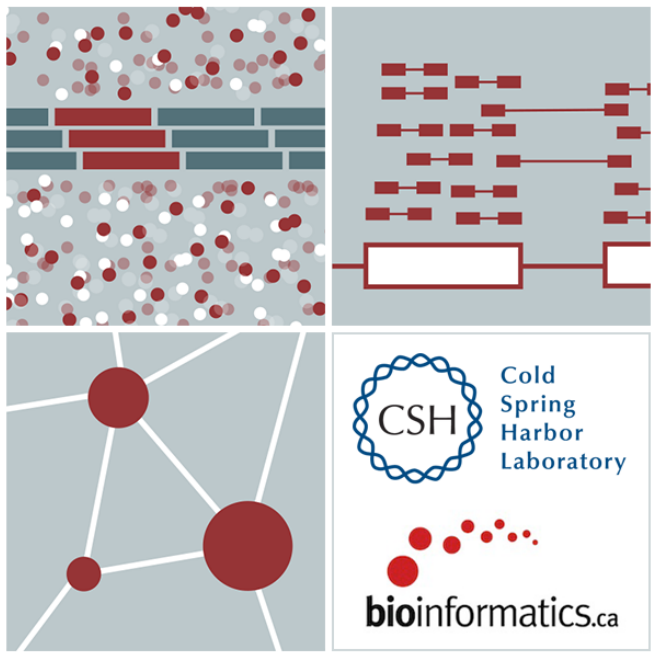
Cold Spring Harbor Laboratory

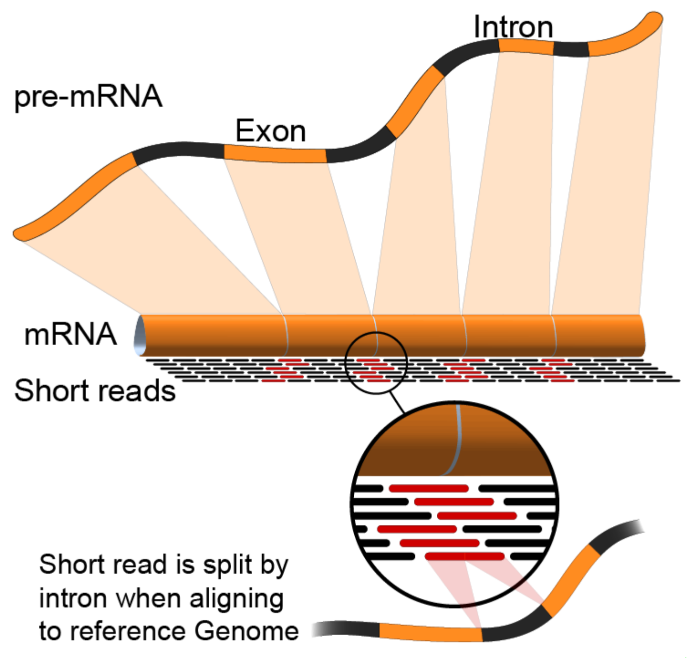
PMBIO Module 3

# DNA Alignment

Felicia Gomez, Obi Griffith, Malachi Griffith, Chris Miller
Elizabeth Garfinkle, My Hoang, Charlz Jerold, Mariam Khanfar,
Kartik Singhal, Jennie Yao

Advanced Sequencing Technologies & Bioinformatics Analysis. November 9-22, 2025

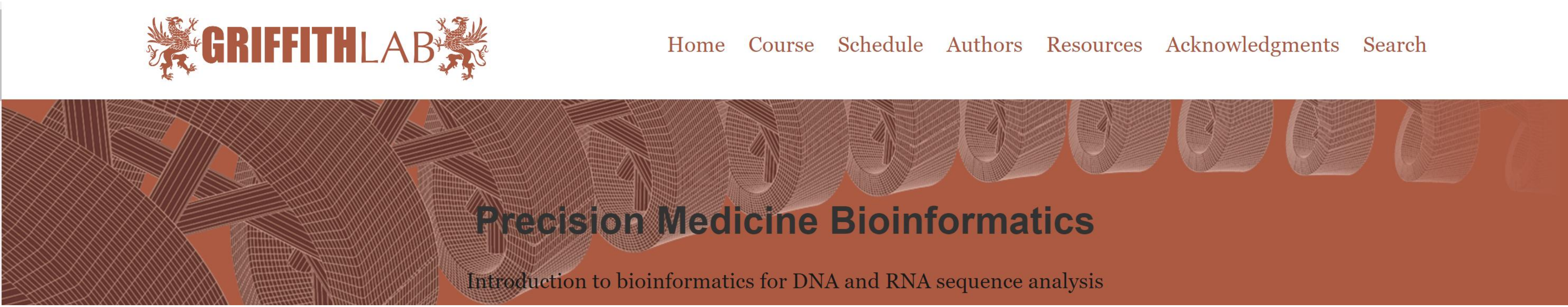



Intron
pre-mRNA
Exon
mRNA
Short reads
Short read is split by intron when aligning to reference Genome

Washington University in St. Louis
School of Medicine

# Learning Objectives

- Understand the process of obtaining and indexing raw DNA sequence data for alignment
- Perform a quality assessment using the aligned data and understand clean-up steps
- Prepare and visualize alignment results using genome viewers



