR)



1. **BaseRecalibrator**:
   - Uses reference genome to distinguish true mismatches from sequencer biases.
   - Creates a recalibration table by comparing observed versus expected error rates across multiple covariates (e.g., read cycle, sequence context).
2. **ApplyBQSR**:
   - Adjusts base quality scores in the BAM/CRAM using the recalibration table.

Variant Filtering: Variant Quality Score Recalibration (VQSR)

Not in `nf-core/sarek`



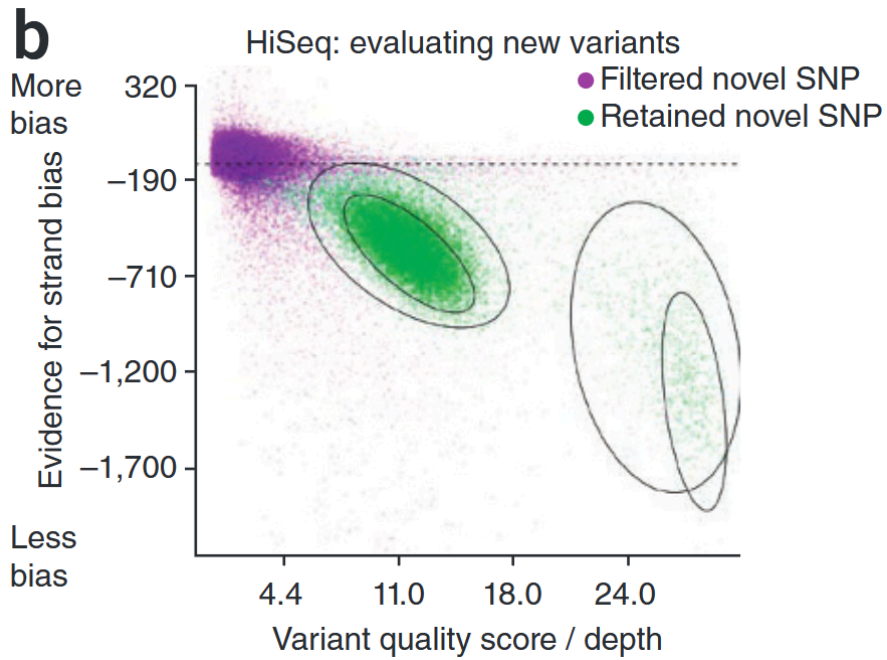
DePristo et al. 2011

1. **VariantRecalibrator**

- Learns the distribution of annotation values for high-confidence “true” variant sites using reference variant data as training sets. --> **maintains sensitivity**
- Constructs a Gaussian mixture model that differentiates likely true variants from false positives.

Variant Filtering: Variant Quality Score Recalibration (VQSR)

Not in `nf-core/sarek`



DePristo et al. 2011

2. **ApplyVQSR**

- Applies the model to the full variant set to produces recalibrated scores or filters that retain high-confidence variants and remove suspicious calls.
- Fewer false positives by penalizing variants with suspicious annotation profiles.->**better specificity**

Variant Filtering: BQSR + VQSR

1. Perform **Read-Level Correction (BQSR)** after alignment, before variant calling.
    - Make sure base qualities fed into the variant caller accurately reflect real sequencing errors.
2. Perform **Variant-Level Refinement (VQSR)** after generating raw variant calls
    - Use reference variant data to make sure the scores for false positive variants are penalized.

Why do we need this?

- BQSR and VQSR together increase the accuracy of base quality scores used in variant calling algorithms.

Variant Calling

- The process of identifying genetic variants—differences from a reference genome—in sequencing data
- Typically focuses on SNPs (Single Nucleotide Polymorphisms) and indels (insertions/deletions), although some pipelines also address structural variants.

Variant calling

1. **Germline variant calling**: the process of calling variants that are ubiquitous across the organism (i.e. almost all cells carry these variants)
2. **Somatic variant calling**: the process of calling variants that differ between cells within a single organism and these variants are not passed through the germline.
    - More difficult than germline variant calling because low frequency variants and sequencing artifacts are difficult to distinguish from sequencing errors.

Germline Variant Callers

Characteristic	Germline Variant Callers
Purpose	Identify inherited variants present in all cells; assume a stable, diploid (or polyploid) genome.
Allele Frequency	Expect high allele fractions: heterozygous (~50%) or homozygous (~100%).
Data Requirements	Typically run on single samples or cohorts; analyses are usually unpaired.
Statistical Models	Rely on models that assume a fixed ploidy and well-defined genotype priors.
Variant Filtering	Use population databases (e.g., dbSNP, 1kGP) and quality metrics to filter out sequencing artifacts.

Somatic Variant Callers

Characteristic	Somatic Variant Callers
Purpose	Detect variants acquired in specific tissues (e.g., tumors); variants may be present in only a subset of cells.
Allele Frequency	Often exhibit low and variable allele frequencies due to tumor heterogeneity, subclonality, or mosaicism.
Data Requirements	Frequently require tumor-normal paired samples to distinguish somatic mutations from germline variants.
Statistical Models	Must account for mixed cell populations, variable purity, and complex copy number alterations.
Variant Filtering	Incorporate filters for strand bias, low variant allele frequency, and tumor-specific artifacts.

`nf-core/sarek` has a table of summary for tool compatibility [8]

Tool	WGS	WES	Panel	Germline	Tumor-Only	Somatic (Tumor-Normal)
DeepVariant	x	x	x	x	-	-
FreeBayes	x	x	x	x	x	x
GATK HaplotypeCaller	x	x	x	x	-	-
GATK Mutect2	x	x	x	-	x	x
lofreq	x	x	x	-	x	-
mpileup	x	x	x	x	x	-
Strelka	x	x	x	x	-	x
Manta	x	x	x	x	x	x
indexcov	x	-	-	x	-	x
TIDDIT	x	x	x	x	x	x
ASCAT	x	x	-	-	-	x
CNVKit	x	x	-	x	x	x
Control-FREEC	x	x	x	-	x	x
MSIsensorPro	x	x	x	-	-	x

How to use this table?

- Pick at least one tool for detecting each of SNPs/indels, structural variants, and copy-number.

Tool	WGS	WES	Panel	Germline	Tumor-Only	Somatic (Tumor-Normal)
DeepVariant	x	x	x	x	-	-
FreeBayes	x	x	x	x	x	x
GATK HaplotypeCaller	x	x	x	x	-	-
GATK Mutect2	x	x	x	-	x	x
lofreq	x	x	x	-	x	-
mpileup	x	x	x	x	x	-
Strelka	x	x	x	x	-	x
Manta	x	x	x	x	x	x
indexcov	x	-	-	x	-	x
TIDDIT	x	x	x	x	x	x
ASCAT	x	x	-	-	-	x
CNVKit	x	x	-	x	x	x
Control-FREEC	x	x	x	-	x	x
MSIsensorPro	x	x	x	-	-	x