Human Genomics and Epigenomics

Pratical 1 – 18/01/2021
Pratical 2 – 19/01/2021
Pratical 3 – 25/01/2021
Pratical 4 – 26/01/2021

Prof. Massimo Delledonne
Functional Genomics lab

ALIGNMENT AND VARIANT CALLING

1° Day (3h): Pre-processing of raw reads

- The fastq file
- Quality control of fastq files
- Adapter removing and trimming of fastq files
  - Sickle and scythe
  - Trimmomatic
- Reads alignment:
  - The human reference genome (hg19 and hg38, main differences)
  - The BAM file

2° Day (3h): Alignment

- Alignment of trimmed reads to the reference genome
  - BWA-mem
  - Isaac2 pipeline
- Duplicates removal
- Read Clipping
- Visualization of aligned reads on IGV

ALIGNMENT AND VARIANT CALLING

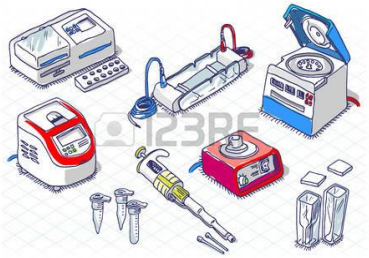
3° Day (3h): Statistics and Variant Calling

- Statistics on reads alignment: main parameters for the evaluation of NGS data
  - Average coverage and uniformity
  - Fold enrichment (on/near/off target)
  - Genotypability (mapping quality besides coverage)
- Variant calling:
  - The VCF and gVCF files
  - Germline variant calling
    - GATK4 Best practice pipeline

4° Day (3h): Variant Calling

- Germline variant calling
  - GATK4 Best practice pipeline
  - Strelka2
- Visualization of genetic variants on IGV
- CNV detection

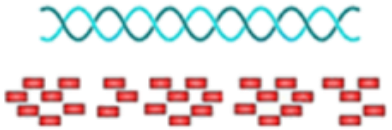
Library preparation



Sequencing

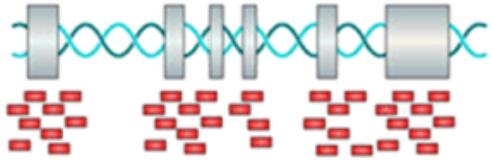
Bioinformatic analysis

Whole genome sequencing



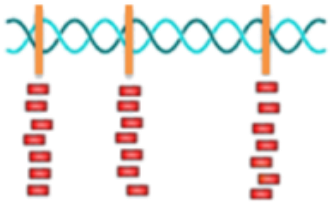
- Sequencing region : whole genome
- Sequencing Depth: >30X
- Covers everything – can identify all kinds of variants including SNPs, INDELs and SV.

Whole exome sequencing



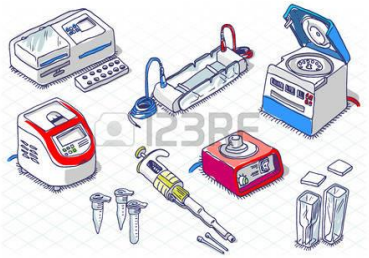
- Sequencing region: whole exome
- Sequencing Depth : >50X ~ 100X
- Identify all kinds of variants including SNPs, INDELs and SV in coding region.
- Cost effective

Targeted sequencing



- Sequencing region: specific regions (could be customized)
- Sequencing Depth : >500X
- Identify all kinds of variants including SNPs, INDELs in specific regions
- Most Cost effective

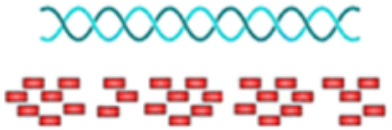
Library preparation



Sequencing

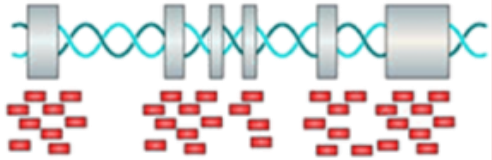
Bioinformatic analysis

Whole genome sequencing



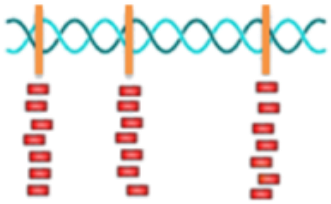
- Sequencing region : whole genome
- Sequencing Depth: >30X
- Covers everything – can identify all kinds of variants including SNPs, INDELs and SV.

Whole exome sequencing



- Sequencing region: whole exome
- Sequencing Depth : >50X ~ 100X
- Identify all kinds of variants including SNPs, INDELs and SV in coding region.
- Cost effective

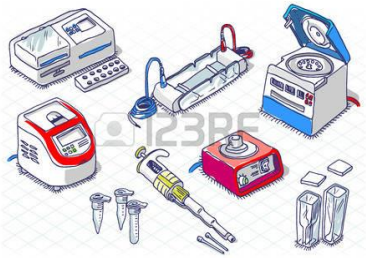
Targeted sequencing



- Sequencing region: specific regions (could be customized)
- Sequencing Depth : >500X
- Identify all kinds of variants including SNPs, INDELs in specific regions
- Most Cost effective