Precision Medicine Bioinformatics

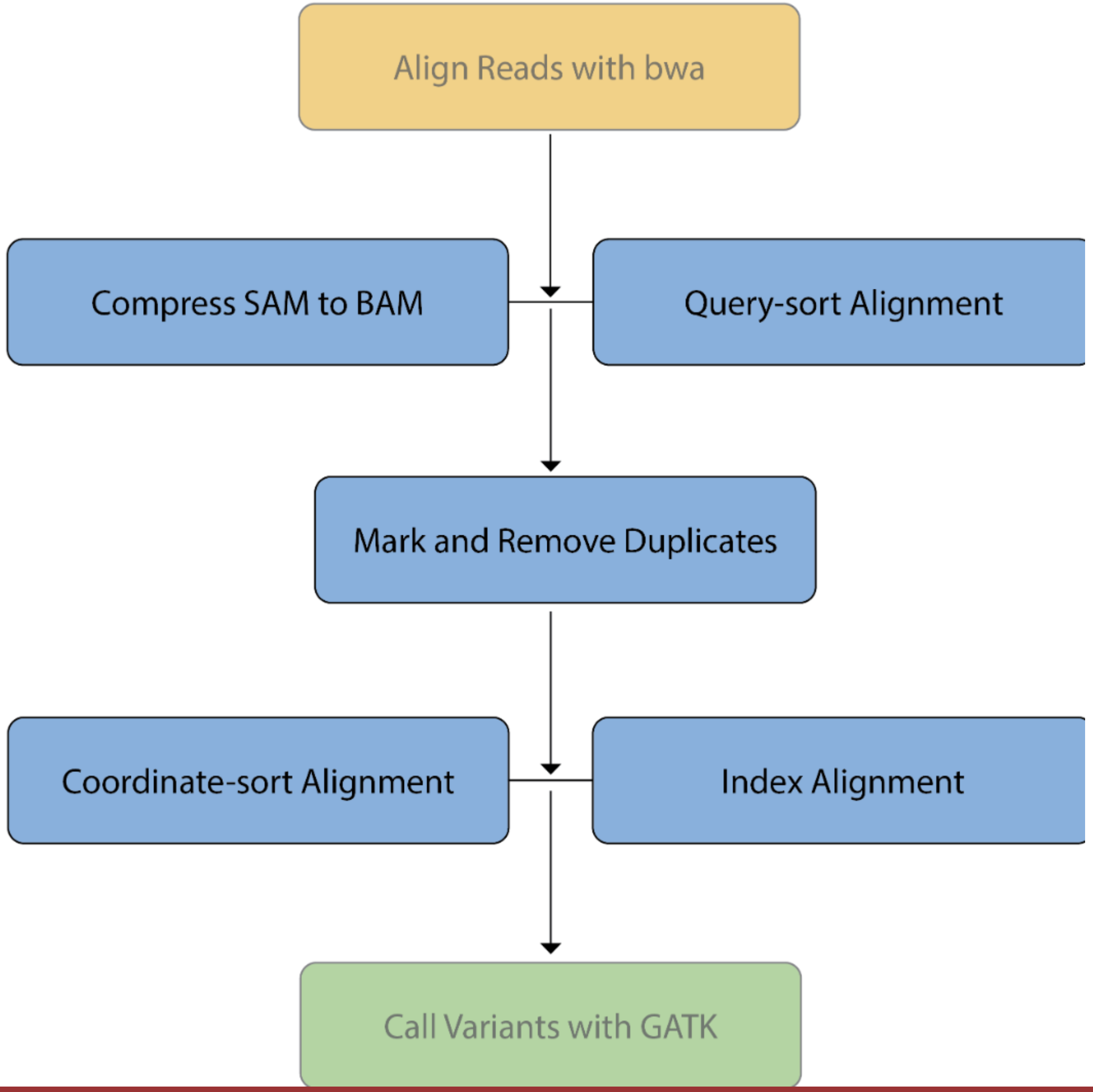
Introduction to bioinformatics for DNA and RNA sequence analysis

Based on: https://pmbio.org/module-03-align/0003/02/01/Alignment/

## For our exercise:

https://gist.github.com/MariamKhanfar/8eae80dfc5bd9011d2ccbab27458ca04

# Main QC steps in DNA alignment



# Overview of main steps

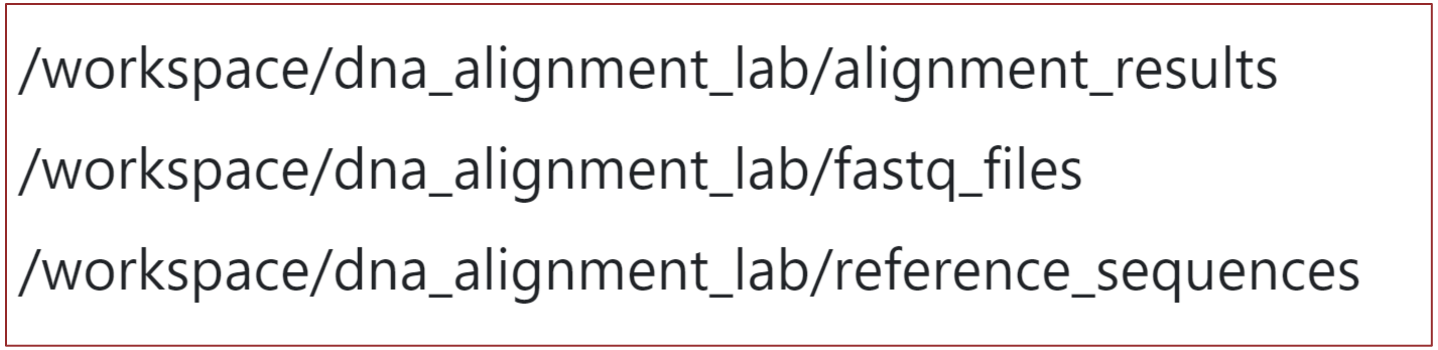
- Create directories
- Download the data
- Download the reference
- Index Reference with BWA
- Align the data to the reference → Result in SAM
- Convert SAM to BAM
- Sort by query name
- Mark duplicates
- Sort by coordinates
- Index the final BAM
- Generate a flagstat report for the BAM

# Overview of main steps

- Create directories
- Download the data
- Download the reference
- Index Reference with BWA
- Align the data to the reference → Result in SAM
- Convert SAM to BAM
- Sort by query name
- Mark duplicates
- Sort by coordinates
- Index the final BAM
- Generate a flagstat report for the BAM

