

DNA-SEQ: BRCA1/2 TARGET SEQUENCING - UPSTREAM ANALYSIS

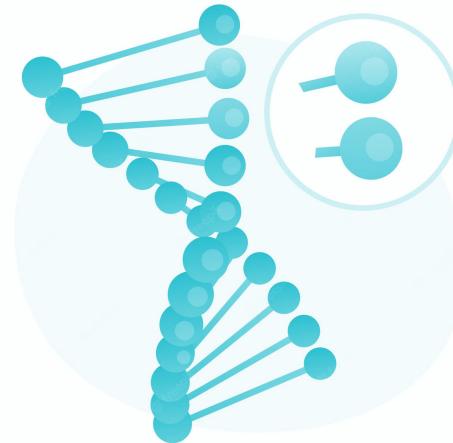


Presenter: Duy Dao

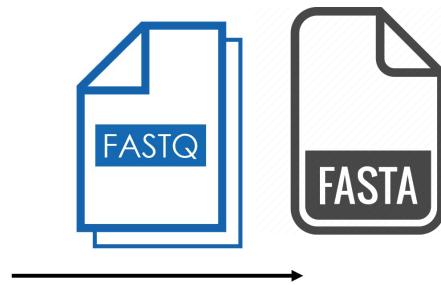
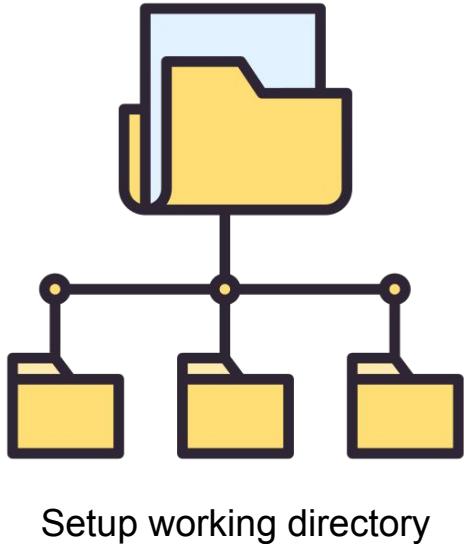
TABLE OF CONTENTS

DNA SEQ: UPSTREAM ANALYSIS

- 01 INTRODUCTION**
- 02 RAW DATA PROCESSING**
- 03 ALIGNMENT**
- 04 MAPPED READS POST-PROCESSING**
- 05 ALIGNMENT DATA: QUALITY CONTROL**



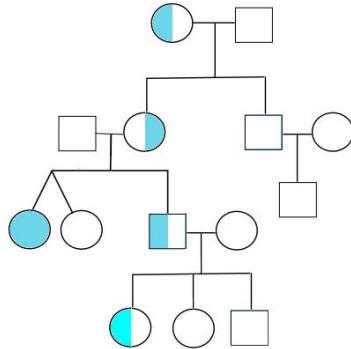
PREPARATION



INTRODUCTION

INTRODUCTION

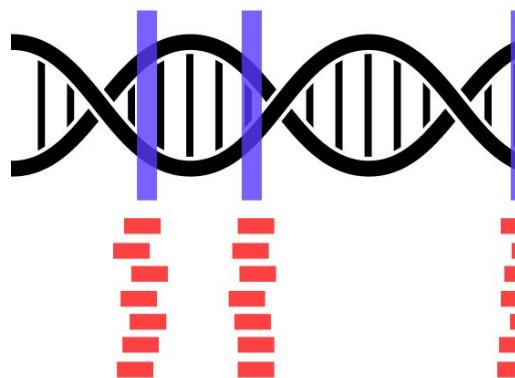
Heredity breast and ovarian cancer
in a pedigree chart



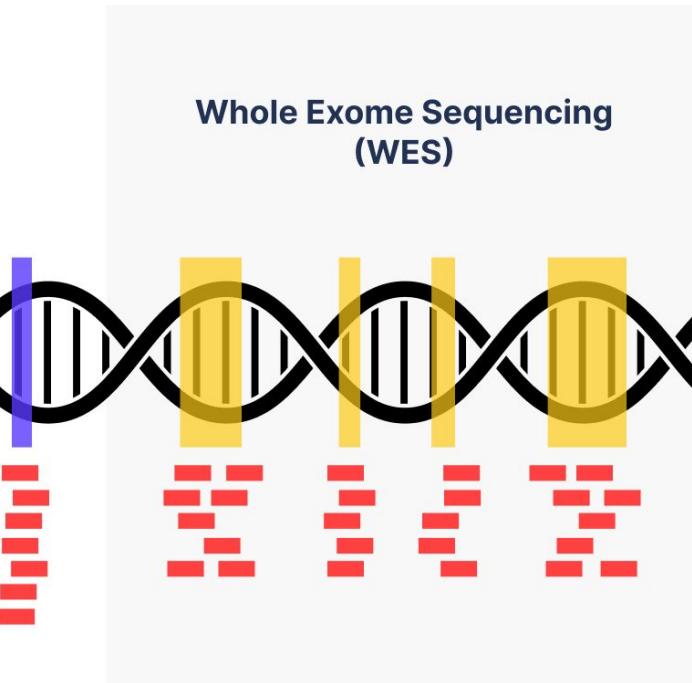
Case: A patient suspected of having hereditary breast cancer was recommended to undergo BRCA1/2 testing.

INTRODUCTION

Targeted Sequencing
(Panels)



Whole Exome Sequencing
(WES)

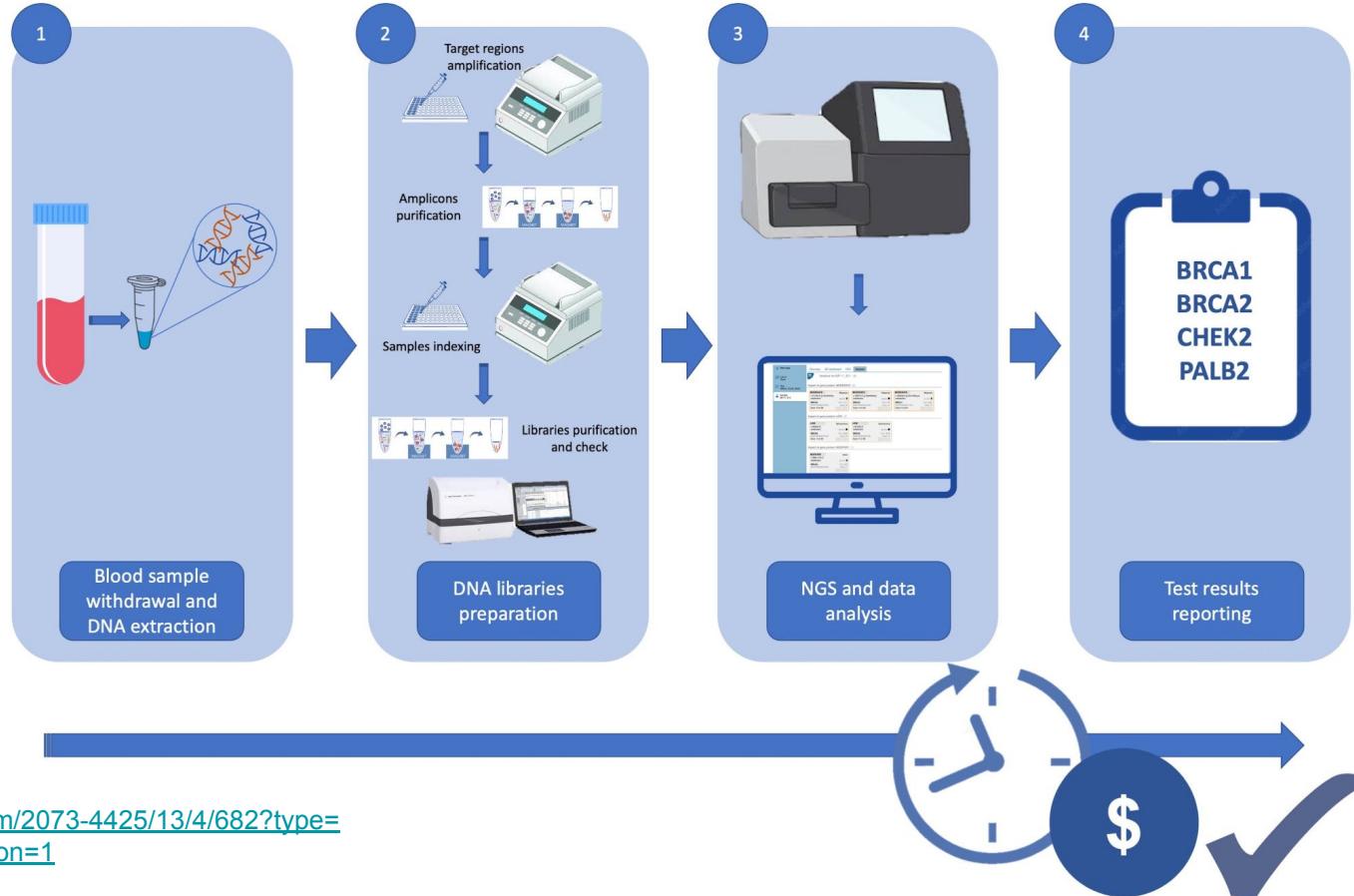


Whole Genome Sequencing
(WGS)



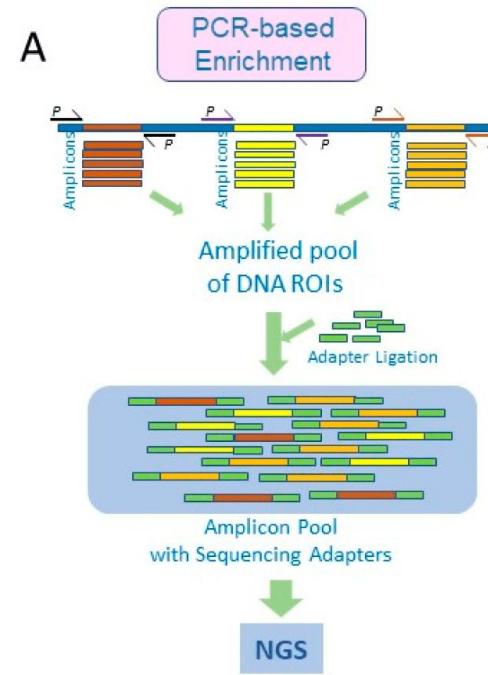
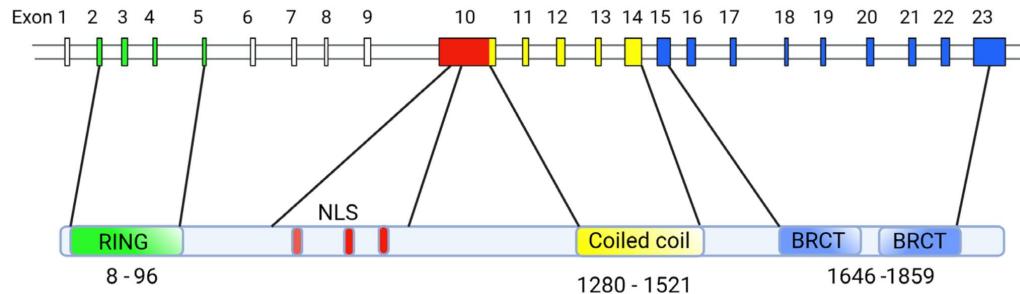
INTRODUCTION

Target sequencing workflow



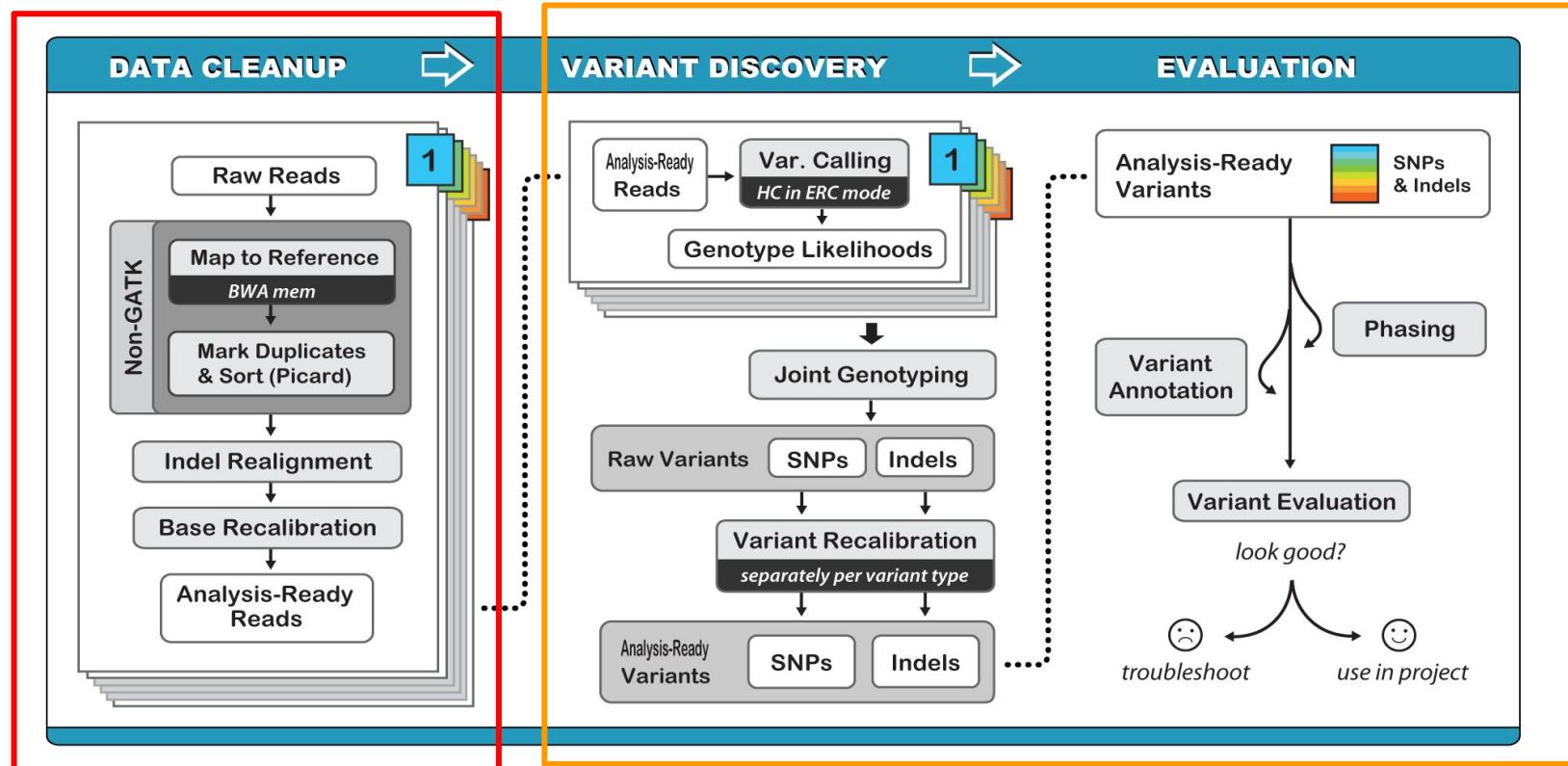
INTRODUCTION

BRCA1 Domain structure



<https://www.mdpi.com/2075-4418/12/7/1539>
<https://www.mdpi.com/1422-0067/24/5/4982>