Library preparation

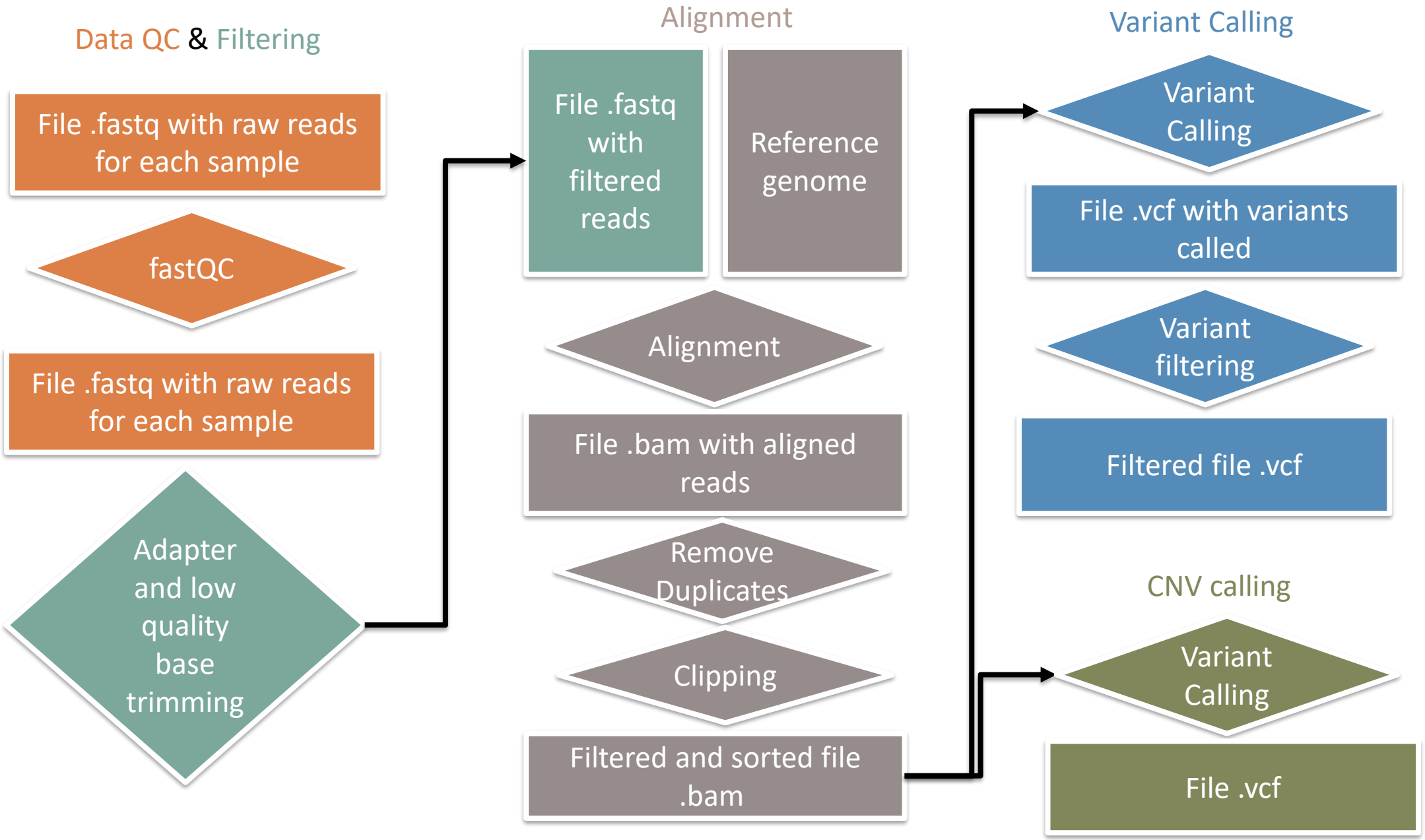
Bioinformatic analysis



Sequencing

Data QC → Alignment → Variant calling → Variant annotation → Variant prioritization

Pipeline


Data QC & Filtering
File .fastq with raw reads for each sample
fastQC
File .fastq with raw reads for each sample
Adapter and low quality base trimming
Alignment
File .fastq with filtered reads
Reference genome
Alignment
File .bam with aligned reads
Remove Duplicates
Clipping
Filtered and sorted file .bam
Variant Calling
Variant Calling
File .vcf with variants called
Variant filtering
Filtered file .vcf
CNV calling
Variant Calling
File .vcf

Shell

Windows:

- https://mobaxterm.mobatek.net/download.html

MAC & Linux:

- Open terminal

Connect to the server

1. Enter in the server:
   a. ssh lessons@157.27.80.26
   b. Password: lez2021
2. Create your folder: mkdir HGE_2021/your_name
3. Enter in the created folder: cd HGE_2021/your_name

Work on the server

1. Create a symbolic link of the files in your folder:

   ln -s ../trio_1351S/1351S/R*fastq.gz .

2. Check you have copied the files: ls

3. Open the file to see what is inside:

   less R1.fastq.gz

4. Close the visualization: q

.fq / .fastq file

For each sample we obtain 2 fastq files containing all the sequences generated

Sequence1.fastq.gz

```
@HWI-1KL152:89:C118FACXX:8:1209:20044:2901/1
CTCTGTGGCTGGGAGAGGAGTCTGGGGGGGCCCCGGGCGCCAGCCAGGGATAGCCTGATCTCTGCTCCAGTCGACAGATCCTTAACGGATTTTCTTTCTCT
+
/%53@DDAFGIDCCD;EBCE:C>DDCBB&:9<FD5=02&4?8=<CCCA<BB>FFDD=B@C98,>7:<9-9/0&74>CEBEHHBCDHA?=;&A9DHB#####
@HWI-1KL152:89:C118FACXX:8:1313:2904:90657/1
TAGACCCATCAGGACTGGAGTTCCACTCATTCTTGATAGGCAGCCATTGTGTAACTCTGCAATCGCCTTTTCTCTCTGACAGATGGAGAACTCTCTATATC
+
##########################################################################################################
@HWI-1KL152:89:C118FACXX:8:2311:2270:56622/1
CCGTGCCACACTAGATGGAAAGGCAGACATTGTCTAAGACTTCTATCACCCTTTCTCTCTGCCAGATGGCTATCTCTCTGTATACCATTAATGTGGTCTCC
+
##########################################################################################################
```

Sequence2.fastq.gz

```
@HWI-1KL152:89:C118FACXX:8:1209:20044:2901/2
CCTCTCCCTCCGTCTCCTTCTCCCTCTCAGCTCGCCGGGCGACCCTGCTCCTGCCTCCCACATTAATGGCGGCATCCTCGGAGGATGATATAGACCGGCGG
+
#####A<CFEE=BDDBHE@A?EFGEAD?D@;A<6BD>A@DACAAGHDCEBHH@CGEB>CHFHDB?EAD=@@ACC:BEGH=CEDAAB=F@G>AE;89?DA:D
@HWI-1KL152:89:C118FACXX:8:1313:2904:90657/2
ATGTTCCACTCTTTCTTGAAAGTCAGACATTGTCTAACTCTGCAATTTTCTTTTCCCTCTGACACATCTGGTACTATCGATATCCAATGAAGAGGACTNAC
+
##########################################################################################################
@HWI-1KL152:89:C118FACXX:8:2311:2270:56622/2
TCAGCAGGTGGGAATTACTACTATTTCTTGAAAGGCAGACCTTGTATAACTCTGCCATATCCTTTTCTCTCTGCCAGATGGGGATCTATCTATTTCCACTG
+
64=@'<A#################################################################################################
```

.fq / .fastq file

Each read is represented by four rows:

```
@HWI-1KL152:89:C118FACXX:8:1209:20044:2901/1
CTCTGTGGCTGGGAGAGGAGTCTGGGGGGGCCCCGGGCGCCAGCCAGGGATAGCCTGATCTCTGCTCCAGTCGACAGATCCTTAACGGATTTTCTTTCTCT
+
/%53@DDAFGIDCCD;EBCE:C>DDCBB&:9<FD5=02&4?8=<CCCA<BB>FFDD=B@C98,>7:<9-9/0&74>CEBEHHBCDHA?=;&A9DHB#####
```

First row identifies the sequence:

Instrument ID

Flowcell ID

Tile Coordinates

```
@HWI-1KL152:89:C118FACXX:8:1209:20044:2901/1
```

Run Number

Lane

Read ID

Second row contains the sequence:

```
CTCTGTGGCTGGGAGAGGAGTCTGGGGGGGCCCCGGGCGCCAGCCAGGGATAGCCTGATCTCTGCTCCAGTCGACAGATCCTTAACGGATTTTCTTTCTCT
```

Third row contains a delimiter:

```
+
```

Fourth row indicates the quality of each sequenced base:

```
CTCTGTGGCTGGGAGAGGAGTCTGGGGGGGCCCCGGGCGCCAGCCAGGGATAGCCTGATCTCTGCTCCAGTCGACAGATCCTTAACGGATTTTCTTTCTCT
/%53@DDAFGIDCCD;EBCE:C>DDCBB&:9<FD5=02&4?8=<CCCA<BB>FFDD=B@C98,>7:<9-9/0&74>CEBEHHBCDHA?=;&A9DHB#####
```