# Prepare directories for analysis and obtain data

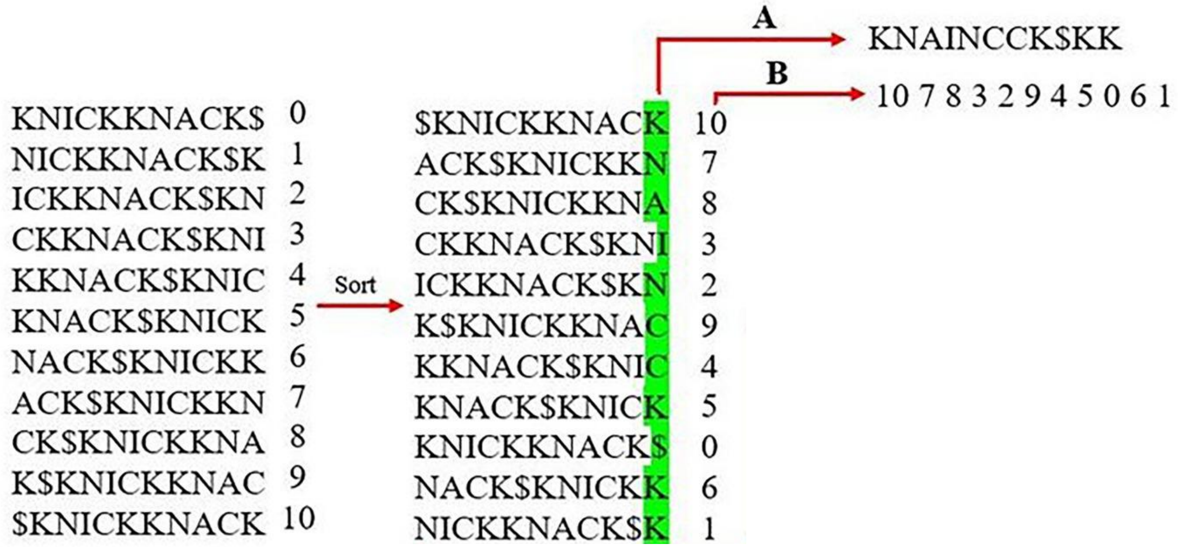
- Create the main directory “*dna_alignment_lab*”
- Create subdirectories within “*dna_alignment_lab*” for organizing data



```
/workspace/dna_alignment_lab/alignment_results
/workspace/dna_alignment_lab/fastq_files
/workspace/dna_alignment_lab/reference_sequences
```

# Index reference genome

- Indexing is like creating a map of the reference genome, allowing the aligner to quickly locate where a sequence might fit; finds information without reading the entire text



| | |
|---|---|
| KNICKKNACK$ | 0 |
| NICKKNACK$K | 1 |
| ICKKNACK$KN | 2 |
| CKKNACK$KNI | 3 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| NACK$KNICKK | 6 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| K$KNICKKNAC | 9 |
| $KNICKKNACK | 10 |

Sort

| | |
|---|---|
| $KNICKKNACK | 10 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| CKKNACK$KNI | 3 |
| ICKKNACK$KN | 2 |
| K$KNICKKNAC | 9 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| KNICKKNACK$ | 0 |
| NACK$KNICKK | 6 |
| NICKKNACK$K | 1 |

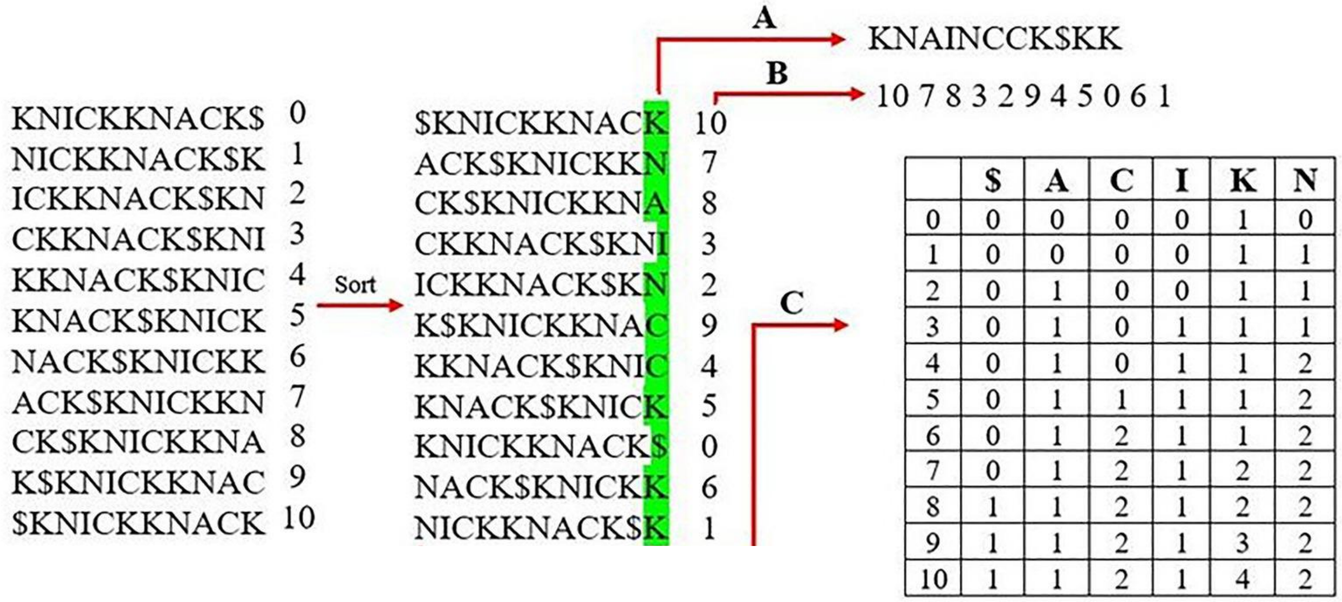
A → KNAINCCK$KK

B → 10 7 8 3 2 9 4 5 0 6 1

https://www.frontiersin.org/articles/10.3389/fpls.2021.657240/full

# Index reference genome

- Indexing is like creating a map of the reference genome, allowing the aligner to quickly locate where a sequence might fit; finds information without reading the entire text