DNA sequencing Workflow



Upstream Analysis

Downstream Analysis

INTRODUCTION

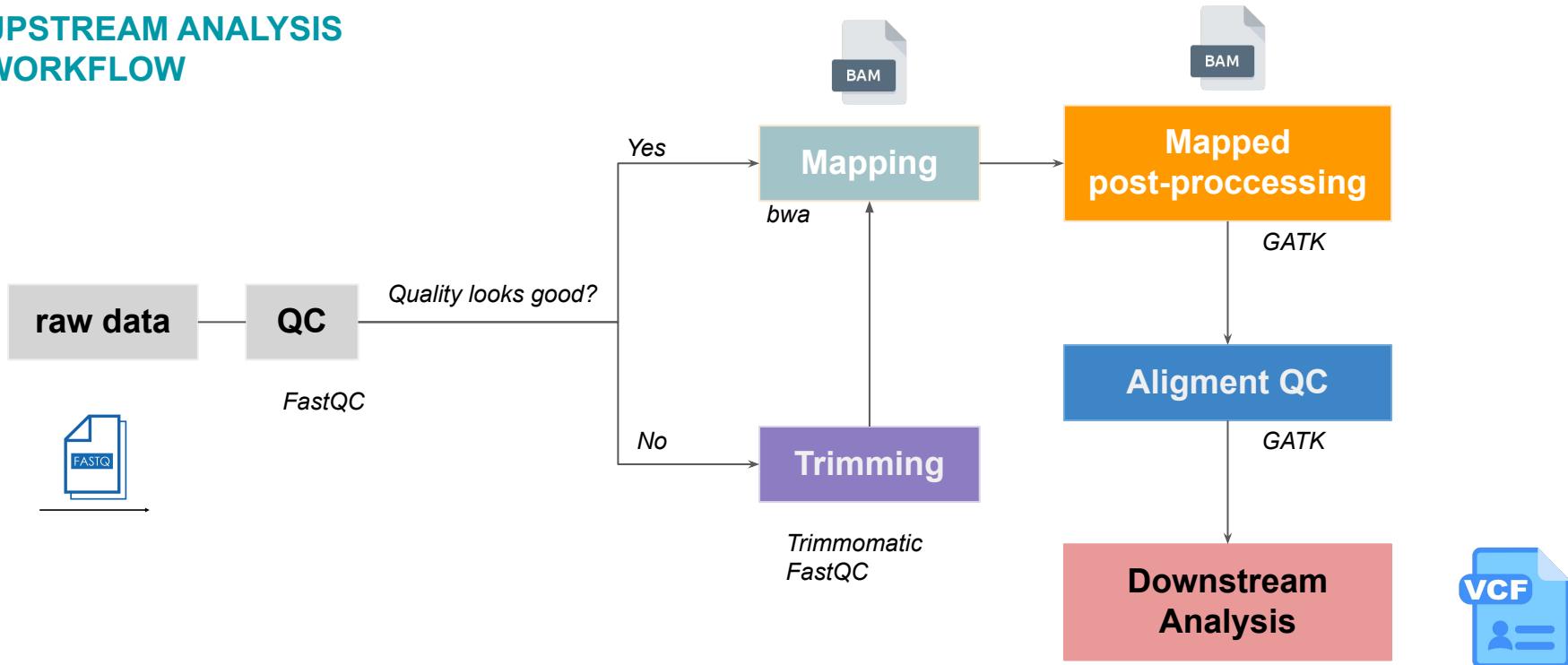
CONCEPT



Why we need to pre-process the NGS data before further analysis?

INTRODUCTION

UPSTREAM ANALYSIS WORKFLOW



RAW DATA PRE-PROCESSING

RAW DATA PROCESSING

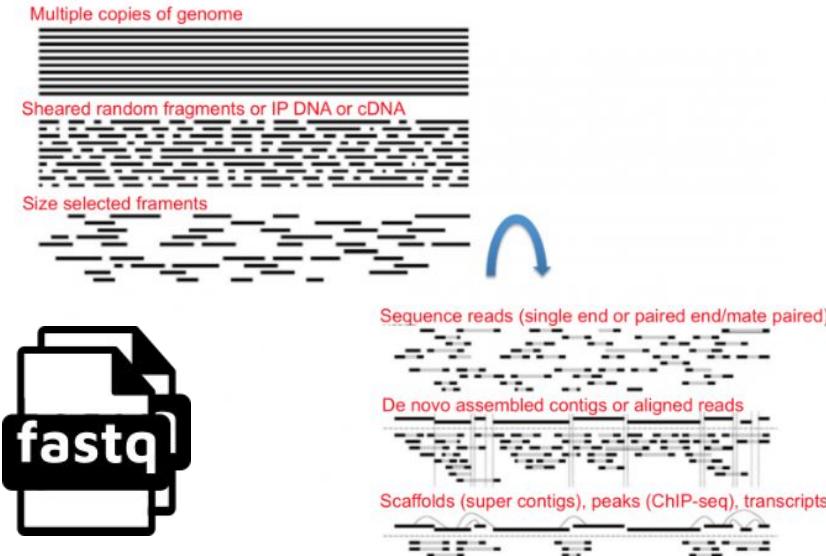
- ★ *What is raw data?*
- ★ *What is a fastq file (Illumina)?*
- ★ *Does your data look good? How to check it?*
- ★ *How to keep the good and eliminate the bad quality?*



RAW DATA PROCESSING

★ What is raw data?

The reads (sequences)

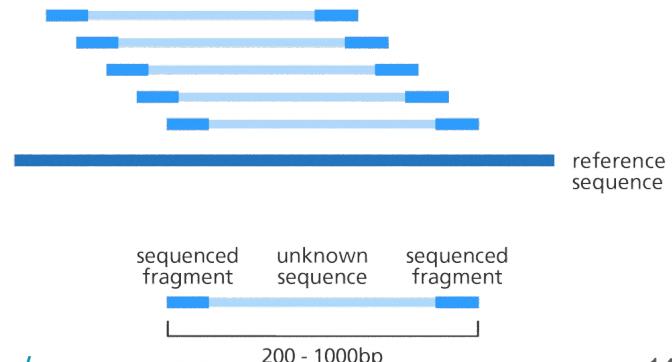


Short reads

Single-end reads



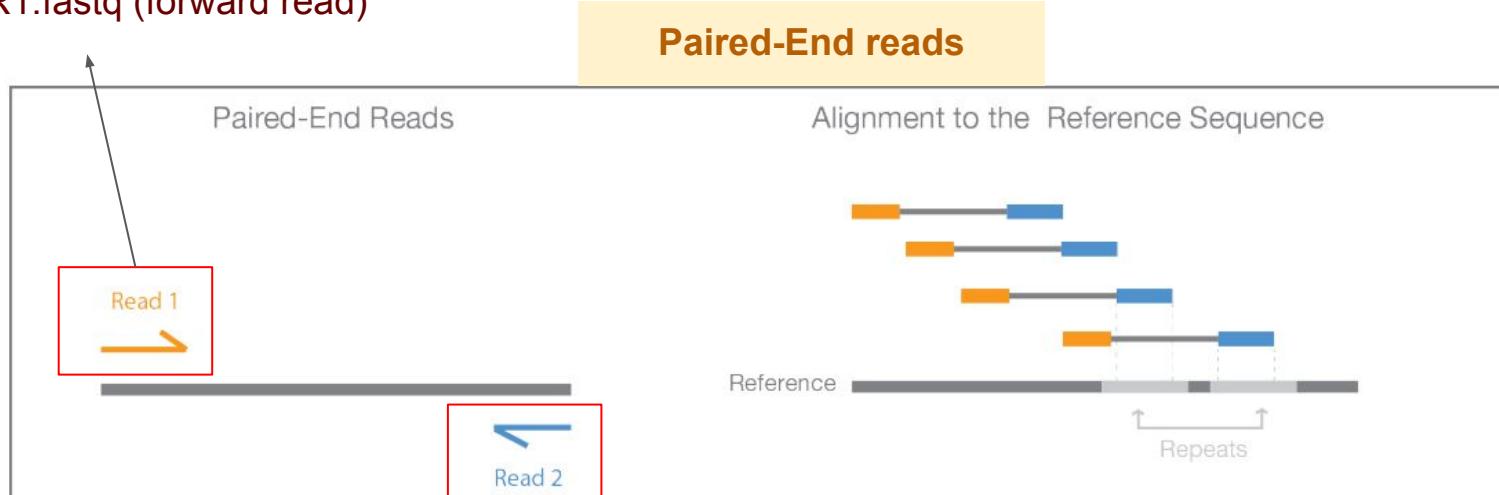
Paired-end reads



RAW DATA PROCESSING

★ What is raw data?

R1.fastq (forward read)



R2.fastq (reverse read)



RAW DATA PROCESSING

INTRODUCE TO FASTQ FILE

The diagram illustrates the structure of a FASTQ file. It shows a sequence of lines starting with an '@' symbol, followed by a title, the sequence itself, a '+', and then quality scores. Four boxes with arrows point to specific parts:

- Label**: Points to the first line starting with '@FORJUSP02AJWD1'.
- Sequence**: Points to the second line 'CCGTCAATTCACTTAAAGTTAACCTTGCAGCGTACTCCCCAGGCGGT'.
- Q scores (as ASCII chars)**: Points to the third line '+AAAAAAA:::99@:::?:?@:@::FFAAAAACCAA:::BB@@?A?'.
- Base=Q**: Points to the bottom line 'Base=T, Q='!'=25'.

File Format

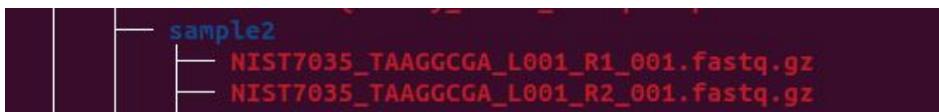
★ *What is a fastq file (Illumina)?*

```
@<title and optional description>
<sequence line>
+<optional repeat of title line>
<quality line>
```

RAW DATA PROCESSING

INTRODUCE TO FASTQ FILE

Illumina FASTQ file naming scheme



NIST7035_TAAGGCGA_L001_R1_001.fastq.gz

- sample_name: NIST7035
- barcode_sequence: TAAGGCGA
- lane: L001
- read_number: R1
- set_number: 001

> *What is the meaning of fastq file's name?*

RAW DATA PROCESSING

INTRODUCE TO FASTQ FILE

Illumina FASTQ read naming scheme (header of each read)

The header Begin with an '@'



fastq header format (version > 1.8)

Sequence Header +Sequence ID
a b c d e f g h i j k
@HWI-ST486:166:C06K9ACXX:7:1101:1443:1995 1:N:0:ACAGTG

- a. unique instrument name**
 - b. run id
 - c. flowcell id
 - d. flowcell lane
 - e. tile number within the flowcell lane
 - f. x-coordinate of the cluster within the tile
 - g. y-coordinate of the cluster within the tile
 - h. the member of a pair, 1 or 2 (paired-end or mate-pair reads only)**
 - i. Y if the read fails filter (read is bad), N otherwise
 - j. 0 when no control bits are on
 - k. index sequence

> What is the meaning of the reads' name?

RAW DATA PROCESSING

INTRODUCE TO FASTQ FILE

Base Quality Score (Q-score)

Table 5.2. Base Quality and ASCII Encoding.

ASCII character	Decimal value	Phred score
!	33	0
"	34	1
#	35	2
\$	36	3
:	:	:
A	65	22
B	66	23
:	:	:
x	120	87
y	121	88
z	122	89
{	123	90
	124	91
}	125	92
~	126	93



;B<97><A89>;<9?>?>9?<9=66;<<6@A@B?7<@<99@7<8:6?=66;@:6<;666778

“Measure of the confidence level in the accuracy of each nucleotide base call.”

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

Q30 is considered a benchmark for quality in next-generation sequencing.

RAW DATA PROCESSING

Classwork 1

Interpret these headers

1 @HWI-D00107:50:H6BP8ACWV:5:2204:10131:51624 2:N:0:AGGCAGAA

2 @Machine42:1:FC7:7:19:4229:1044 1:N:0:TTAGGC

- What is the ID of the instrument?
- What is the ID of the flowcell?
- Which lane is this read in?
- What tile is this read in?
- Forward or Reverse read?

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> FASTQC Summary

FastQC Report

Summary

- [Basic Statistics](#)
- [Per base sequence quality](#)
- [Per tile sequence quality](#)
- [Per sequence quality scores](#)
- [Per base sequence content](#)
- [Per sequence GC content](#)
- [Per base N content](#)
- [Sequence Length Distribution](#)
- [Sequence Duplication Levels](#)
- [Overrepresented sequences](#)
- [Adapter Content](#)

“FASTQC is a useful tool to check sequences quality.”

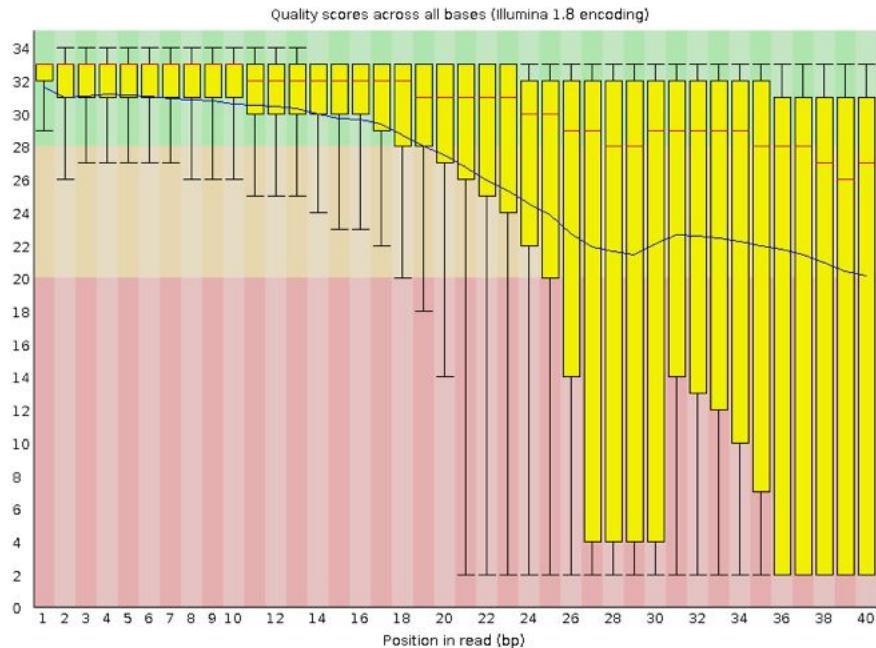
Basic Statistics

Measure	Value
Filename	NIST7035_TAAGGCAG_L001_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	20203002
Total Bases	2 Gbp
Sequences flagged as poor quality	0
Sequence length	101
%GC	49

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

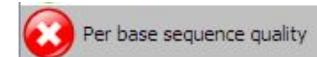
> Per base sequence quality



This module evaluates the quality at each base for all reads.

- Box-plot: Yellow
- Median: Red line
- Mean: Blue line

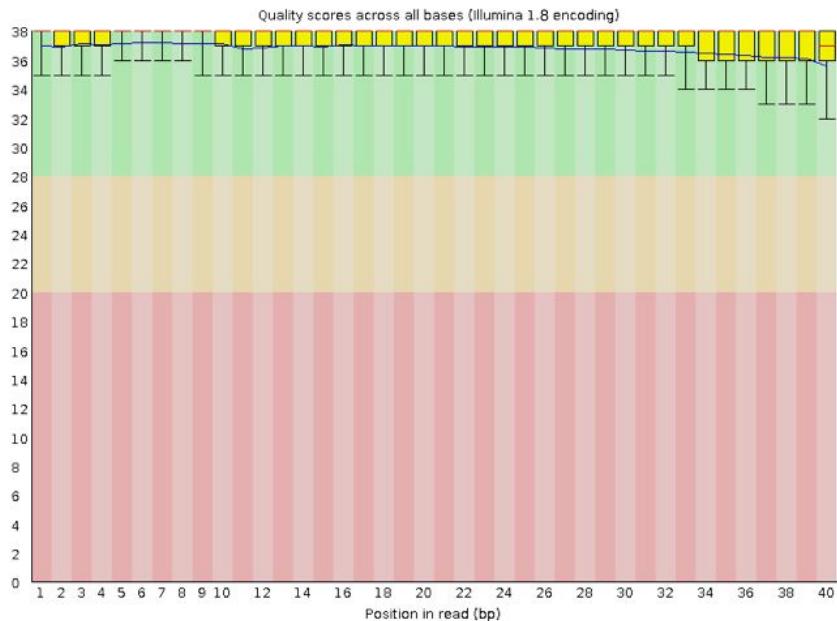
< Example of a bad quality score reads (40 bp)



RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per base sequence quality



Good quality score (Q30)

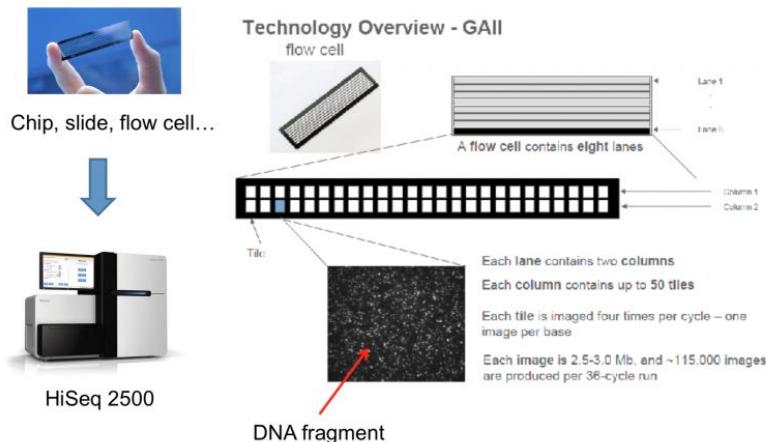


Per base sequence quality

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per tile sequence quality



What is a tile?