Q score as ASCII chars: "/" = 47

ASCII CODE

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

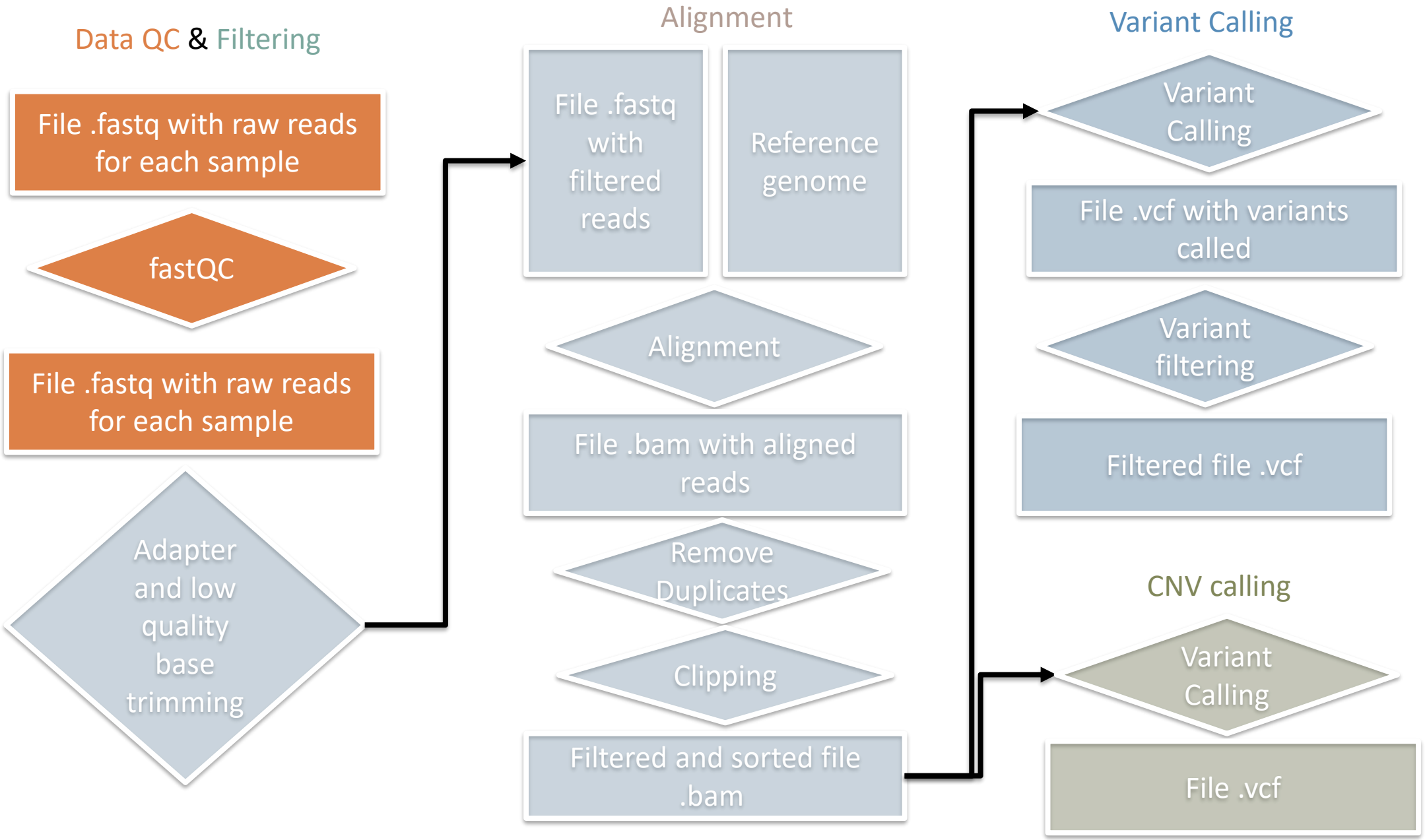
Illumina Quality

Q = ASCII - 33

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Pipeline


Data QC & Filtering
File .fastq with raw reads for each sample
fastQC
File .fastq with raw reads for each sample
Adapter and low quality base trimming
Alignment
File .fastq with filtered reads
Reference genome
Alignment
File .bam with aligned reads
Remove Duplicates
Clipping
Filtered and sorted file .bam
Variant Calling
Variant Calling
File .vcf with variants called
Variant filtering
Filtered file .vcf
CNV calling
Variant Calling
File .vcf

CHECK QUALITY OF YOUR FASTQ FILES WITH FASTQC

Fastqc command

1. In your folder, create a folder for fastqc output:

   mkdir fastqc

2. Launch fastQC on both files:

   fastqc R*.fastq.gz -o fastqc

Download the Fastqc files

On the server, we don't have a graphical visualization, therefore:

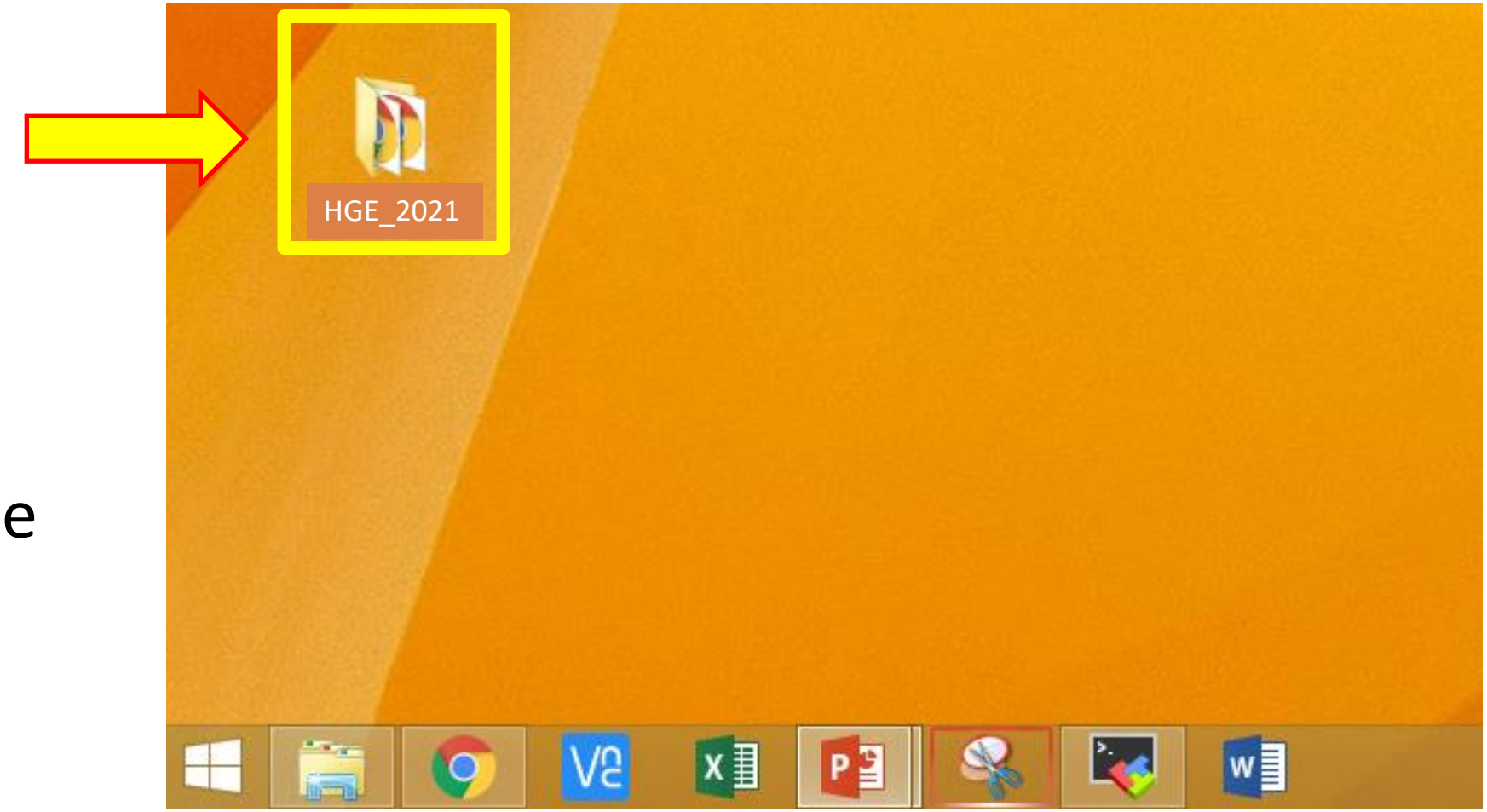
1. Open a new terminal
2. Create a folder on your PC for the course: mkdir Desktop/HGE_2021
3. Enter in the folder: cd Desktop/HGE_2021
4. Download the results from here:

   rsync -auv lessons@157.27.80.26:/home/lessons/HGE_2021/denise/fastqc/R*_fastqc.html .

   Pass: lez2021
5. Check what you have downloaded: ls
6. Close the shell

Open the Fastqc files

1. On your desktop, open the folder «HGE_2021»



2. Open the html file

FastQC

FastQC software allows to do some quality control checks on raw sequence data coming from high throughput sequencing.

HTML Report

FastQC Report

Thu 21 Dec 2017
good_sequence_short.txt

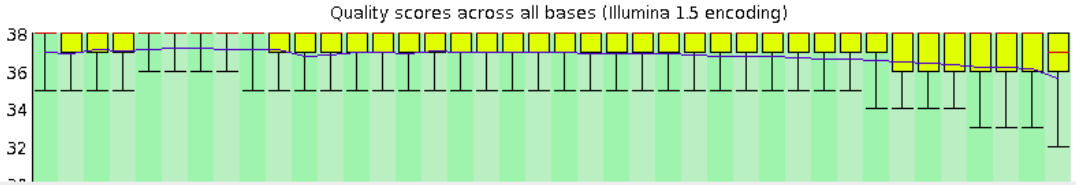
Summary

- 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

Basic Statistics

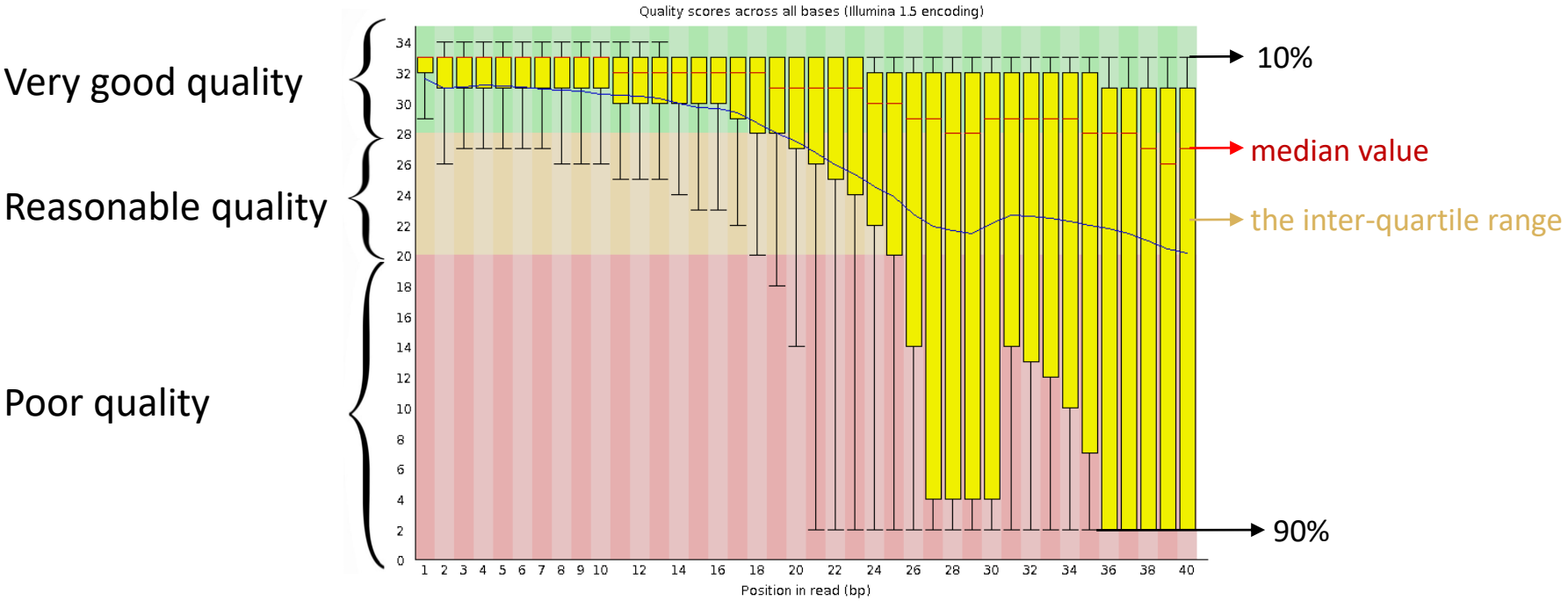
Measure	Value
Filename	good_sequence_short.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	40
%GC	45