| | |
|---|---|
| KNICKKNACK$ | 0 |
| NICKKNACK$K | 1 |
| ICKKNACK$KN | 2 |
| CKKNACK$KNI | 3 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| NACK$KNICKK | 6 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| K$KNICKKNAC | 9 |
| $KNICKKNACK | 10 |

| | |
|---|---|
| $KNICKKNACK | 10 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| CKKNACK$KNI | 3 |
| ICKKNACK$KN | 2 |
| K$KNICKKNAC | 9 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| KNICKKNACK$ | 0 |
| NACK$KNICKK | 6 |
| NICKKNACK$K | 1 |

KNAINCCK$KK

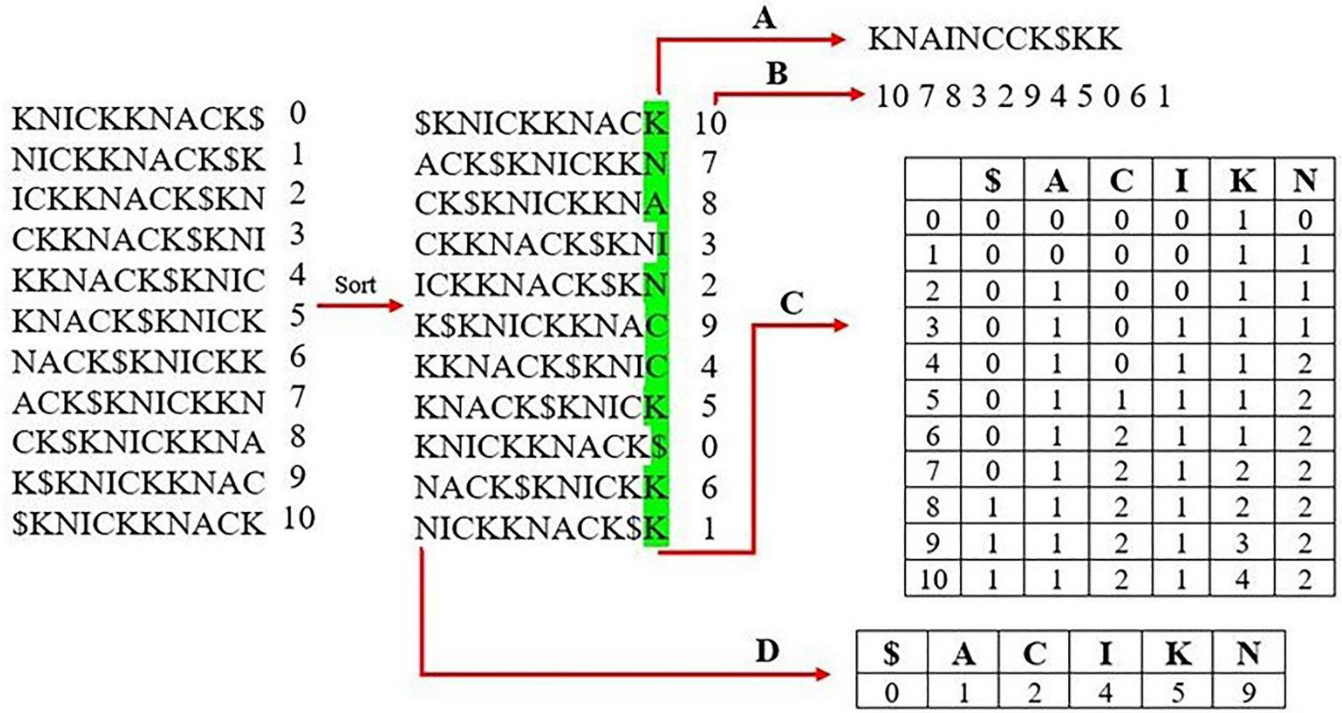
10 7 8 3 2 9 4 5 0 6 1

| | $ | A | C | I | K | N |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 1 | 1 | 2 |
| 5 | 0 | 1 | 1 | 1 | 1 | 2 |
| 6 | 0 | 1 | 2 | 1 | 1 | 2 |
| 7 | 0 | 1 | 2 | 1 | 2 | 2 |
| 8 | 1 | 1 | 2 | 1 | 2 | 2 |
| 9 | 1 | 1 | 2 | 1 | 3 | 2 |
| 10 | 1 | 1 | 2 | 1 | 4 | 2 |

https://www.frontiersin.org/articles/10.3389/fpls.2021.657240/full

# Index reference genome

- Indexing is like creating a map of the reference genome, allowing the aligner to quickly locate where a sequence might fit; finds information without reading the entire text



| | |
|---|---|
| KNICKKNACK$ | 0 |
| NICKKNACK$K | 1 |
| ICKKNACK$KN | 2 |
| CKKNACK$KNI | 3 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| NACK$KNICKK | 6 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| K$KNICKKNAC | 9 |
| $KNICKKNACK | 10 |

| | |
|---|---|
| $KNICKKNACK | 10 |
| ACK$KNICKKN | 7 |
| CK$KNICKKNA | 8 |
| CKKNACK$KNI | 3 |
| ICKKNACK$KN | 2 |
| K$KNICKKNAC | 9 |
| KKNACK$KNIC | 4 |
| KNACK$KNICK | 5 |
| KNICKKNACK$ | 0 |
| NACK$KNICKK | 6 |
| NICKKNACK$K | 1 |