@HWI-ST486:166:C06K9ACXX:7:1101:1443:1995 1:N:0:ACAGTG

a. unique instrument name

b. run id

c. flowcell id

d. flowcell lane

e. tile number within the flowcell lane

f. x-coordinate of the cluster within the tile

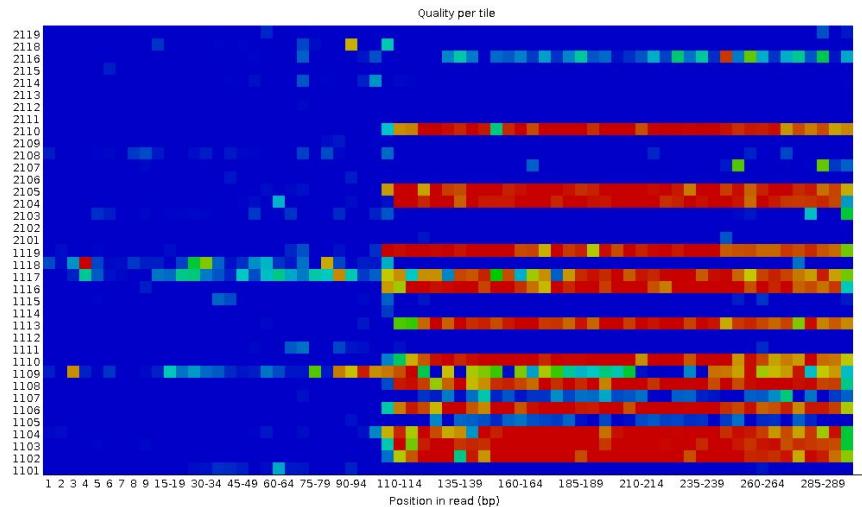
g. y-coordinate of the cluster within the tile

Information about tile stored in header of fastq

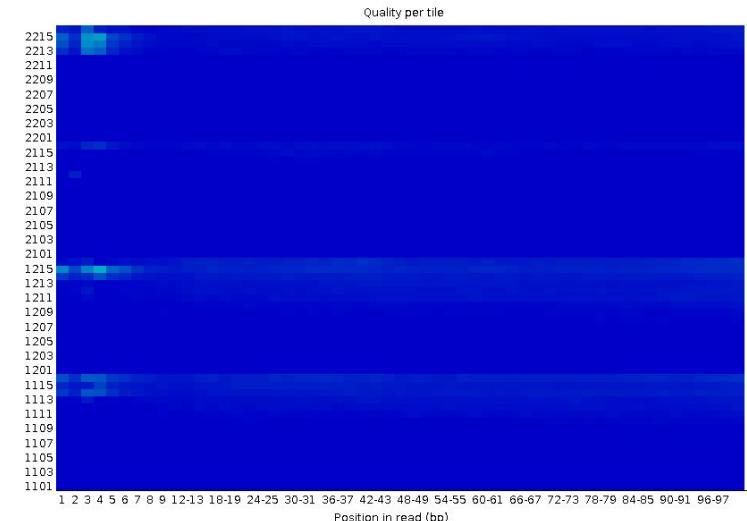
RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per tile sequence quality



Sequencing errors: bubbles, smudges, or dirt.



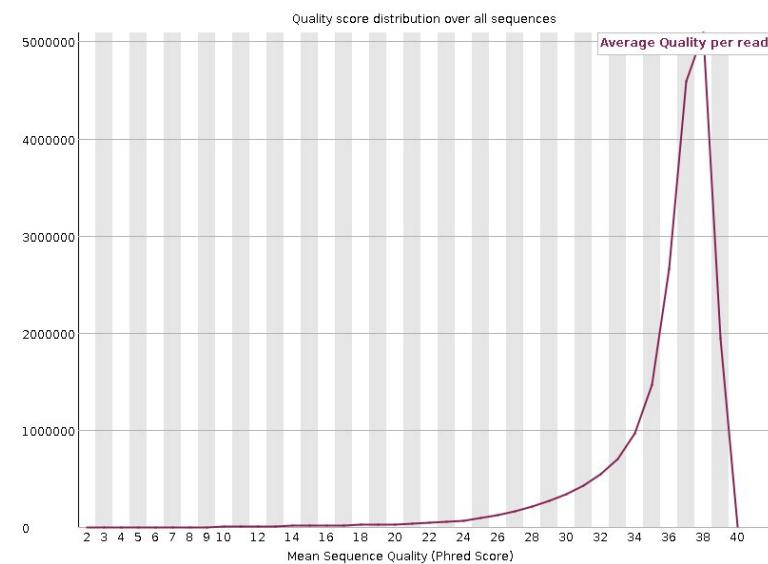
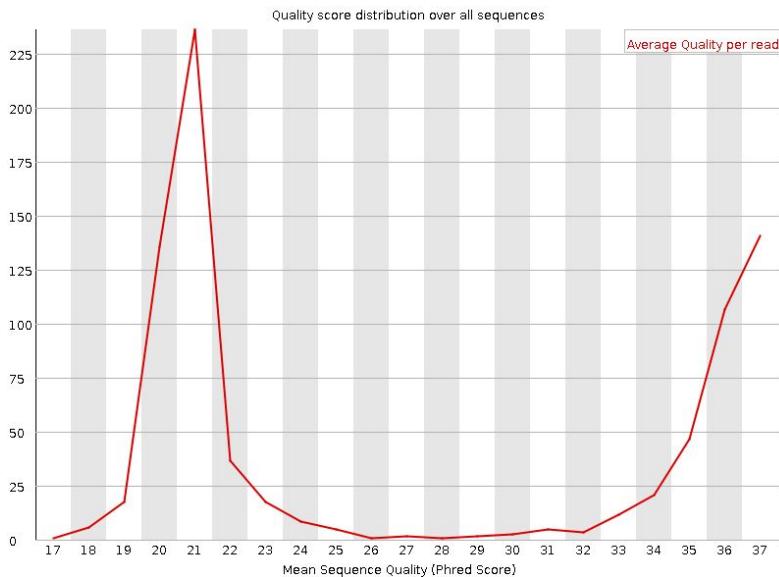
→ Cannot be fixed with bioinformatics

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per sequence quality scores

“The average quality score over the full length of all reads.”

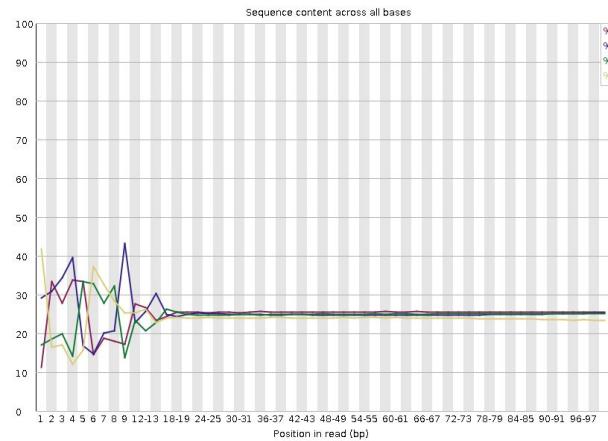
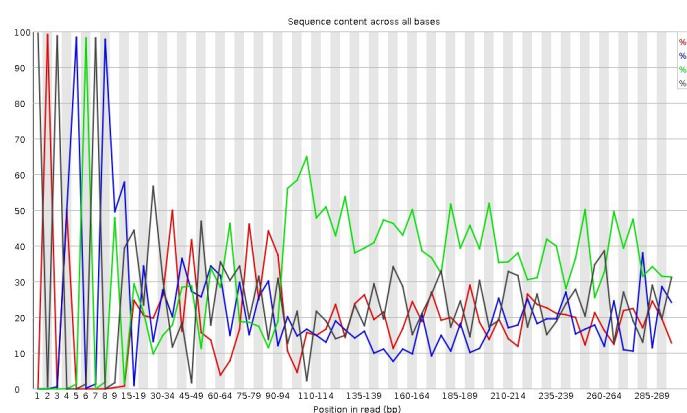


RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per Base Sequence Content

“Per Base Sequence Content” plots the percentage of each of the four nucleotides (T, C, A, G) at each position across all reads in the input sequence file.



Biased fragmentation (first 12 bp)

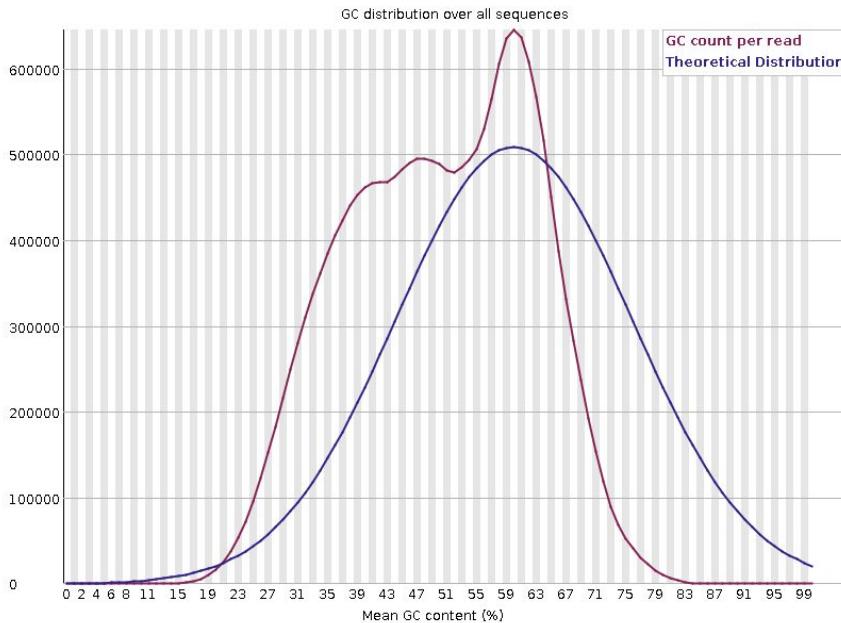
- Parallel
- $\%A = \%T$
- $\%G = \%C$



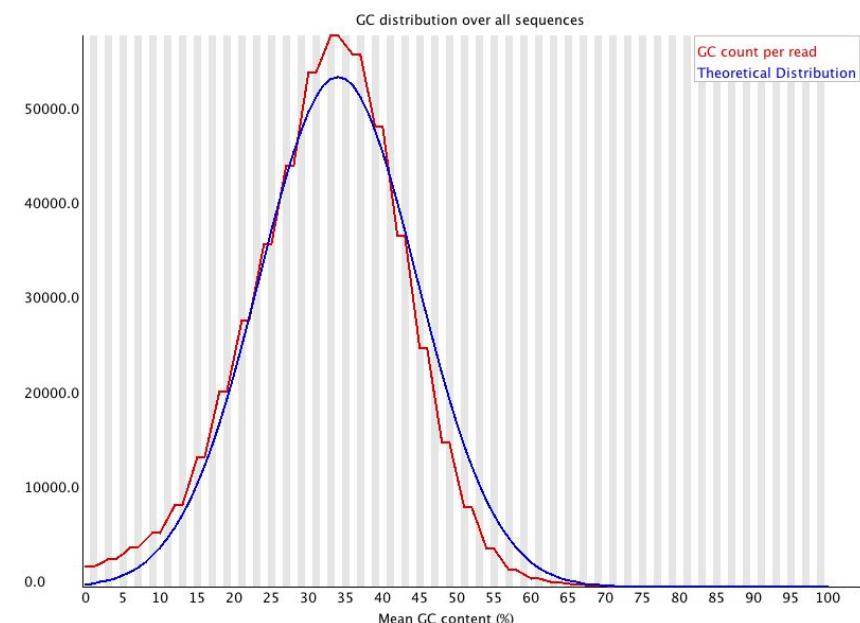
RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Per sequence GC content



Deviations from this theoretical distribution often implies contamination of some kind (adapter/primer dimers, multiple species in the run)

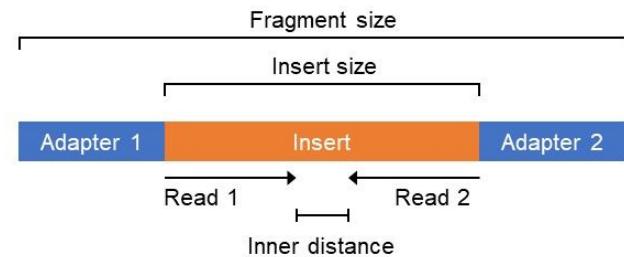
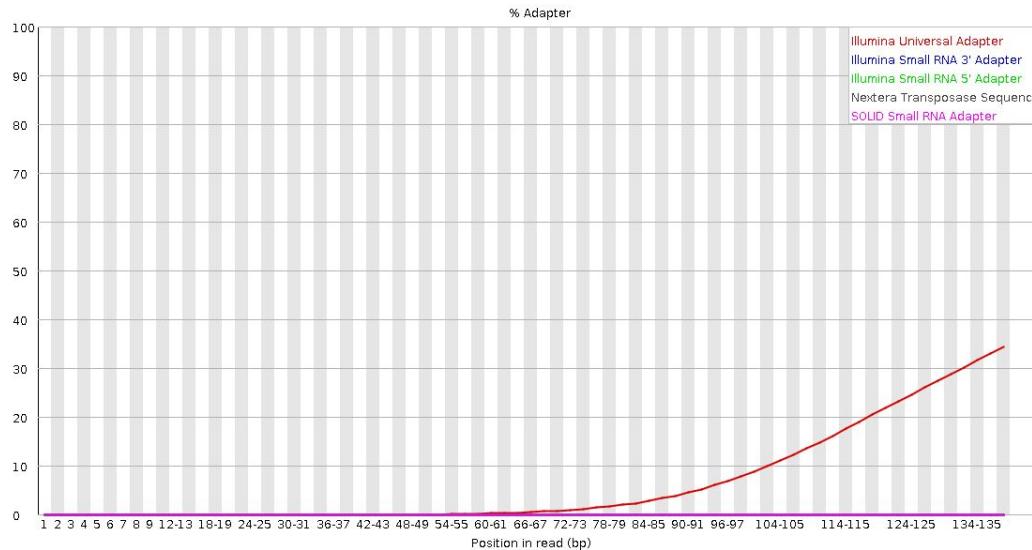


RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Adapter Content

Adapter Content



The Adapters need to be removed

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Overrepresented sequences

✖ Overrepresented sequences

Sequence	Count	Percentage	Possible Source
GTCGGTAAAACCGTCCCAGCCACCGCGGTACATCGATTAAACCAAGCTA	14869	13.721346572662508	No Hit
GTCGGTAAAACCGTCCCAGCAGGAAACTGGGATTAGATAACCCACTATGCTG	10094	9.314901627846886	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTAATACAGAGGTCTAACCGGT	9386	8.66154811561035	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTAATACAGAGGTCCAAAGCGT	8815	8.134620353622974	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTTCTCGAATCCGGTCCAAATAG	4442	4.0991473182975895	No Hit
GTCGGTAAAACCGTCCCAGCAGCCACCGCGGTACACGATTAAACCAAGTCA	3578	3.3018345576021555	No Hit
GTCGGTAAAACCGTCCCAGCCCCACGAGACCAAACGGGATTAGATAACCC	1941	1.7911852644789783	No Hit
GTCGGTAAAACCGTCCCAGCACGATTCCGGAGGGCGTTGCAATATTGGC	1040	0.9597283230593187	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTAACAGAGGTCCCGAGCGT	692	0.6385884611125466	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTAATACGGAGGATCCGAGCGT	669	0.6173637001218116	No Hit
GTCGGTAAAACCGTCCCAGCAGCCGGTAATACGGGATT	623	0.5749141781403417	No Hit
GTCGGTAAACACTCGTCCCAGCAGCCGGTAATACGGATTAAACCAAGCTA	575	0.5306190247683733	No Hit
GTCGGTAAAACCGTCCCAGCAGAGACAGCAAACGGGATTAGATAACCC	570	0.5260049462921266	No Hit

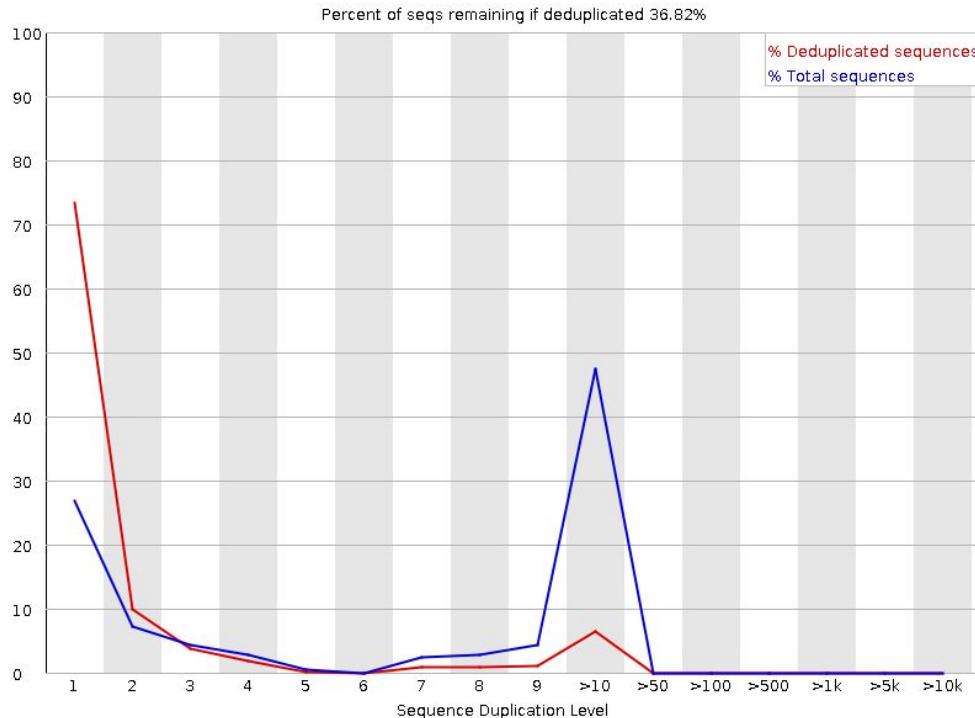
- Contaminated?
- Adapter?
- RNA transcripts?

FastQC lists all of the sequence which make up more than 0.1% of the total.

RAW DATA PROCESSING

SEQUENCE QUALITY CONTROL (FASTQC)

> Sequence duplication levels



2 types of Duplication Error:

- PCR duplication
- Optical duplication

→ Map and detect which type.



RAW DATA PROCESSING

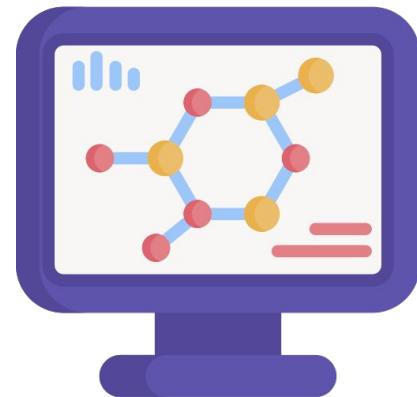
★ *How to deal with these problems?*

Things that we (bioinformaticians) can do:

- Filter and remove bad sequences
- Remove duplicated sequences
- Remove adapter
- Crop bad quality bases at the head/tail of sequences.

Things we can't:

- Completely remove contaminants
- Control GC contents.
(Control previous steps: library prep, sequencing)



RAW DATA PROCESSING

READ TRIMMING & FILTERING

usadellab/
Trimmomatic

2
Contributors

25
Issues

131
Stars

56
Forks



This program does adaptive quality trimming, head and tail crop, and adaptor removal.

Check QC → Trim → Check QC again.



Trimming:

- Quality trimming
- Adapter trimming.

RAW DATA PROCESSING

Quality Trimming & Adapter removal

Syntax:

```
# trim S11 exec  
trimmomatic PE \  
-phred33 \  
-threads 56 \  
-trimlog $p_trim/S11.log \  
-summary $p_trim/S11_sum.log \  
$p_raw/S11_L001_R1_001.fastq.gz \  
$p_raw/S11_L001_R2_001.fastq.gz \  
$p_trim/S11_L001_R1_trimmed_paired.fastq.gz \  
$p_trim/S11_L001_R1_trimmed_unpaired.fastq.gz \  
$p_trim/S11_L001_R2_trimmed_paired.fastq.gz \  
$p_trim/S11_L001_R2_trimmed_unpaired.fastq.gz \  
ILLUMINACLIP:/mnt/rdisk/duydao/PROJECT/DNASEQ_BRCA12/ref/adapters/TruSeq3-PE-2.fa:2:30:10:8:3:t  
rue \  
HEADCROP:5 \  
CROP:140 \  
LEADING:3 \  
TRAILING:10 \  
SLIDINGWINDOW:4:28 \  
MINLEN:36
```

NexteraPE-PE.fa
TruSeq2-PE.fa
TruSeq2-SE.fa
TruSeq3-PE-2.fa
TruSeq3-PE.fa
TruSeq3-SE.fa

RAW DATA PROCESSING

TRIMMOMATIC WORKFLOW

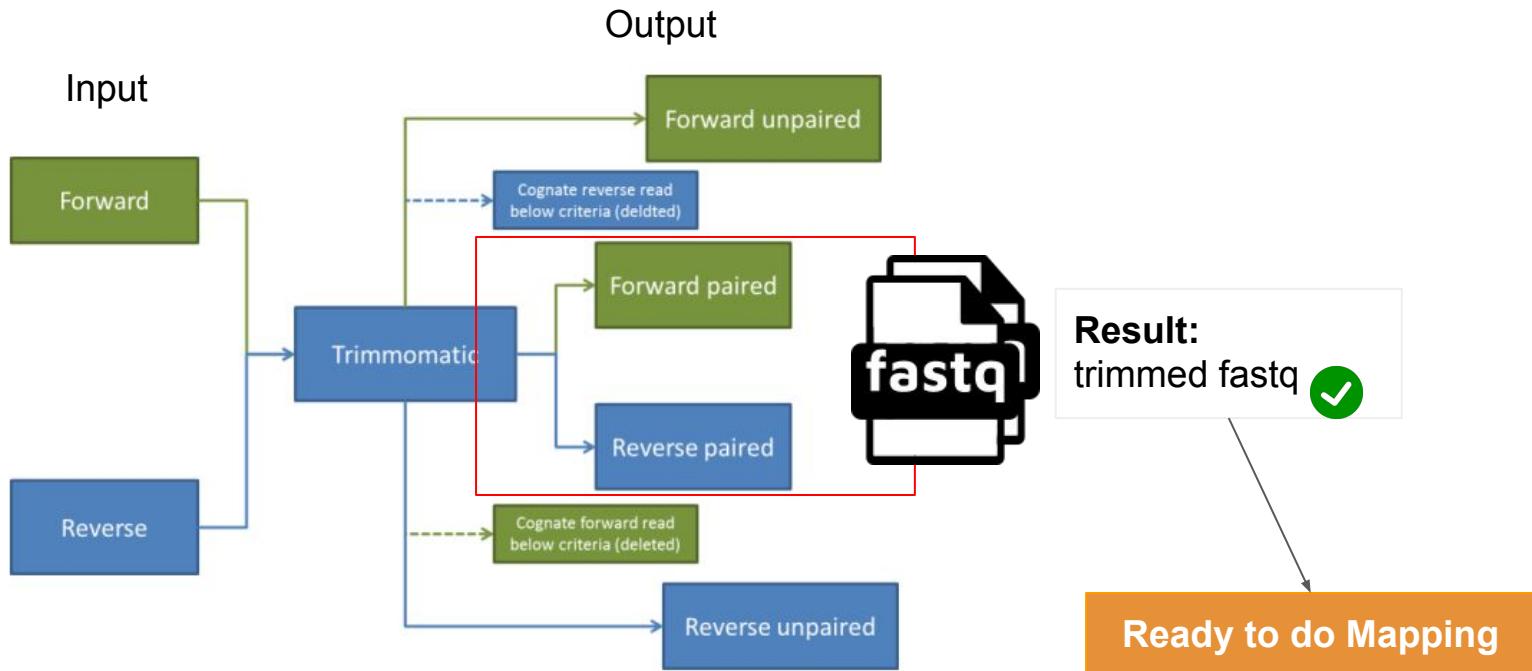
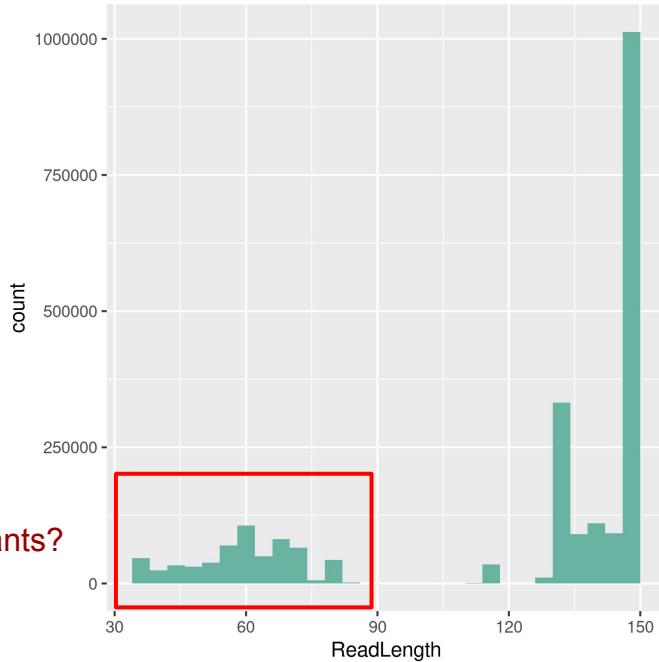


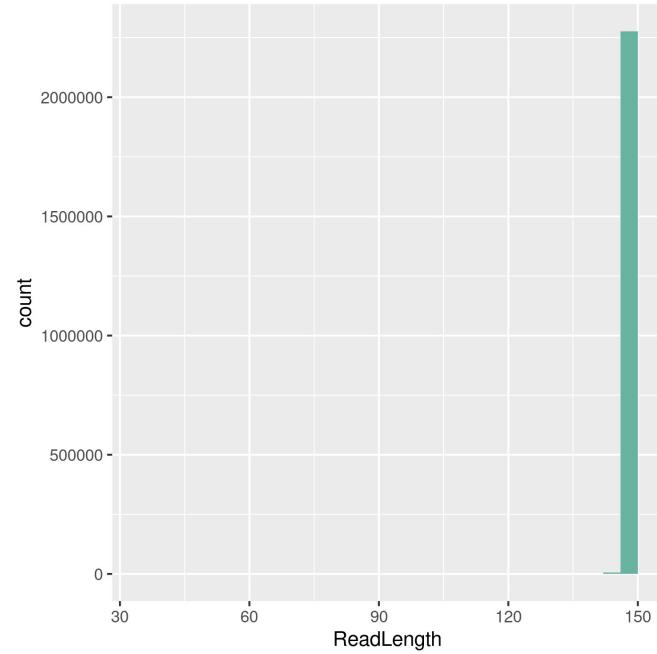
Figure 1: Flow of reads in Trimmomatic Paired End mode