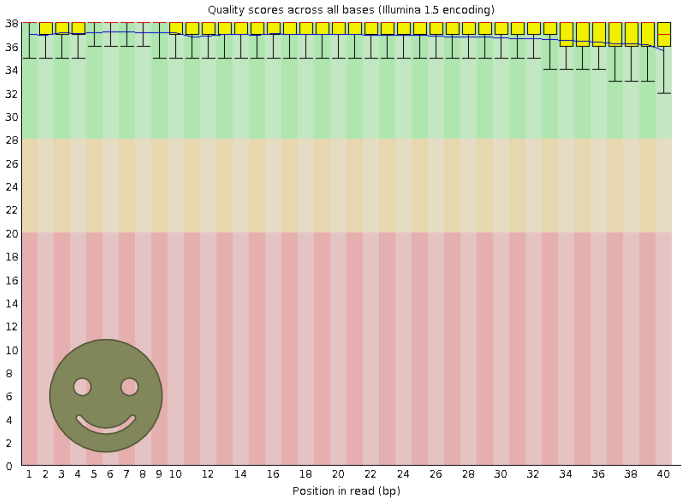
Per base sequence quality

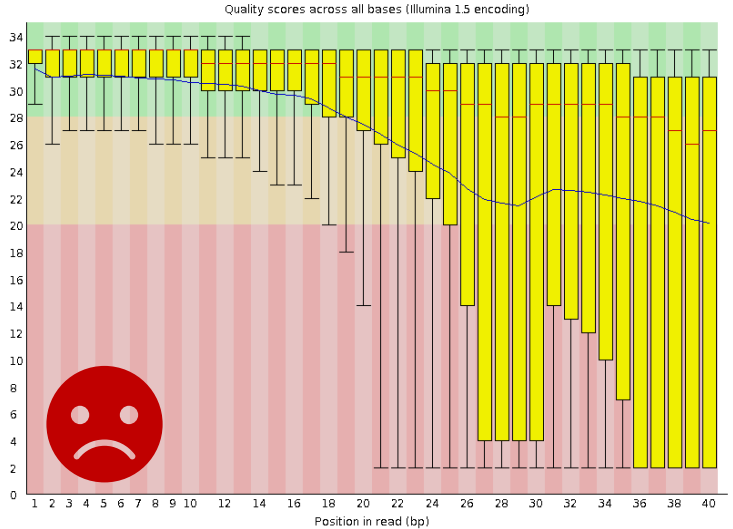


Produced by FastQC (version 0.11.6)