KNAINCCK$KK

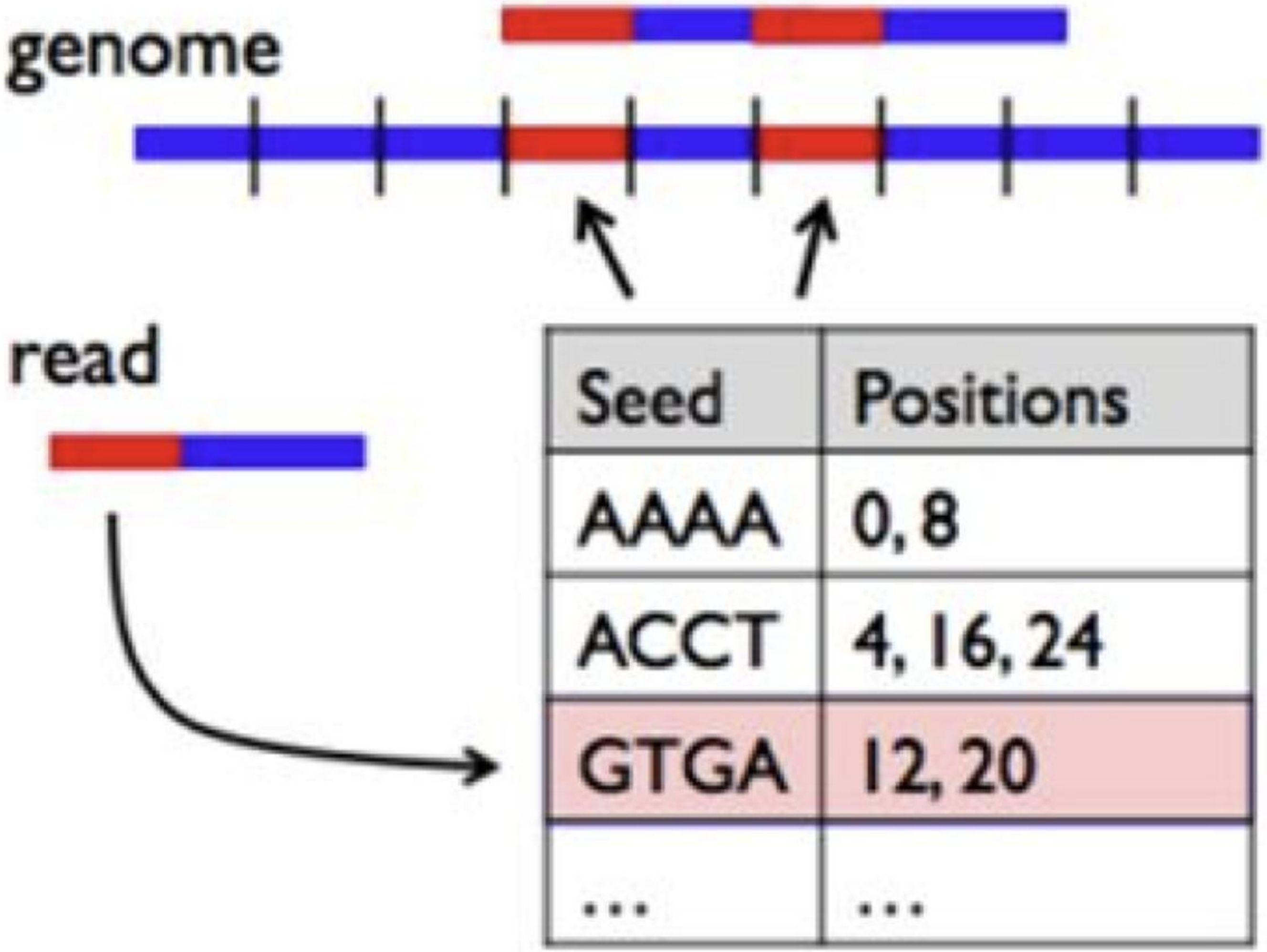
10 7 8 3 2 9 4 5 0 6 1

| | $ | A | C | I | K | N |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 1 | 1 | 2 |
| 5 | 0 | 1 | 1 | 1 | 1 | 2 |
| 6 | 0 | 1 | 2 | 1 | 1 | 2 |
| 7 | 0 | 1 | 2 | 1 | 2 | 2 |
| 8 | 1 | 1 | 2 | 1 | 2 | 2 |
| 9 | 1 | 1 | 2 | 1 | 3 | 2 |
| 10 | 1 | 1 | 2 | 1 | 4 | 2 |

| $ | A | C | I | K | N |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 4 | 5 | 9 |