RAW DATA PROCESSING

Before trim



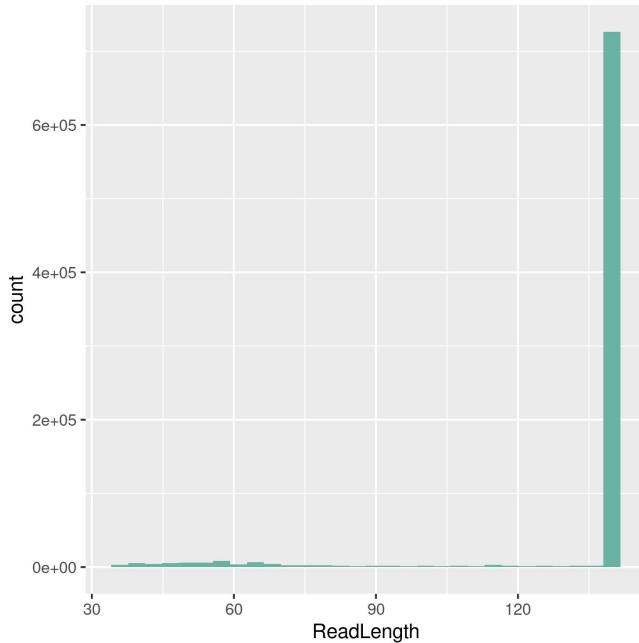
Read 1



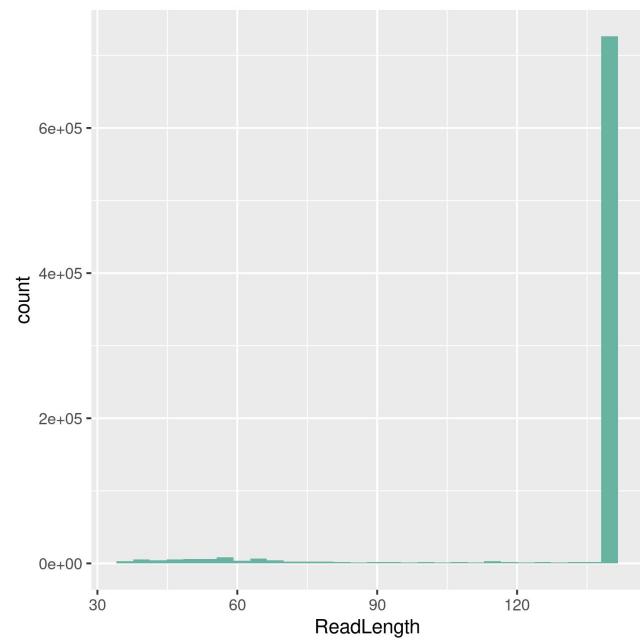
Read 2

RAW DATA PROCESSING

After trim



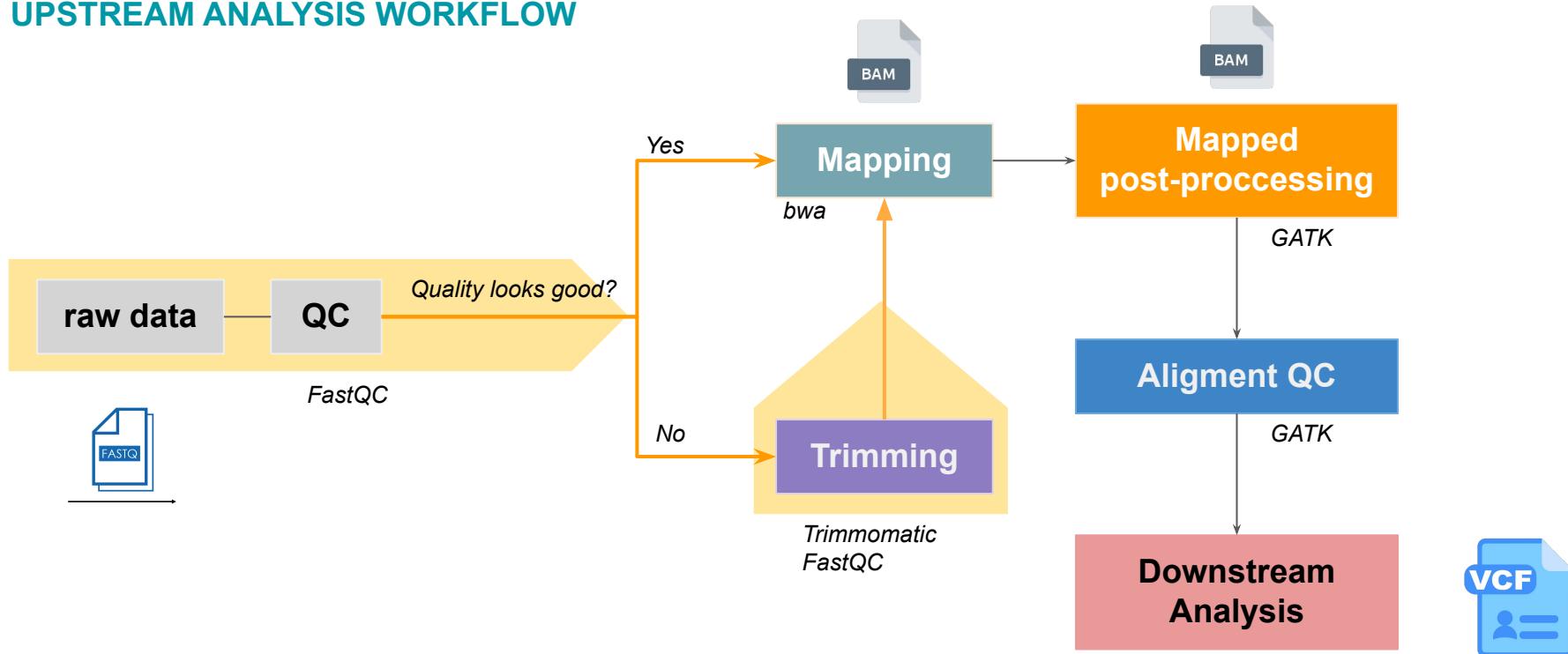
Read 1



Read 2

RAW DATA PROCESSING

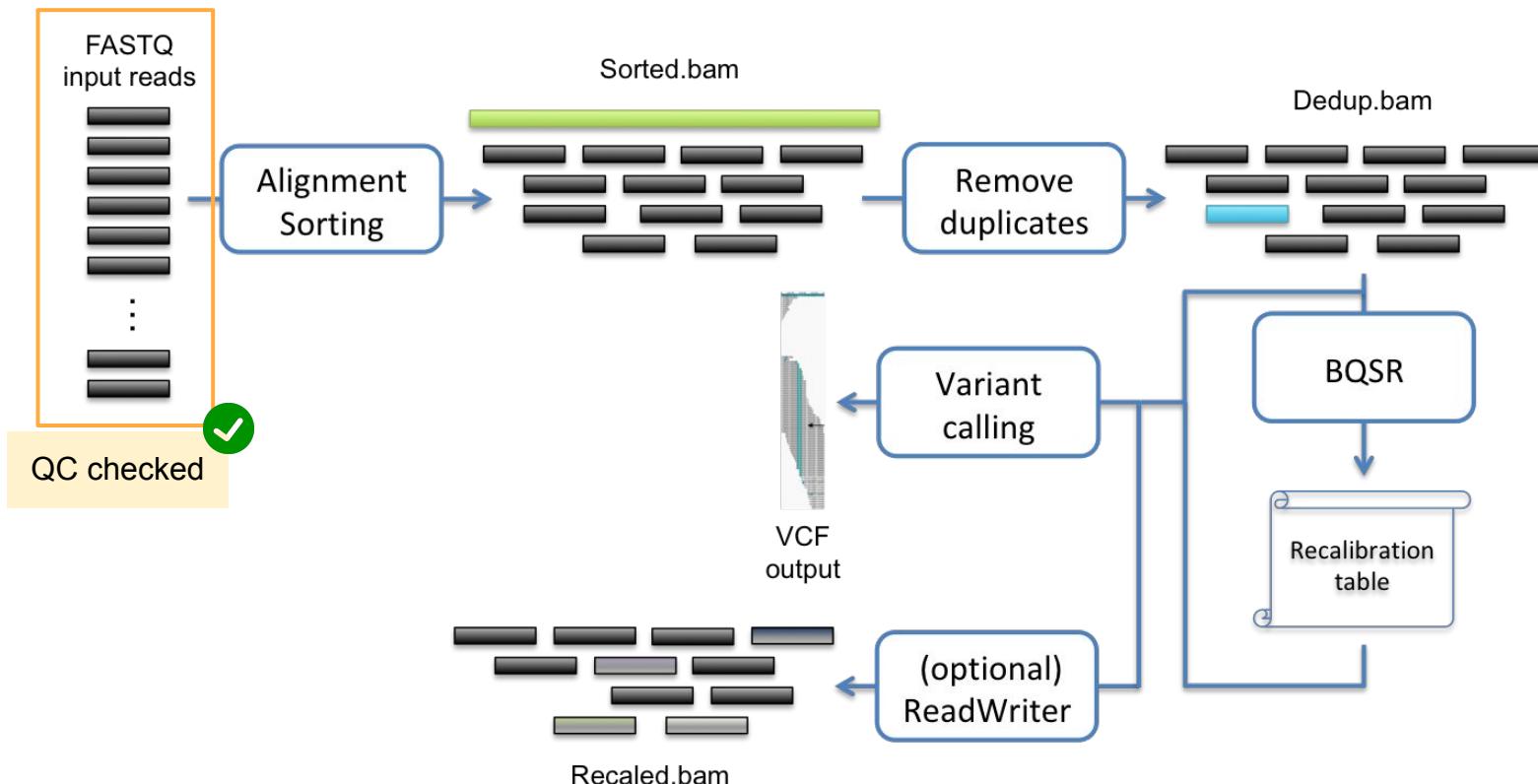
UPSTREAM ANALYSIS WORKFLOW



ALIGNMENT

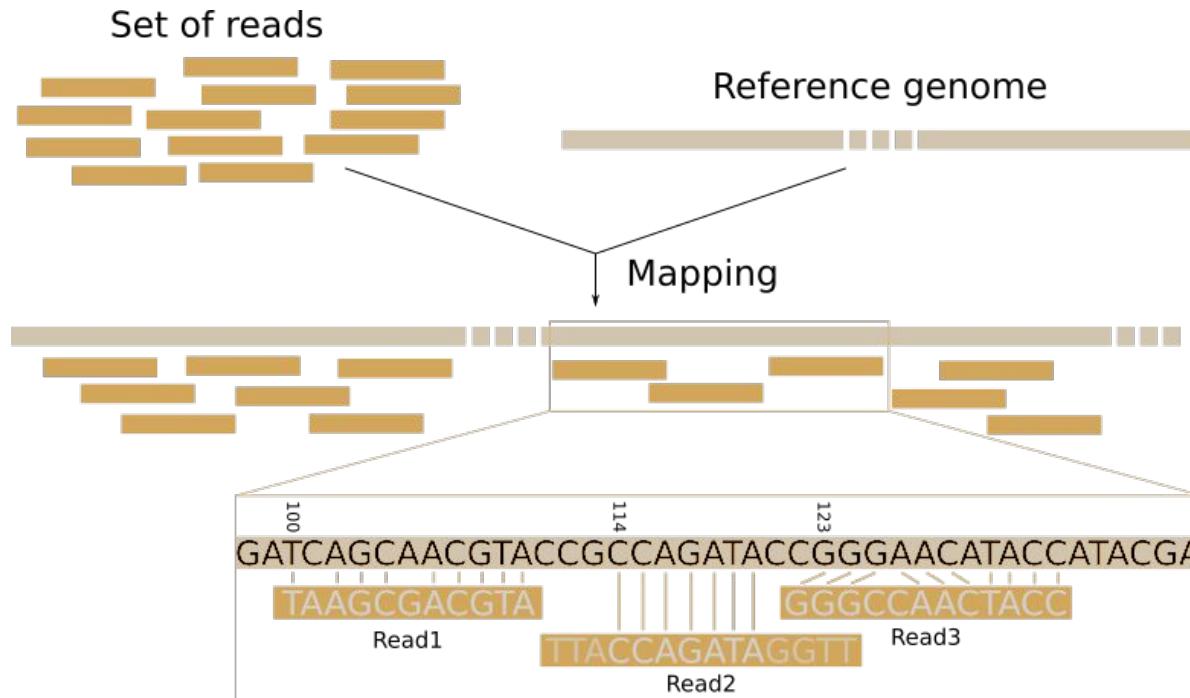
ALIGNMENT & MAPPED POST-PROCESSING

ALIGNMENT & POST-PROCESSING WORKFLOW



ALIGNMENT / MAPPING

ALIGNMENT

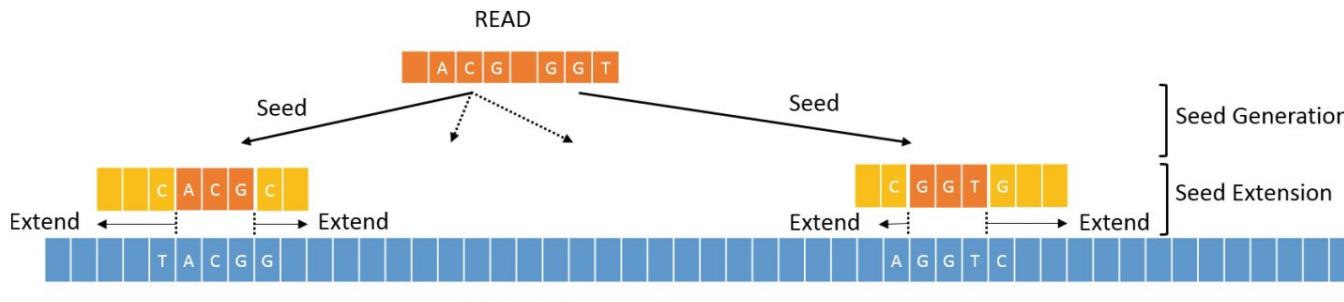


ALIGNMENT / MAPPING

ALIGNMENT

BWA: Burrows-Wheeler Alignment tool

- The most widely used aligners for Illumina data
- Based on BWT Algorithms
- **BWA-MEM** was developed for reads of ≥ 70 bases, for shorter reads it is advisable to use the standard BWA algorithm. Both are provided by the same tool.



BWA mem algorithm

ALIGNMENT / MAPPING

ALIGNMENT

Hands-on: Using BWA to Align NIST7035 data (sample2). (This may take a while)

First: Index the reference genome.

Syntax :

```
bwa index -a bwtsw hg38.fa
```

When finished, we will have the following files like this:

```
ref_genome
  |- hg19.chr5_12_17.fa
  |- hg38
    |- hs38DH.fa hs38DH.fa
    |- hs38DH.fa.amb
    |- hs38DH.fa.ann
    |- hs38DH.fa.bwt
    |- hs38DH.fa.pac
    |- hs38DH.fa.sa
```



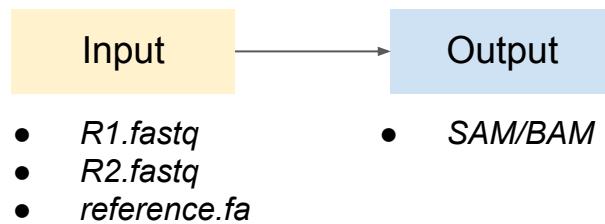
ALIGNMENT / MAPPING

ALIGNMENT

Hands-on: Using BWA to align S11 data.

Syntax:

```
bwa mem -t 4 \
-R '@RG\tID:rg1\tSM:S11\tPL:illumina\tLB:lib1\tPU:M07220:1:TGCAGCTA+CCTAGAGT' \
$p_ref/hg38.fa \
$p_trim/S11_L001_R1_trimmed_paired.fastq.gz \
$p_trim/S11_L001_R2_trimmed_paired.fastq.gz > $p_align/S11_aln.sam
```



ALIGNMENT / MAPPING

ALIGNMENT

Introduce to SAM/BAM file > SAM format

Sequence Alignment/Map (SAM) format is a tab-delimited text format that aims to be a universal format for storing alignments of NGS reads to a reference genome.

Header (begin with '@') + Alignment section.

```
@SQ SN:chrUn_KI270744v1 LN:168472
@SQ SN:chrUn_KI270745v1 LN:41891
@SQ SN:chrUn_KI270746v1 LN:66486
@SQ SN:chrUn_KI270747v1 LN:198735
@SQ SN:chrUn_KI270748v1 LN:93321
@SQ SN:chrUn_KI270749v1 LN:158759
@SQ SN:chrUn_KI270750v1 LN:148850
@SQ SN:chrUn_KI270751v1 LN:150742
@SQ SN:chrUn_KI270752v1 LN:27745
@SQ SN:chrUn_KI270753v1 LN:62944
@SQ SN:chrUn_KI270754v1 LN:40191
@SQ SN:chrUn_KI270755v1 LN:36723
@SQ SN:chrUn_KI270756v1 LN:79590
@SQ SN:chrUn_KI270757v1 LN:71251
@SQ SN:chrX LN:156040895
@SQ SN:chrX_KI270880v1 alt LN:284869
@SQ SN:chrX_KI270881v1_alt LN:144206
@SQ SN:chrX_KI270913v1_alt LN:274009
@SQ SN:chrY LN:57227415
@SQ SN:chrY_KI270740v1_random LN:37240
@RG ID:rg1 SM:NA12878 PL:illumina LB:lib1 PU:H7AP8ADXX:1:TAAGGCGA
@PG ID:bwa PN:bwa VN:0.7.17-r1188 CL:bwa mem -t 8 -R @RG\tID:rg1\tSM:NA12878\tPL:illumina\tLB:lib1\tPU:H7AP8ADXX:1:TAAGGCGA /DA
@PG ID:samtools PN:samtools PP:bwa VN:1.13 CL:samtools view -Sb NIST7035_aln.sam
@PG ID:samtools.1 PN:samtools PP:samtools VN:1.13 CL:samtools view -h NIST7035_aln.bam
HWI-D00119:50:H7AP8ADXX:1:1101:1322:2086 113 chr17 74243430 60 86M = 74243430 0 GCCTC
HWI-D00119:50:H7AP8ADXX:1:1101:1322:2086 177 chr17 74243430 60 86M = 74243430 0 GCCTC
HWI-D00119:50:H7AP8ADXX:1:1101:1322:2086 175 chr17 74243430 60 86M = 74243430 0 GCCTC
```

Header of a SAM file



ALIGNMENT / MAPPING

ALIGNMENT

SAM/BAM file > SAM format

Alignment sections have 11 mandatory fields, as well as a variable number of optional fields.

QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ
HWI-D00119:50:H7APBADXX:1:1101:1322:2086	113	chr17	74243430	60	86M =	74243430	0	GCCTGGACTGGCAAGGGCTTCAAGTGTCCCTGGCTCGTAGTATCCTACCCGTTGGACCTGGACAAAGG	
HWI-D00119:50:H7APBADXX:1:1101:1322:2086	177	chr17	74243430	60	86M =	74243430	0	GCCTGGACTGGCAAGGGCTTCAAGTGTCCCTGGCTCGTAGTATCCTACCCGTTGGACCTGGACAAAGG	
HWI-D00119:50:H7APBADXX:1:1101:1294:2087	65	chr1	49798791	60	73M =	49798791	0	CCAACTCTGGAAGTTTTTTTTTTAATAAATTTCAGCATTTCTGTAATTAAAGTGTAAATCACCATG	
HWI-D00119:50:H7APBADXX:1:1101:1294:2087	129	chr1	49798791	60	73M =	49798791	0	CCAACTCTGGAAGTTTTTTTTTTAATAAATTTCAGCATTTCTGTAATTAAAGTGTAAATCACCATG	
HWI-D00119:50:H7APBADXX:1:1101:1406:2159	65	chr18	9733113	60	86M =	9733113	0	CCATGCTATAAACCTTTATTTTCCACTTGGTTGAGTGGGATCCTTCATTGGAGCTAATCCAGAAAATGCTAATTCTAGTGG HGI	
HWI-D00119:50:H7APBADXX:1:1101:1406:2159	129	chr18	9733113	60	86M =	9733113	0	CCATGCTATAAACCTTTATTTTCCACTTGGTTGAGTGGGATCCTTCATTGGAGCTAATCCAGAAAATGCTAATTCTAGTGG HGI	
HWI-D00119:50:H7APBADXX:1:1101:1303:2183	65	chr6	132570953	60	86M =	132570953	0	TTATTATGATAATTCTGTATGTAACATATTCTGTGCTGAGCACGACGGGAAAAGATAAGAAAATACTGGTA	
HWI-D00119:50:H7APBADXX:1:1101:1303:2183	129	chr6	132570953	60	86M =	132570953	0	TTATTATGATAATTCTGTATGTAACATATTCTGTGCTGAGCACGACGGGAAAAGATAAGAAAATACTGGTA	
HWI-D00119:50:H7APBADXX:1:1101:1413:2244	81	chr9	56204243	0	86M =	56195531	-8798	TTGCCCGCTCTAGCAGTTGGACTCTTCAATTCTGAGGACCTTCTATCTCTCCCATATCGCTGCTG	
HWI-D00119:50:H7APBADXX:1:1101:1413:2244	161	chr9	56195531	0	86M =	56204243	8798	GGGATGTAATTAGAGCAGGGATATTGGAGACAGTTAGAACCTCTCAGAAAATTGAAAGTCACAACTGCA	
HWI-D00119:50:H7APBADXX:1:1101:1372:2246	113	chr19	50402338	60	86M =	50402338	0	CGGCCCTACGAGGCCAACGTCGACTTTGAGATCCGGTACGGCTCTGCTACTTCCGGCTCTATCCCCAC	
HWI-D00119:50:H7APBADXX:1:1101:1372:2246	177	chr19	50402338	60	86M =	50402338	0	CGGCCCTACGAGGCCAACGTCGACTTTGAGATCCGGTACGGCTCTGCTACTTCCGGCTCTATCCCCAC	
HWI-D00119:50:H7APBADXX:1:1101:1613:2099	113	chr1	172749268	27	86M =	172749268	0	TGGTATTCTCGCTCACTTGGTAACCTACCGGGATGTTTACAAACCATATCGCAGCTTACCAACTGAAAGT	
HWI-D00119:50:H7APBADXX:1:1101:1613:2099	177	chr1	172749268	27	86M =	172749268	0	TGGTATTCTCGCTCACTTGGTAACCTACCGGGATGTTTACAAACCATATCGCAGCTTACCAACTGAAAGT	
HWI-D00119:50:H7APBADXX:1:1101:1644:2101	113	chr1	76700193	46	86M =	76700193	0	GGAACTGGCATCACTGAAGGGTTTCTGGAGCACAAAGTCATCGCAGATAACCTGAAAGACTGGACATGGAGCA	
HWI-D00119:50:H7APBADXX:1:1101:1644:2101	177	chr1	76700193	46	86M =	76700193	0	GGAACTGGCATCACTGAAGGGTTTCTGGAGCACAAAGTCATCGCAGATAACCTGAAAGACTGGACATGGAGCA	
HWI-D00119:50:H7APBADXX:1:1101:1657:2113	65	chr8	85661599	0	86M =	85654762	-6838	GCTGTG	
HWI-D00119:50:H7APBADXX:1:1101:1657:2113	129	chr8	85654762	0	86M =	85661599	6838	GCTGTG	
HWI-D00119:50:H7APBADXX:1:1101:1777:2104	113	chr19	18784989	60	86M =	18784989	0	GGAGCT	
HWI-D00119:50:H7APBADXX:1:1101:1777:2104	177	chr19	18784989	60	86M =	18784989	0	GGAGCT	
HWI-D00119:50:H7APBADXX:1:1101:1830:2136	65	chr1	227731993	60	86M =	227731993	0	CAGGG	
HWI-D00119:50:H7APBADXX:1:1101:1830:2136	129	chr1	227731993	60	86M =	227731993	0	CAGGG	
HWI-D00119:50:H7APBADXX:1:1101:2021:2071	65	chr17	18541691	0	86M =	18541691	0	ACCAA	
HWI-D00119:50:H7APBADXX:1:1101:2021:2071	129	chr17	18541691	0	86M =	18541691	0	ACCAA	
HWI-D00119:50:H7APBADXX:1:1101:2027:2162	65	chr2	39673045	60	86M =	39673045	0	GTGCT	
HWI-D00119:50:H7APBADXX:1:1101:2027:2162	129	chr2	39673045	60	86M =	39673045	0	GTGCT	
HWI-D00119:50:H7APBADXX:1:1101:2340:2087	65	chr2	232386840	60	86M =	232386840	0	GGCCC	
HWI-D00119:50:H7APBADXX:1:1101:2340:2087	129	chr2	232386840	60	86M =	232386840	0	GGCCC	
H	r4		9326689	0	86M =	9331435	4747	CGAGGGAGGGCCAGGAGATC	
H	r4		9326689	0	86M =	9331435	-4747	CGAGGGAGGGCCAGGAGATC	

Table 9.1. Mandatory fields of the SAM Format.

Col	Field	Description	Example
1	QNAME	Query template NAME	read_1
2	FLAG	Bitwise FLAG	0
3	RNAME	Reference sequence NAME	chrE
4	POS	Left-most mapping POSition (1-based)	11
5	MAPQ	MAPping Quality	37
6	CIGAR	CIGAR string	10M
7	RNEXT	Ref. name of the mate or NEXT read	*
8	PNEXT	Position of the mate or NEXT read	0
9	TLEN	Observed Template LENgth	0

GATCTCG GATCTCG CG BCC CG BCC

Alignment section

Table 9.1. Mandatory fields of the SAM Format

Alignment section

SAM format: 11 mandatory fields

Note: Each line in the alignment section of a SAM file comprises 11 mandatory fields.

ALIGNMENT / MAPPING

ALIGNMENT

SAM/BAM file > Flagstats

Flagstats

```
1629692 + 0 in total (QC-passed reads + QC-failed reads)
1629194 + 0 primary
0 + 0 secondary
498 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1618417 + 0 mapped (99.31% : N/A)
1617919 + 0 primary mapped (99.31% : N/A)
1629194 + 0 paired in sequencing
814597 + 0 read1
814597 + 0 read2
1603792 + 0 properly paired (98.44% : N/A)
1617768 + 0 with itself and mate mapped
151 + 0 singletons (0.01% : N/A)
108 + 0 with mate mapped to a different chr
53 + 0 with mate mapped to a different chr (mapQ>=5)
```

The flagstat function of SAMtools provides a summary of the number of records corresponding to each of the bit flags.

ALIGNMENT / MAPPING

ALIGNMENT

SAM/BAM file > Bit wise Flag

Picard
[Build Status](#)
A set of command line tools (in Java) for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF.

Decoding SAM flags

This utility makes it easy to identify what are the properties of a read based on its SAM flag value, or conversely, to find what the SAM Flag value would be for a given combination of properties.

To decode a given SAM flag value, just enter the number in the field below. The encoded properties will be listed under Summary below, to the right.

SAM Flag: [Explain](#)

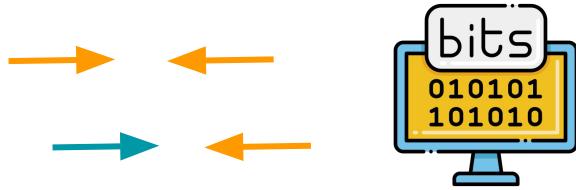
Switch to mate | Toggle first in pair / second in pair

Find SAM flag by property:
To find out what the SAM Flag value would be for a given combination of properties, tick the boxes for those that you'd like to include. The flag value will be shown in the SAM Flag field above.

<input checked="" type="checkbox"/> read paired	Summary: read paired (0x1)
<input checked="" type="checkbox"/> read mapped in proper pair	read mapped in proper pair (0x2)
<input type="checkbox"/> read unmapped	read reverse strand (0x10)
<input type="checkbox"/> mate unmapped	second in pair (0x80)
<input type="checkbox"/> read reverse strand	
<input type="checkbox"/> mate reverse strand	
<input type="checkbox"/> first in pair	
<input checked="" type="checkbox"/> second in pair	
<input type="checkbox"/> not primary alignment	
<input type="checkbox"/> read fails platform/vendor quality checks	
<input type="checkbox"/> read is PCR or optical duplicate	
<input type="checkbox"/> supplementary alignment	

FLAG
113
177
65
129
65
129
65
129
81
161
113
177
113
177
113
177
65
129
113
177
65
129
65
129
65
129
113
177

“The bitwise flag is a 16-bit integer that encodes various properties of the read and its alignment to the reference genome.”



Website interpret bitflag

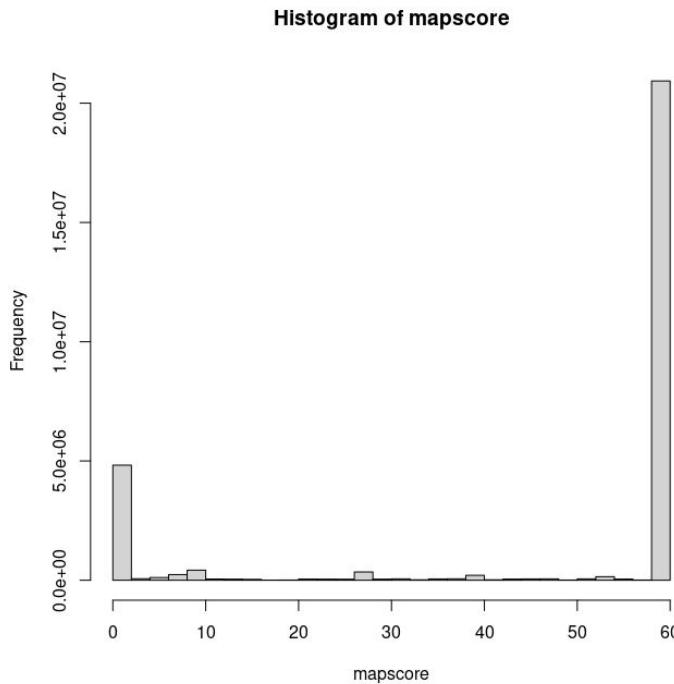
<https://broadinstitute.github.io/picard/explain-flags.html>

ALIGNMENT / MAPPING

ALIGNMENT

SAM/BAM file > MAPQ

MAPQ
60
60
0
0
60
60
27
27
46
46
0
0
60
60
60
60
60
0
0
60
60
60
60
60



“Mapping Quality Scores (MAPQ) quantify the probability that a read is misplaced.”

In BWA-MEM, MAPQ score:

- Based on Phred score
- Range: 0 to 60
- Higher scores indicating greater confidence in the mapping.
 - 0: read could map multiple locations.
 - 60: unique, highly confident mapping.
- Different mapping tools may produce a different MAPQ.

ALIGNMENT / MAPPING

ALIGNMENT

SAM/BAM file > CIGAR

CIGAR: 2M2D3M1I2M

- 2 matches
 - 2 deletes
 - 3 matches
 - 1 insert
 - 2 matches

CIGAR string describes how each read aligns to the reference genome.

Syntax :

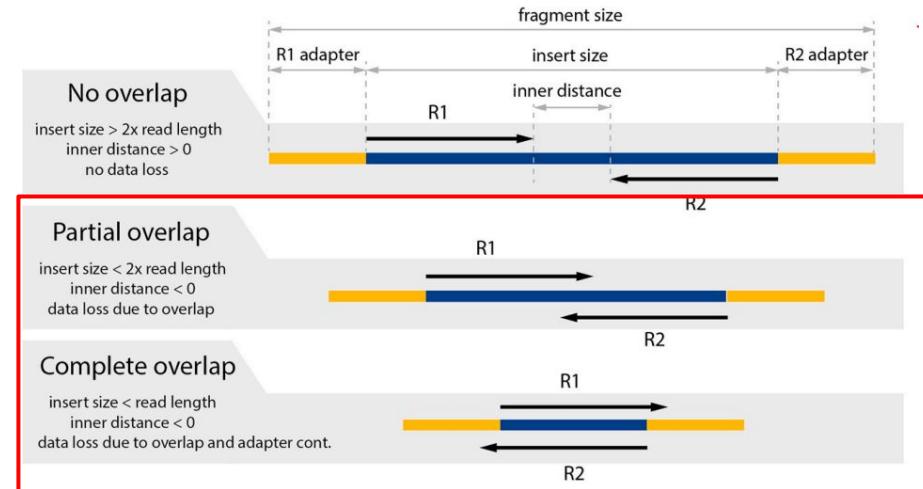
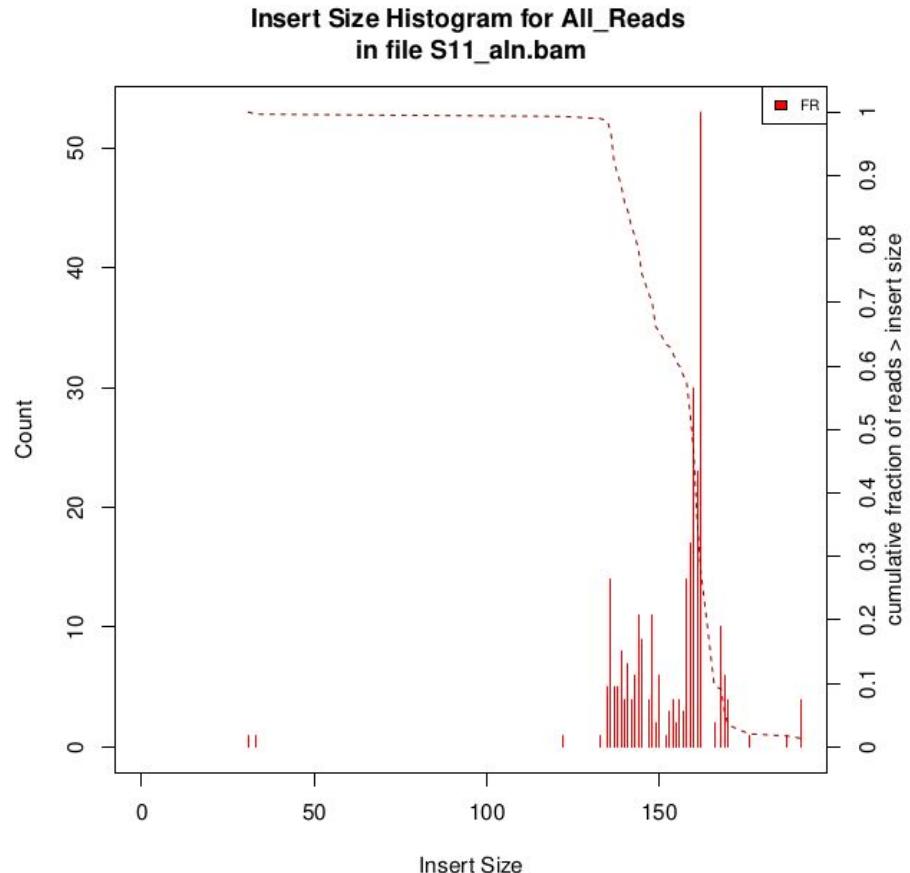
<length><operation>