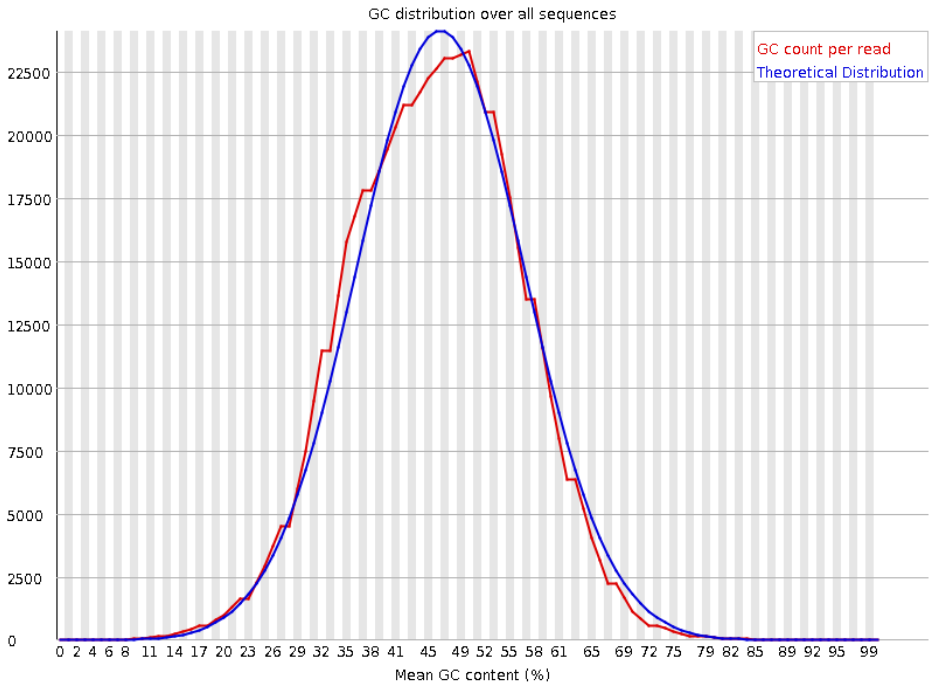
FastQC – Per base sequence quality



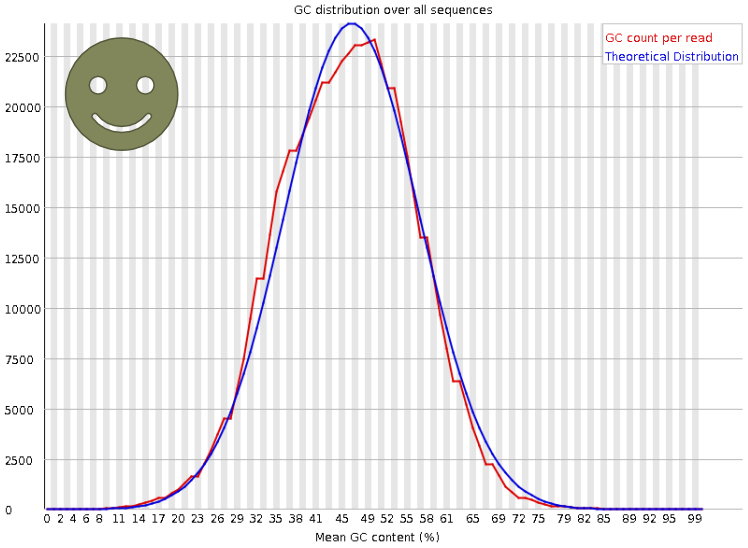


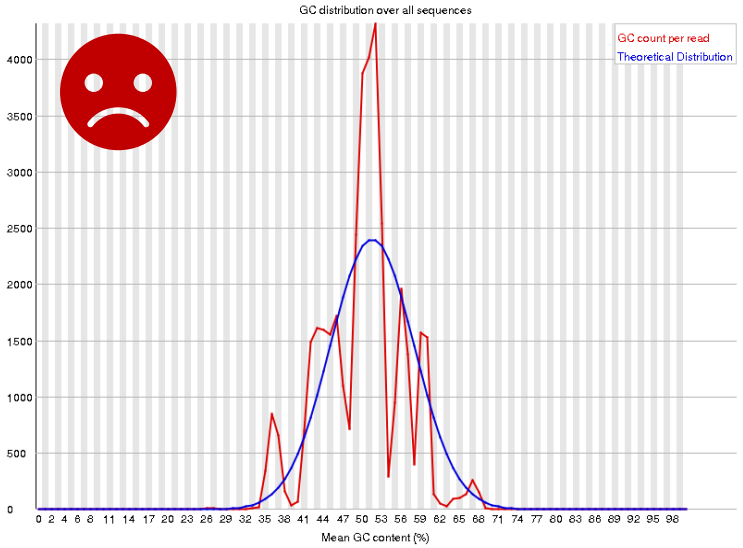


FastQC – Per sequence GC content

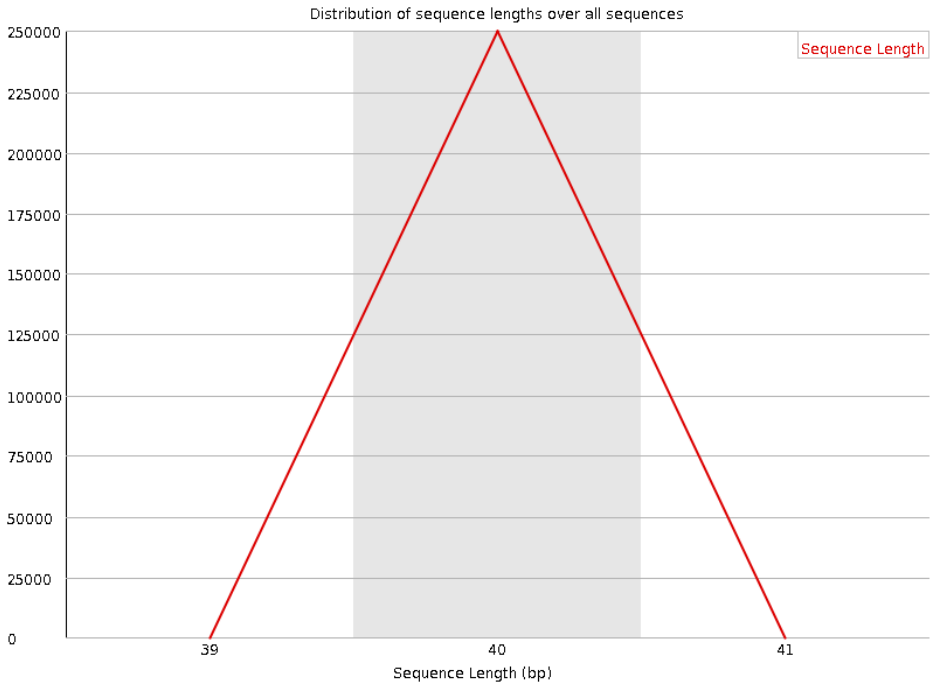


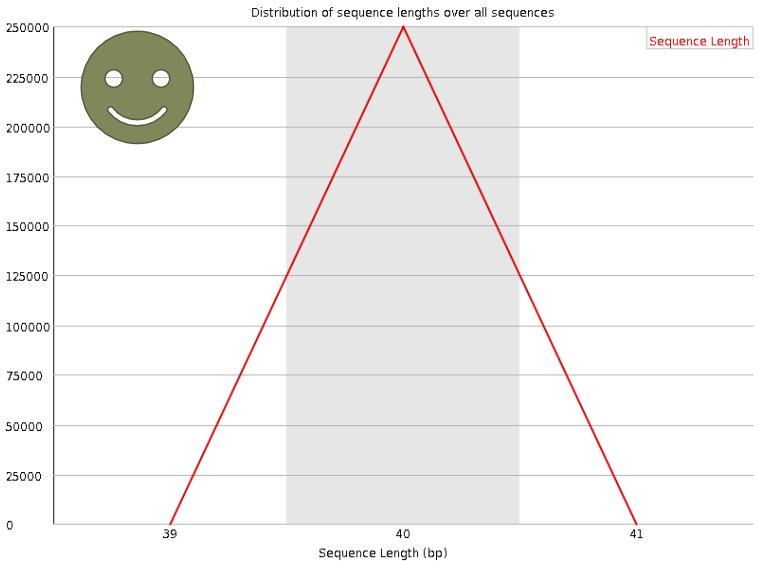
Measure of the GC content across the whole length of each sequence and compares it to a modelled normal distribution of GC content