https://www.frontiersin.org/articles/10.3389/fpls.2021.657240/full

# DNA alignment with BWA



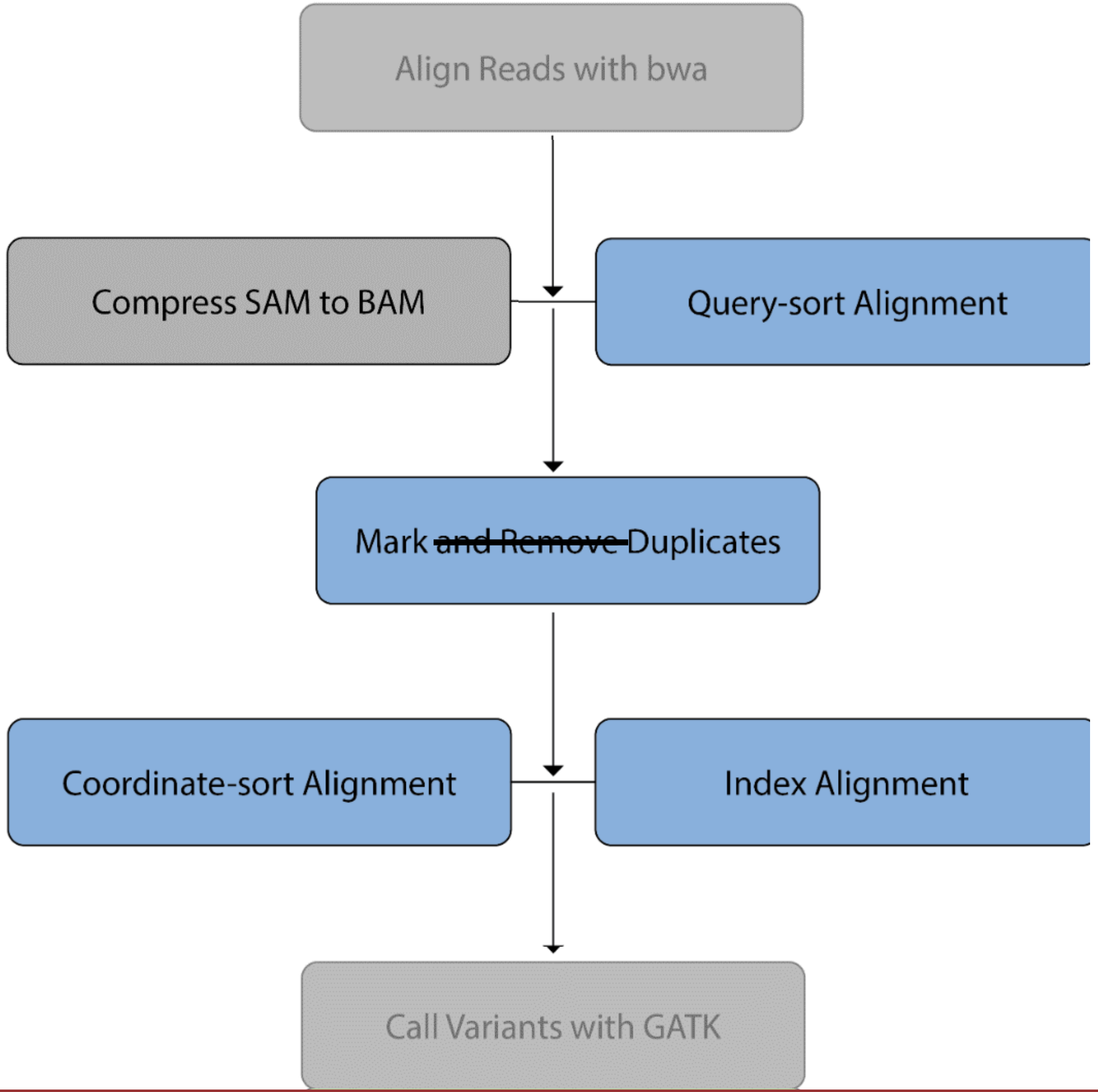
| Seed | Positions |
|---|---|
| AAAA | 0, 8 |
| ACCT | 4, 16, 24 |
| GTGA | 12, 20 |
| ... | ... |

https://rbatorsky.github.io/intro-to-ngs-bioinformatics/lessons/03_Alignment.html

# Convert SAM to BAM

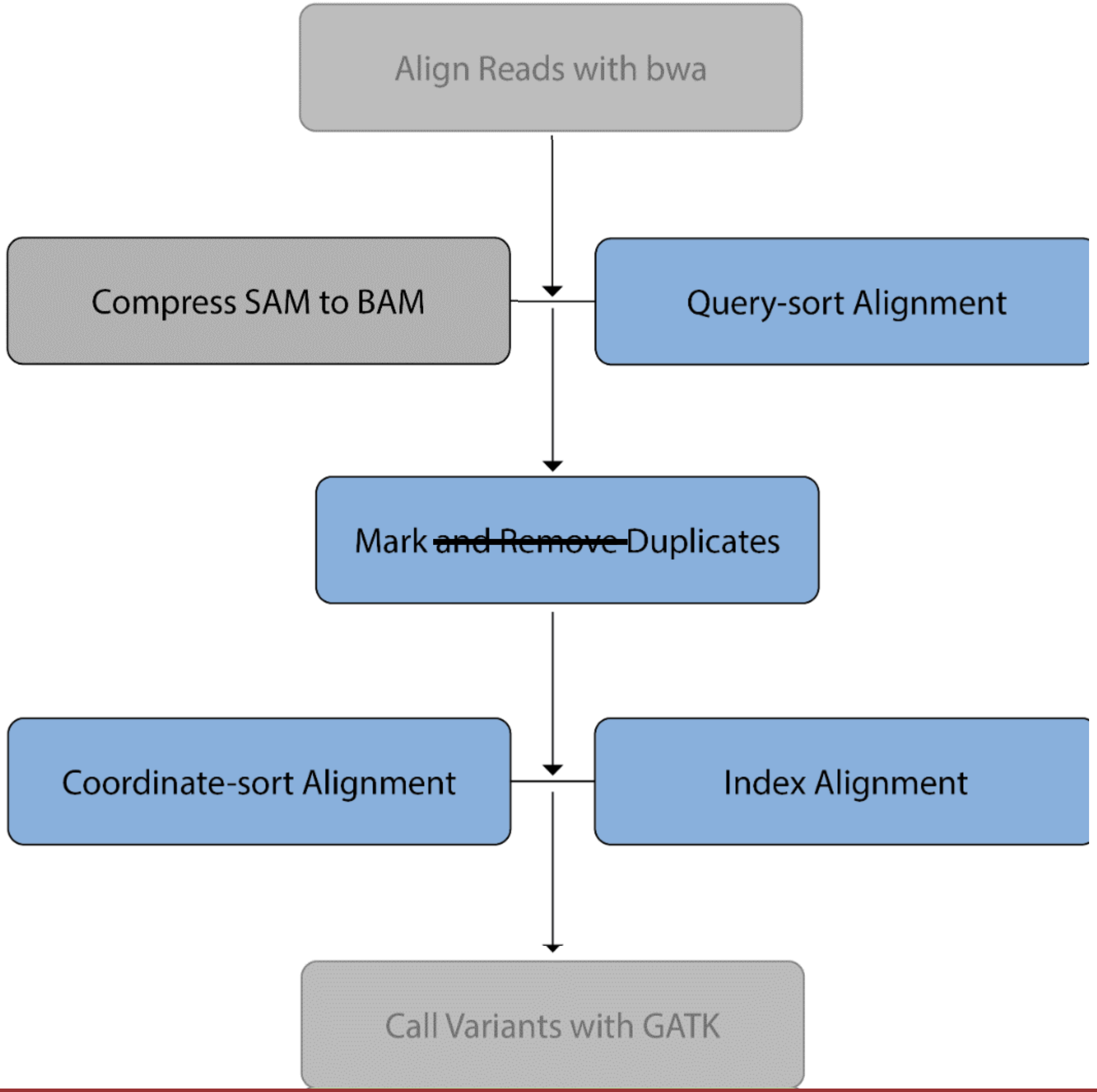
- SAM files are human-readable alignments, but they're large and unwieldy
- BAM files are the binary version of SAM, much smaller and faster for computers to process