Table 9.2. CIGAR operations

Op	Description
M	alignment match (sequence match or mismatch)
I	insertion (additional non-reference base)
D	deletion (reference base missing in the read)
N	skipped region from the reference
S	soft clipping (clipped sequences still present in SEQ)
H	hard clipping (clipped sequences not present in SEQ)
P	padding (silent deletion from padded reference)
=	sequence match
X	sequence mismatch

ALIGNMENT / MAPPING

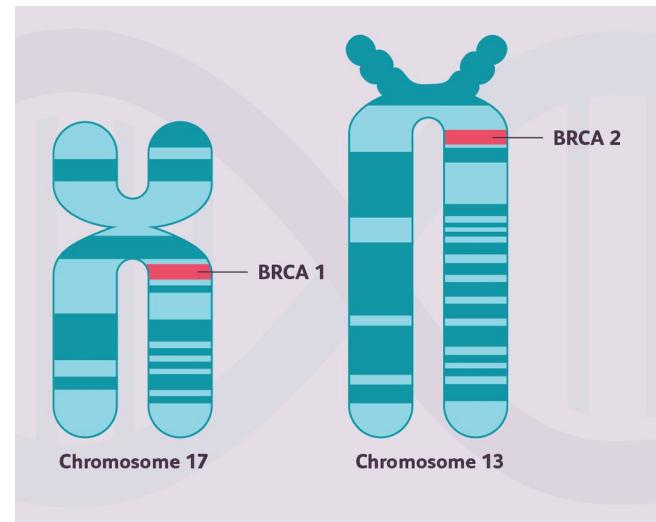
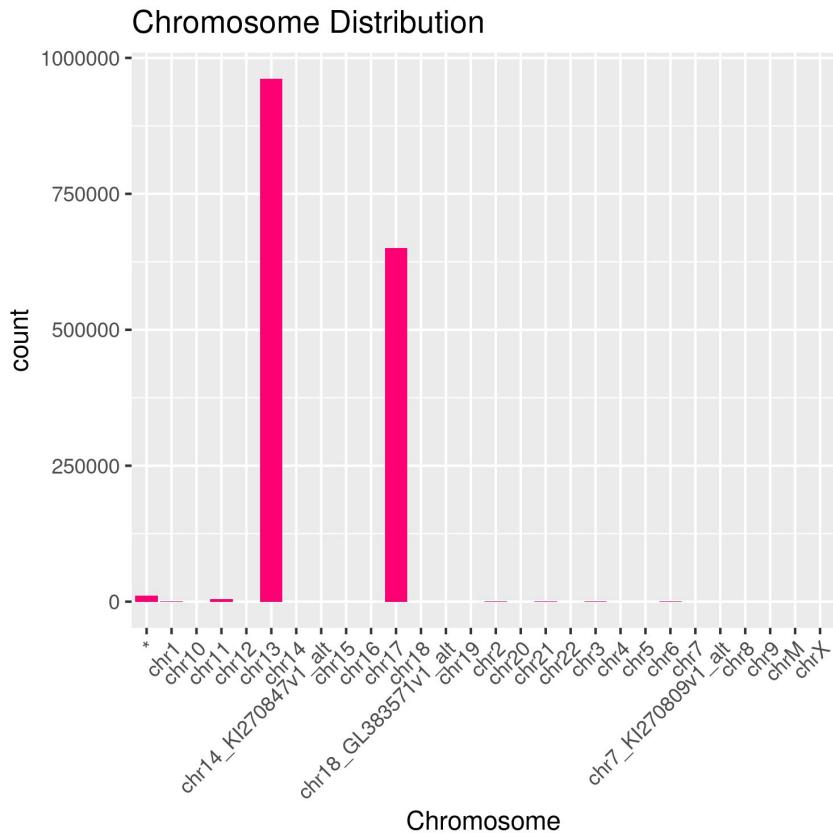
Insert size calculation



Insert size < 2x read length
→ Overlap & Partial overlap
→ Miss information

ALIGNMENT / MAPPING

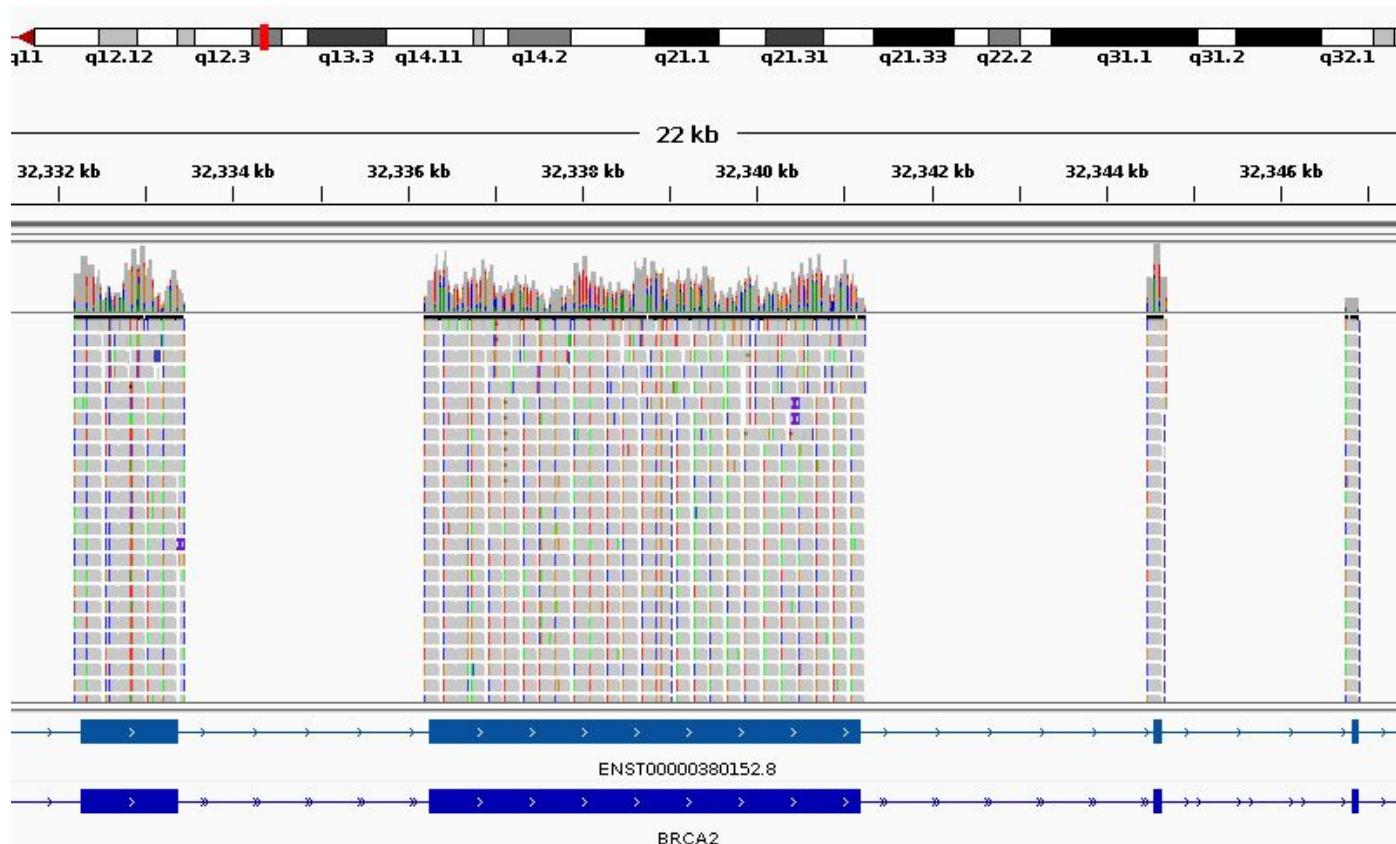
Chromosome mapped



Most reads mapped to chr13 & chr17

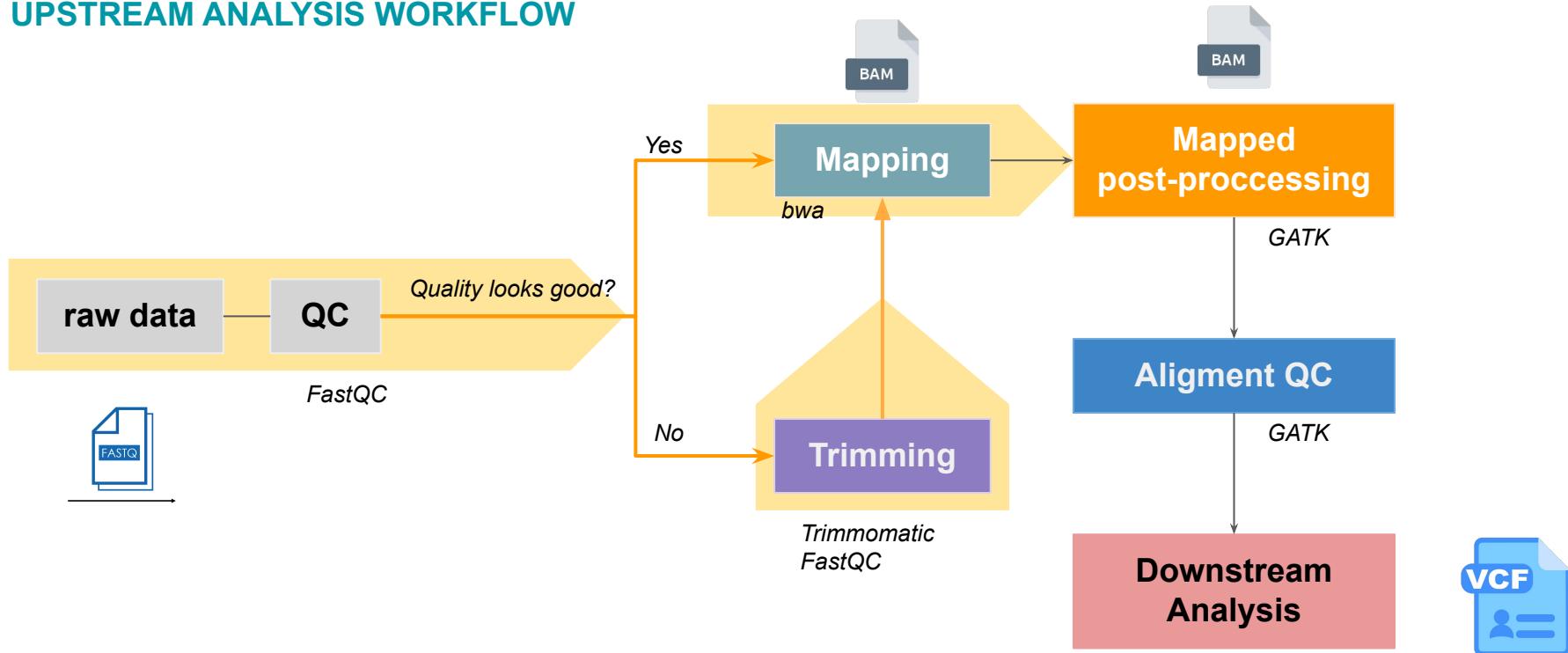
ALIGNMENT / MAPPING

Integrative Genomics Viewer (IGV)



ALIGNMENT / MAPPING

UPSTREAM ANALYSIS WORKFLOW



MAPPED READ POST-PROCESSING

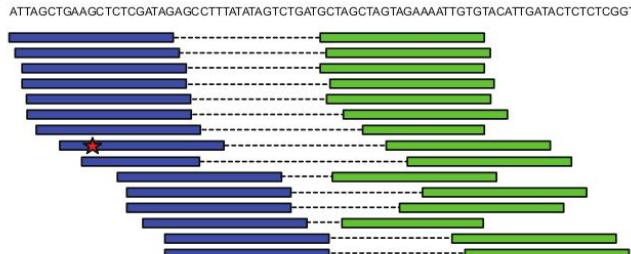
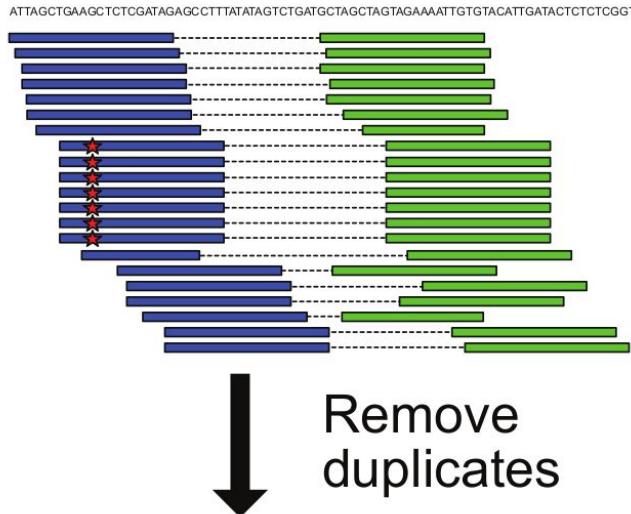
MAPPED READS POST-PROCESSING

Post-processing:

- Sorting, Indexing BAM file and Mark Duplicates

MAPPED READS POST-PROCESSING

Marking & removing duplicates

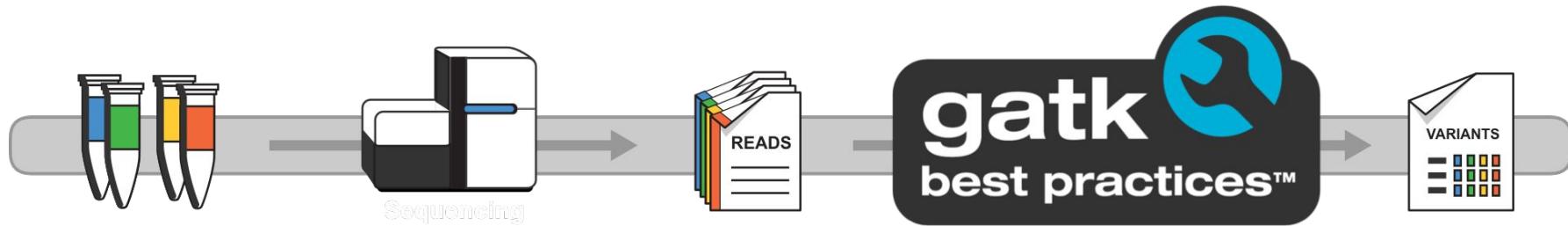


Duplicates read:

- PCR duplicates
- Optical duplicates

→ Duplicate reads can be problematic in downstream analyses, particularly in variant calling

MAPPED READS POST-PROCESSING

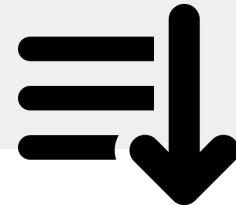


MAPPED READS POST-PROCESSING

Marking & removing duplicates

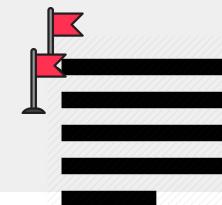
Sort the BAM file by coordinate

```
gatk SortSam \  
--INPUT S11_aln.bam \  
--OUTPUT S11_aln_sorted.bam \  
--SORT_ORDER coordinate
```



Mark Duplicates

```
gatk MarkDuplicates \  
--INPUT S11_aln_sorted.bam \  
--OUTPUT S11_aln_dedup.bam \  
--METRICS_FILE S11_MarkDup.metrics
```

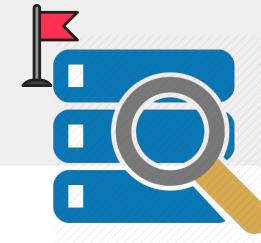


MAPPED READS POST-PROCESSING

View the marked duplicate reads with flag

```
samtools view S11_aln_dedup.bam
```

```
samtools view -f 0x400 S11_aln_dedup.bam
```