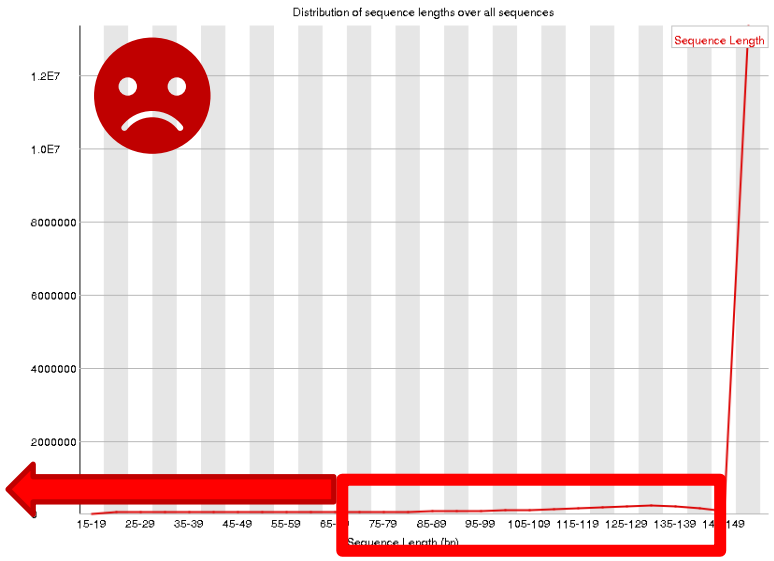


FastQC – Sequence Length Distribution

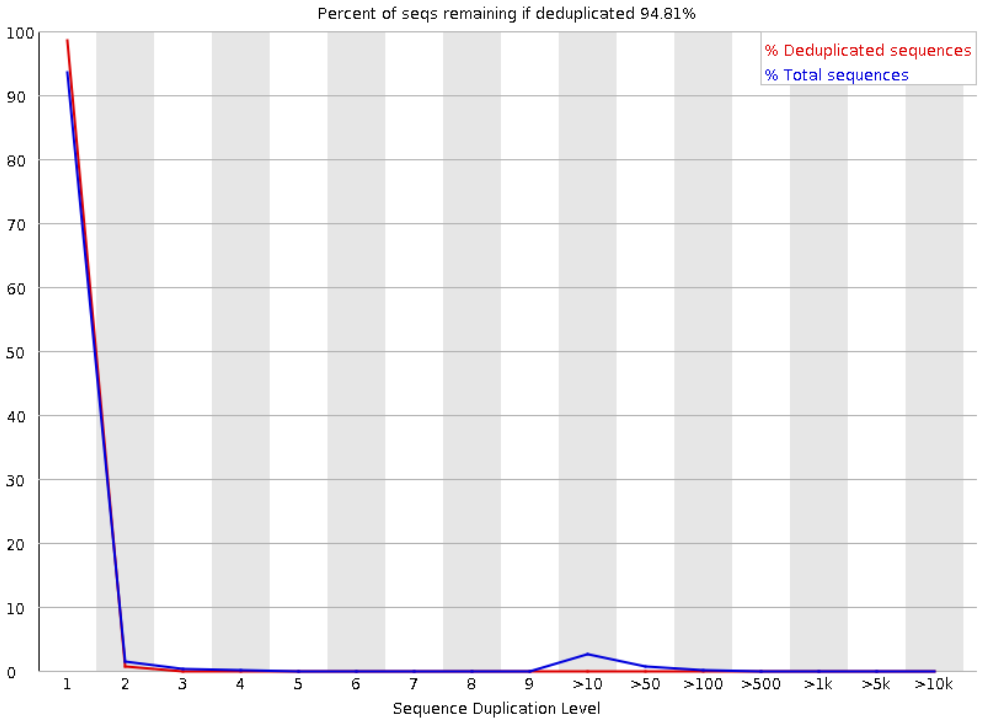


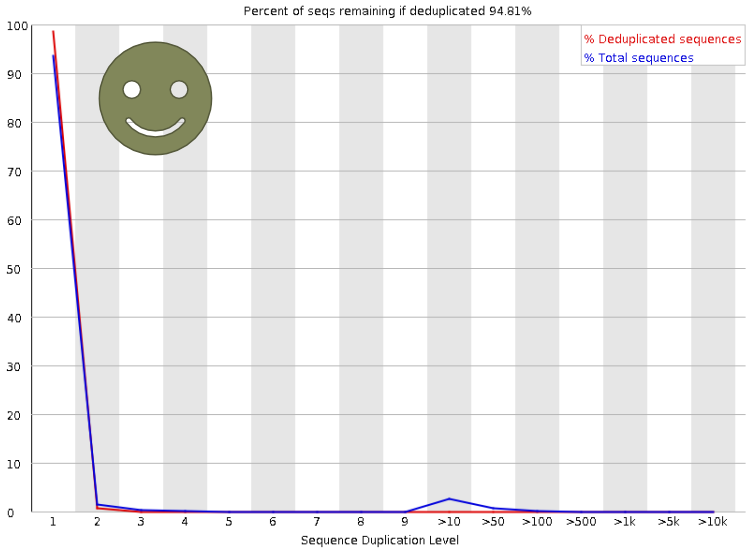


Some sequences have different length or zero length

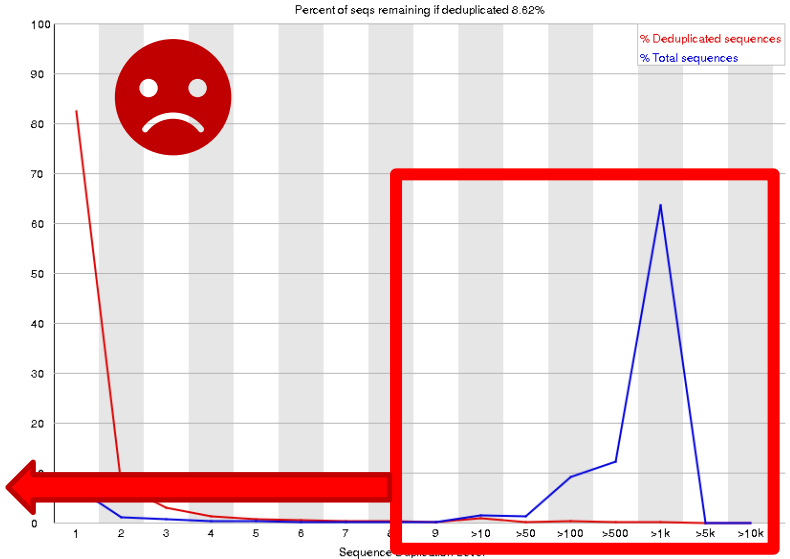


FastQC – Sequence Duplication Levels

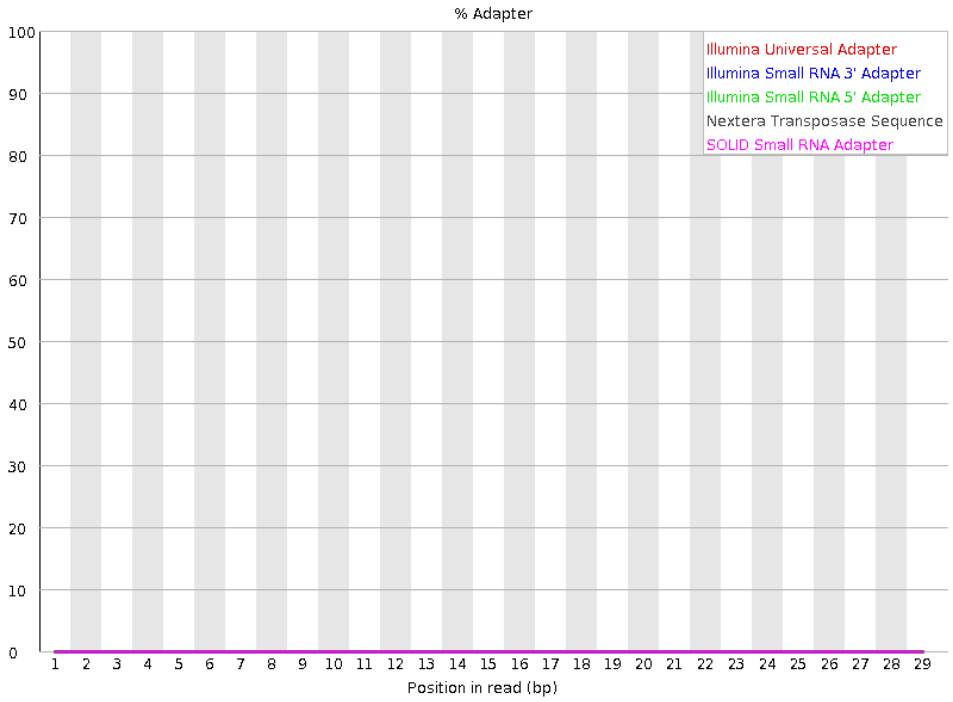


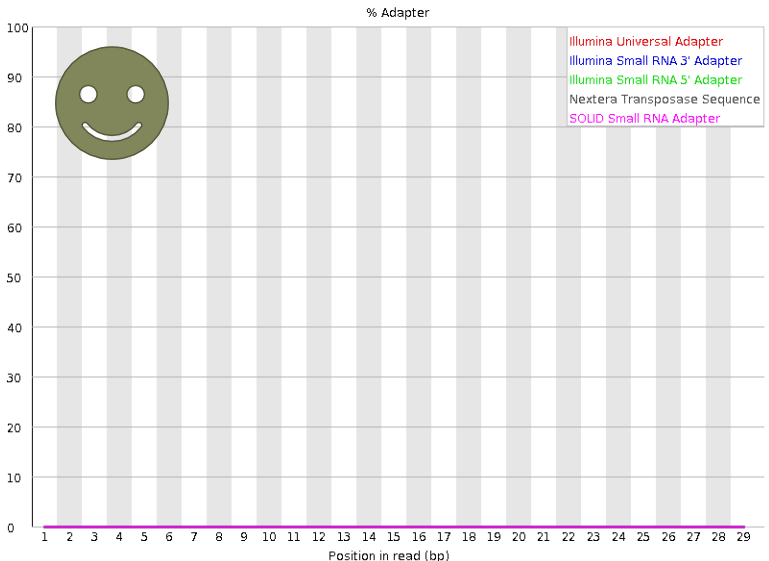


Very large number of sequences with high levels of duplication

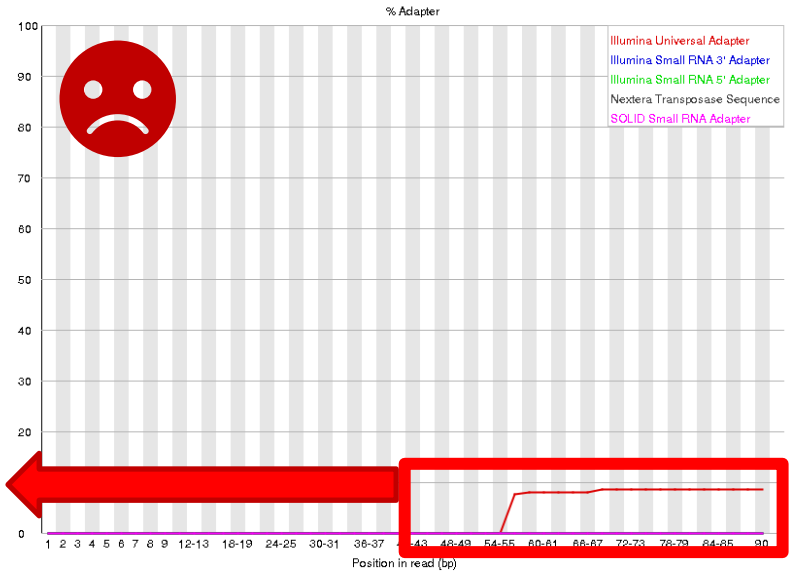


FastQC – Adapter Content

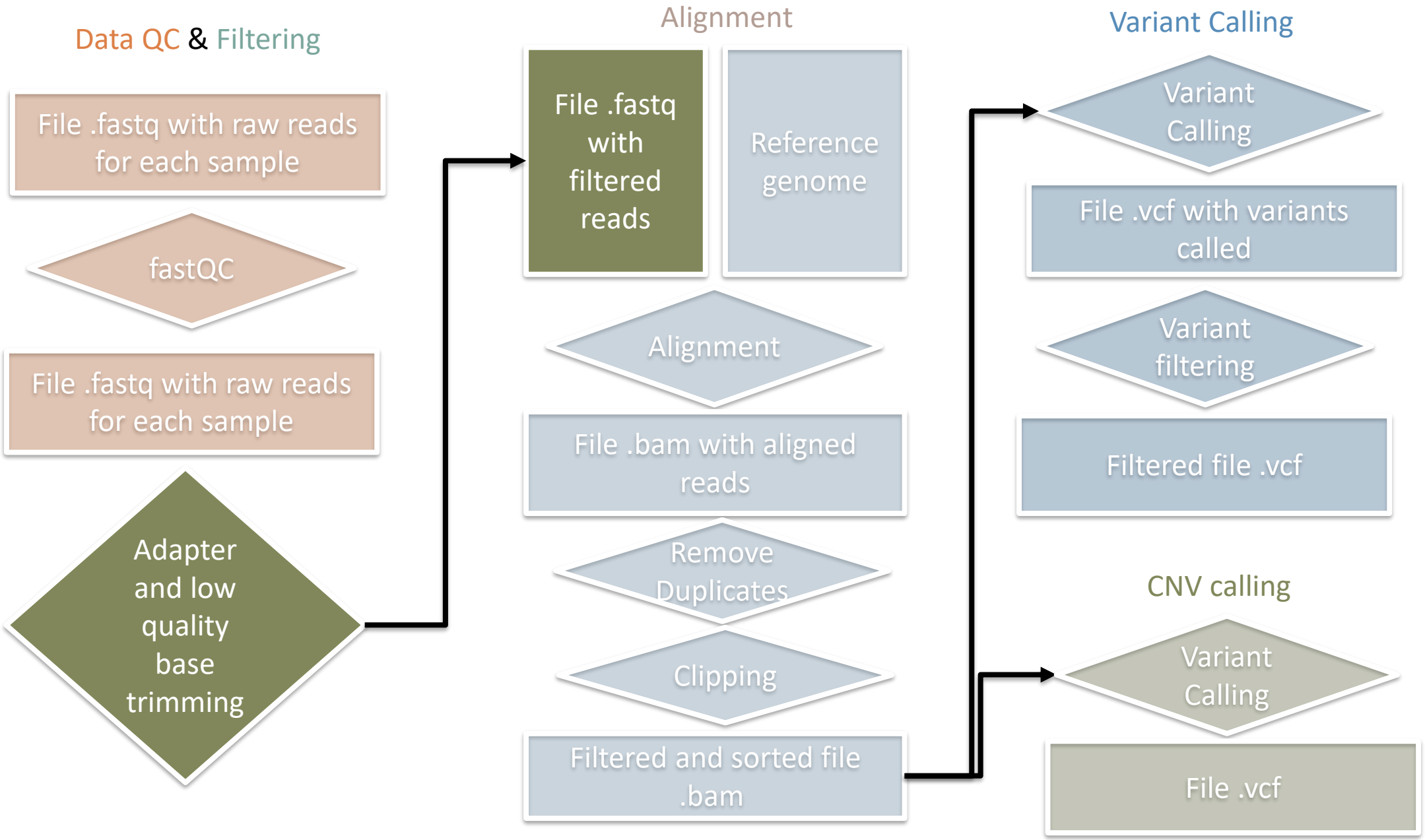