# Main QC steps in DNA alignment



https://hbctraining.github.io/variant_analysis/lessons/06_alignment_file_processing.html

# Main QC steps in DNA alignment



Align Reads with bwa

Compress SAM to BAM

Query-sort Alignment

Mark ~~and Remove~~ Duplicates

Coordinate-sort Alignment

Index Alignment

Call Variants with GATK

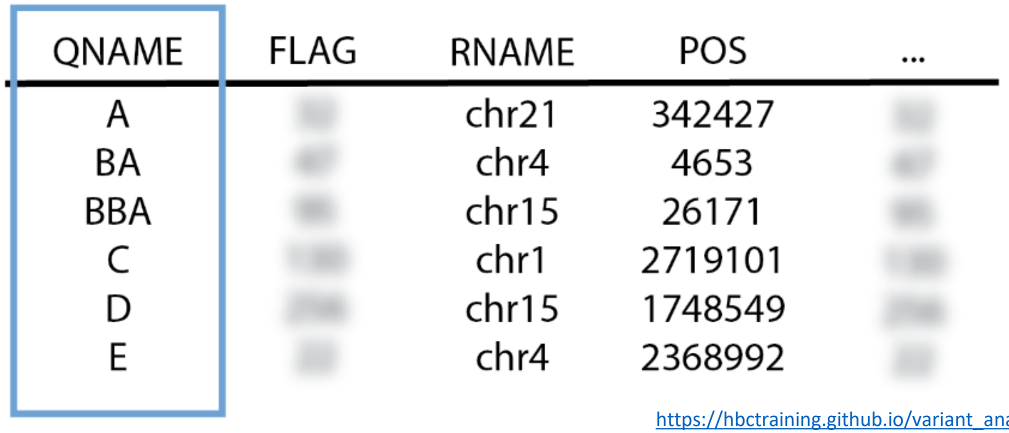
**Can be done on:**
**SAM, BAM, CRAM**

https://hbctraining.github.io/variant_analysis/lessons/06_alignment_file_processing.html

# Sort by query name

- Grouping together reads that came from the same DNA fragment (i.e. identifier given to each read during sequencing)
- This is important for **marking duplicates**, as it is essential to account for biases in sequencing where some fragments are overrepresented.

Query-sorted



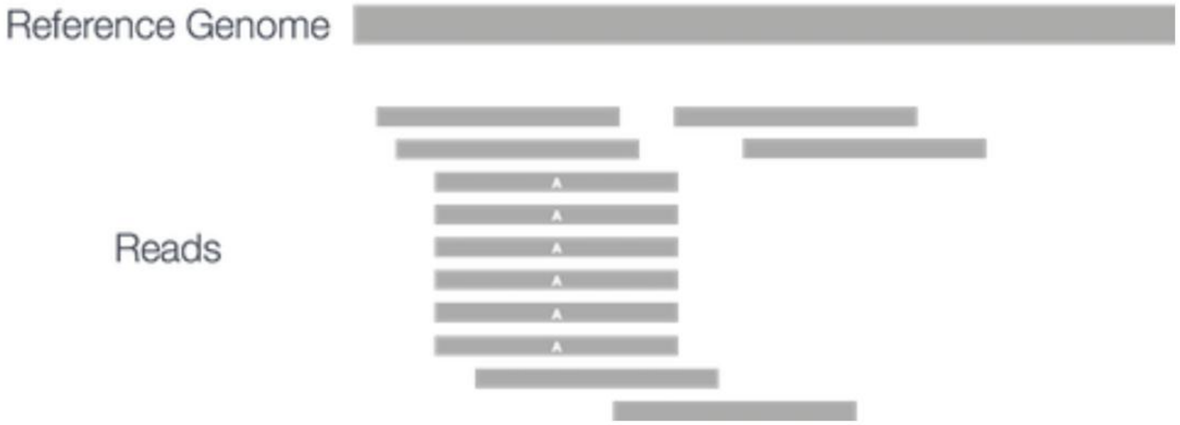
| QNAME | FLAG | RNAME | POS | ... |
|---|---|---|---|---|
| A | [illegible] | chr21 | 342427 | [illegible] |
| BA | [illegible] | chr4 | 4653 | [illegible] |
| BBA | [illegible] | chr15 | 26171 | [illegible] |
| C | [illegible] | chr1 | 2719101 | [illegible] |
| D | [illegible] | chr15 | 1748549 | [illegible] |
| E | [illegible] | chr4 | 2368992 | [illegible] |

https://hbctraining.github.io/variant_analysis/lessons/06_alignment_file_processing.html