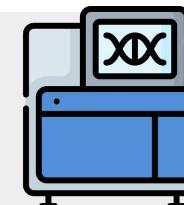


Discriminate Optical and PCR duplication

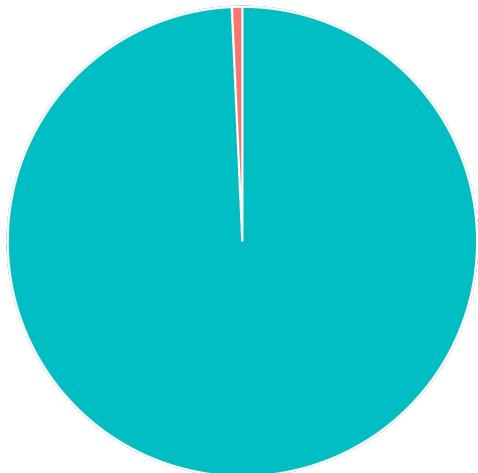
Marked the read as an optical duplicated (DT:Z:SQ) & PCR duplicated (DT:Z:LB)

```
gatk MarkDuplicates \
--INPUT S11_aln_sorted.bam \
--OUTPUT S11_aln_dedup.bam \
--METRICS_FILE S11_MarkDup_PCR_OPT.metrics \
--TAGGING_POLICY All
```

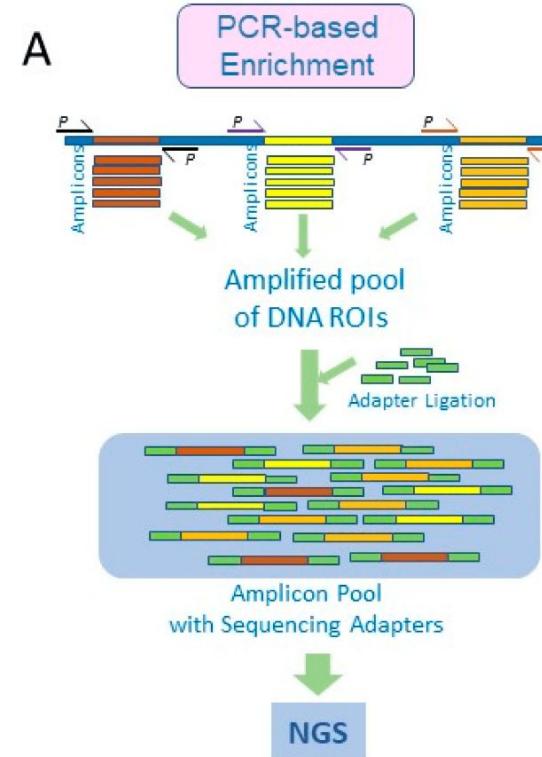
```
samtools view -f 0x400 NIST7035_dedup.bam | grep DT:Z:SQ | less -S
```



MAPPED READS POST-PROCESSING



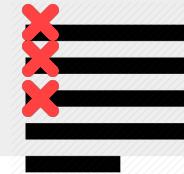
Most are PCR duplicates



MAPPED READS POST-PROCESSING

Remove Duplicate

```
gatk MarkDuplicates \  
--INPUT $p_align/sample2/NIST7035_sorted.bam \  
--OUTPUT $p_align/sample2/NIST7035_remove_dup.bam \  
--METRICS_FILE $p_align/sample2/NIST7035_remove_dup.metrics2 \  
--REMOVE_DUPLICATES true
```



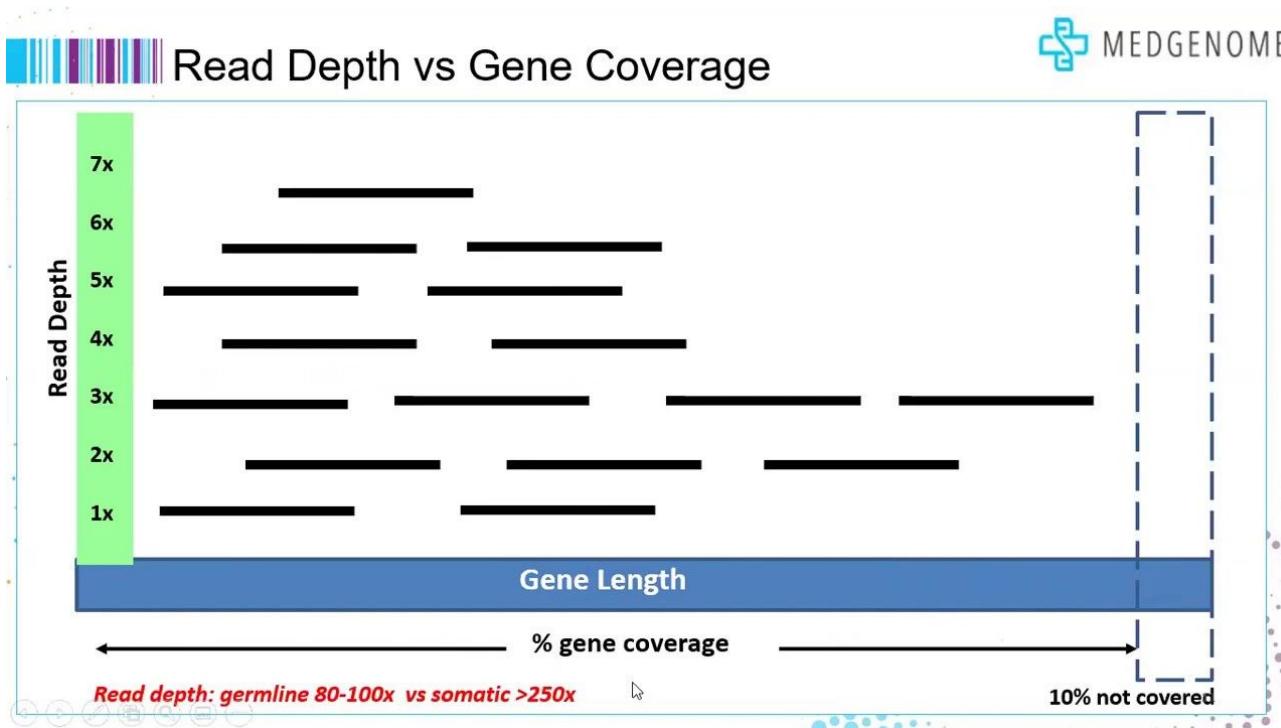
But should we remove duplicates?

ALIGNMENT DATA: QUALITY CONTROL

2 key parameters for Alignment Quality Control:

- Read Depth
- Coverage

ALIGNMENT DATA: QUALITY CONTROL

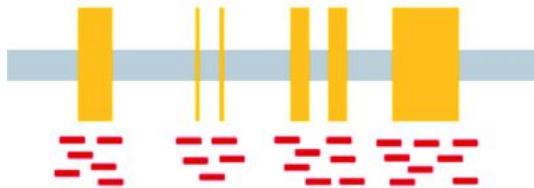


The more reads there are, the more certain we can be about the genotype at any given position.

ALIGNMENT DATA: QUALITY CONTROL

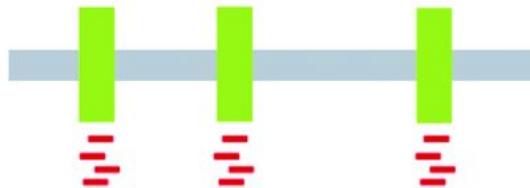
WES & Target Sequencing

Whole exome sequencing



- Region sequencing: whole exome
- Depth of coverage: 20X
- Identify all variants: SNV, INDELs, SV in coding Regions
- Cost effective

Targeted sequencing



- Region sequencing: specific coding region
- Depth of coverage: 300X
- Identify all variants: SNV, INDELs, SV in specific regions
- Most cost effective



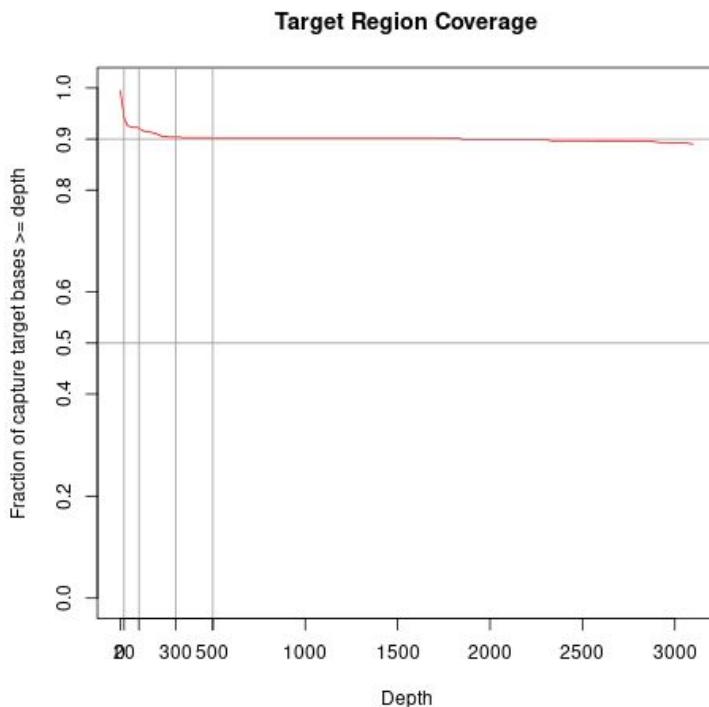
chr13	32316460	32316527
chr13	32370401	32370557
chr13	32370955	32371100
chr13	32376669	32376791
chr13	32379316	32379515
chr13	32379749	32379913
chr13	32380006	32380145
chr13	32394688	32394933
chr13	32396897	32397044
chr13	32398161	32398770
chr17	43045677	43045802
chr17	43047642	43047703
chr17	43049120	43049194
chr17	43051062	43051117
chr17	43057051	43057135
chr17	43063332	43063373
chr17	43063873	43063951
chr17	43067607	43067695
chr17	43070927	43071238

ALIGNMENT DATA: QUALITY CONTROL

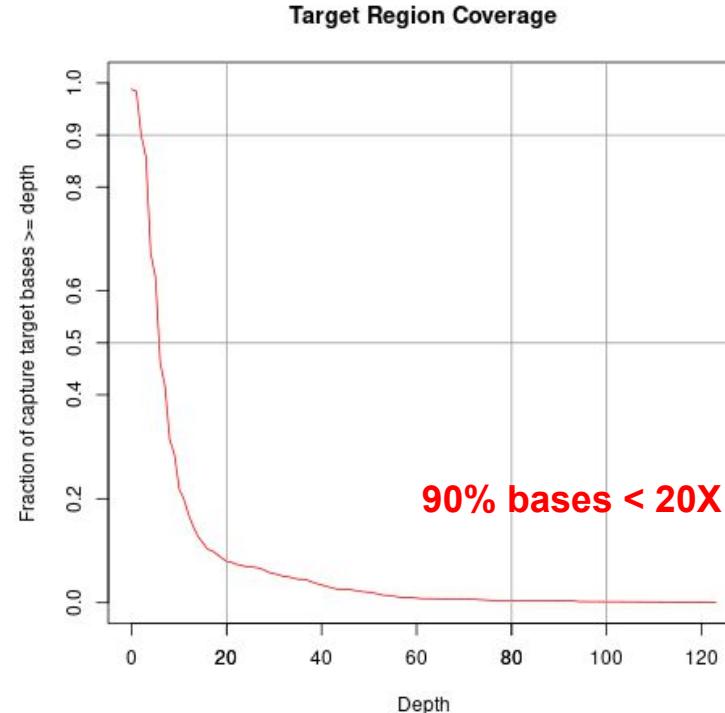
Calculate coverage

```
bedtools coverage \
-hist \
-a $p_ref/hg38_exome.bed \
-b $p_align/sample2/NIST7035_remove_dup.bam > NIST.bed.cov
#
grep ^all NIST.bed.cov > NIST.all.cov
```

ALIGNMENT DATA: QUALITY CONTROL



Keep duplicates

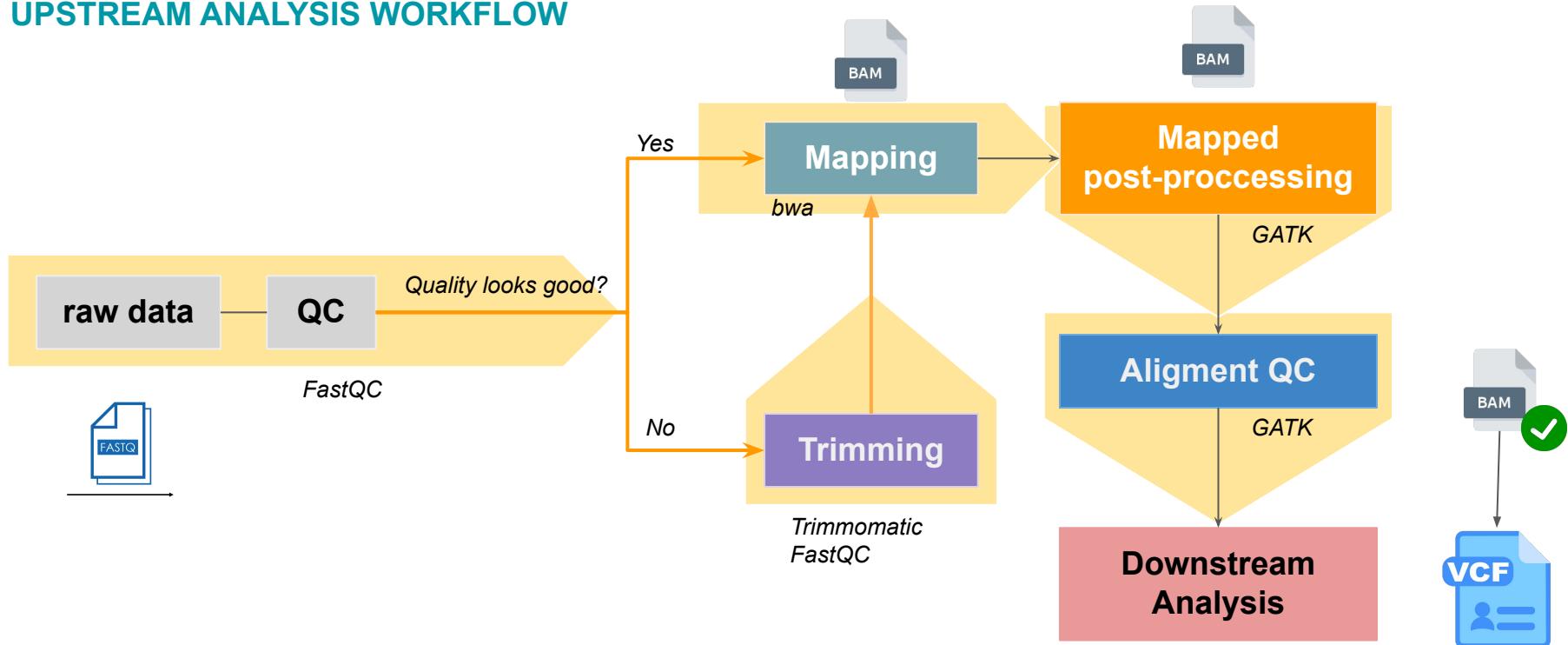


Remove duplicates

A depth of coverage of 300X or more is considered to be sufficient for most applications.

SUMMARY

UPSTREAM ANALYSIS WORKFLOW

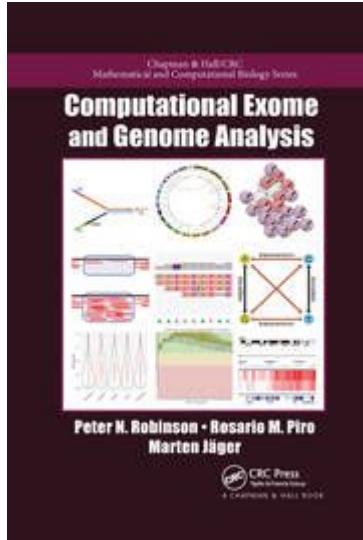


SUMMARY

- Low quality reads filtered
- Contaminants eliminated
- Alignment rate and quality is good
- Almost duplicated reads come from PCR
- Keep duplicates

→ **Quality reads are ready for downstream analysis**

PREFERENCE



Robinson, P.N., Piro, R.M., & Jäger, M. (2017). Computational Exome and Genome Analysis (1st ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9781315154770>

THANK YOU