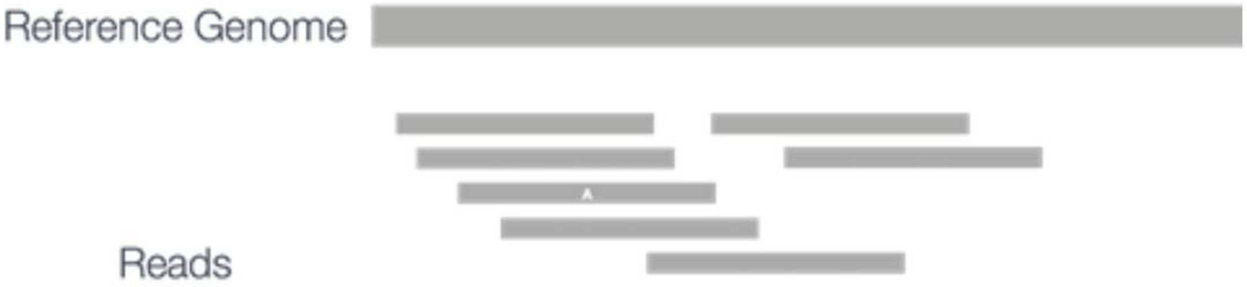
# Mark Duplicates

- To assure proper pair alignment
- Must be used on a name sorted BAM (i.e. mark secondary alignments)
- Locate and tag duplicate reads (both PCR and optical/sequencing-driven)
- Duplicate reads are defined as originating from the same original fragment of DNA
- Produces a metrics file indicating the numbers of duplicates for both single- and paired-end reads.

# Mark Duplicates



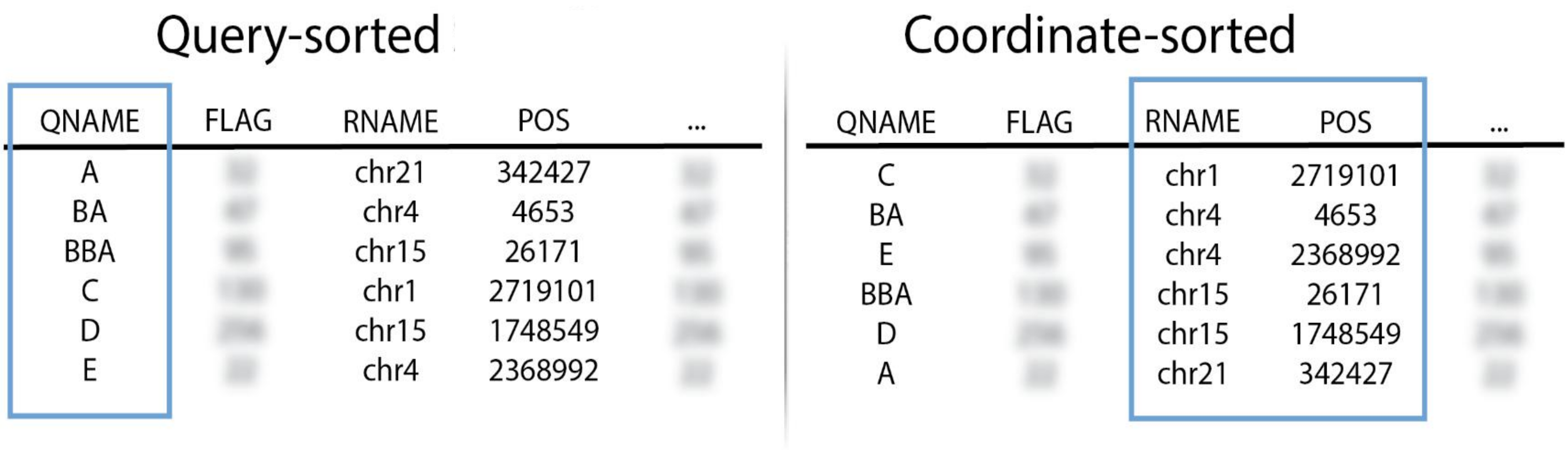
Mark Duplicates →



# Sort by position

- Position sorting is performed to organize the reads by their location on the genome
- This is essential for downstream analysis, which require reads to be in order based on their *chromosomal coordinates*
- Ensure that the subsequent analysis (like variant calling) do not mistakenly count extra copies of the same fragment, which could distort the results
- Position sorting facilitates the identification of regions with genuine high coverage due to biological reasons rather than technical artifacts

# Sort by query name vs position



Query-sorted

| QNAME | FLAG | RNAME | POS | ... |
|---|---|---|---|---|
| A | [illegible] | chr21 | 342427 | [illegible] |
| BA | [illegible] | chr4 | 4653 | [illegible] |
| BBA | [illegible] | chr15 | 26171 | [illegible] |
| C | [illegible] | chr1 | 2719101 | [illegible] |
| D | [illegible] | chr15 | 1748549 | [illegible] |
| E | [illegible] | chr4 | 2368992 | [illegible] |

Coordinate-sorted

| QNAME | FLAG | RNAME | POS | ... |
|---|---|---|---|---|
| C | [illegible] | chr1 | 2719101 | [illegible] |
| BA | [illegible] | chr4 | 4653 | [illegible] |
| E | [illegible] | chr4 | 2368992 | [illegible] |
| BBA | [illegible] | chr15 | 26171 | [illegible] |
| D | [illegible] | chr15 | 1748549 | [illegible] |
| A | [illegible] | chr21 | 342427 | [illegible] |

https://hbctraining.github.io/variant_analysis/lessons/06_alignment_file_processing.html

# Index your final BAM

- In order to efficiently load and search a bam file, downstream applications typically require an index

# SAM Flagstat

- Extract and count the number of reads that are: aligned, primary, not duplicate, and properly paired …etc, based on FLAG field
  https://samtools.github.io/hts-specs/SAMv1.pdf
  https://broadinstitute.github.io/picard/explain-flags.html