Some sequences contains adapters



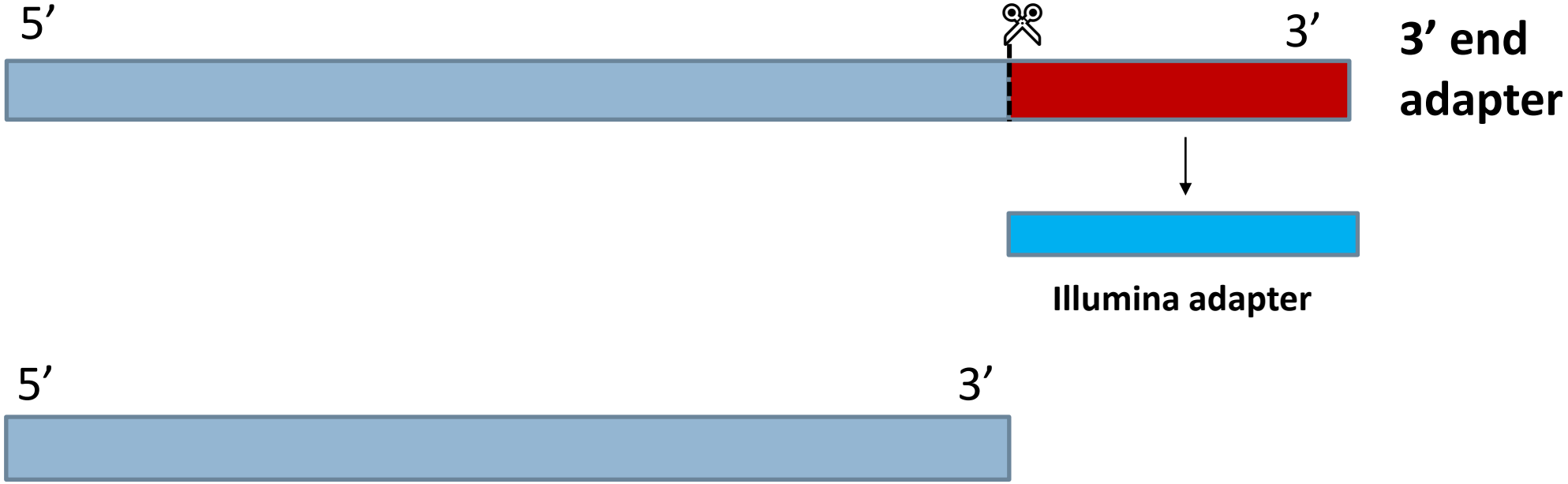
Pipeline


Data QC & Filtering
File .fastq with raw reads for each sample
fastQC
File .fastq with raw reads for each sample
Adapter and low quality base trimming
Alignment
File .fastq with filtered reads
Reference genome
Alignment
File .bam with aligned reads
Remove Duplicates
Clipping
Filtered and sorted file .bam
Variant Calling
Variant Calling
File .vcf with variants called
Variant filtering
Filtered file .vcf
CNV calling
Variant Calling
File .vcf

ADAPTER AND LOW QUALITY BASE TRIMMING

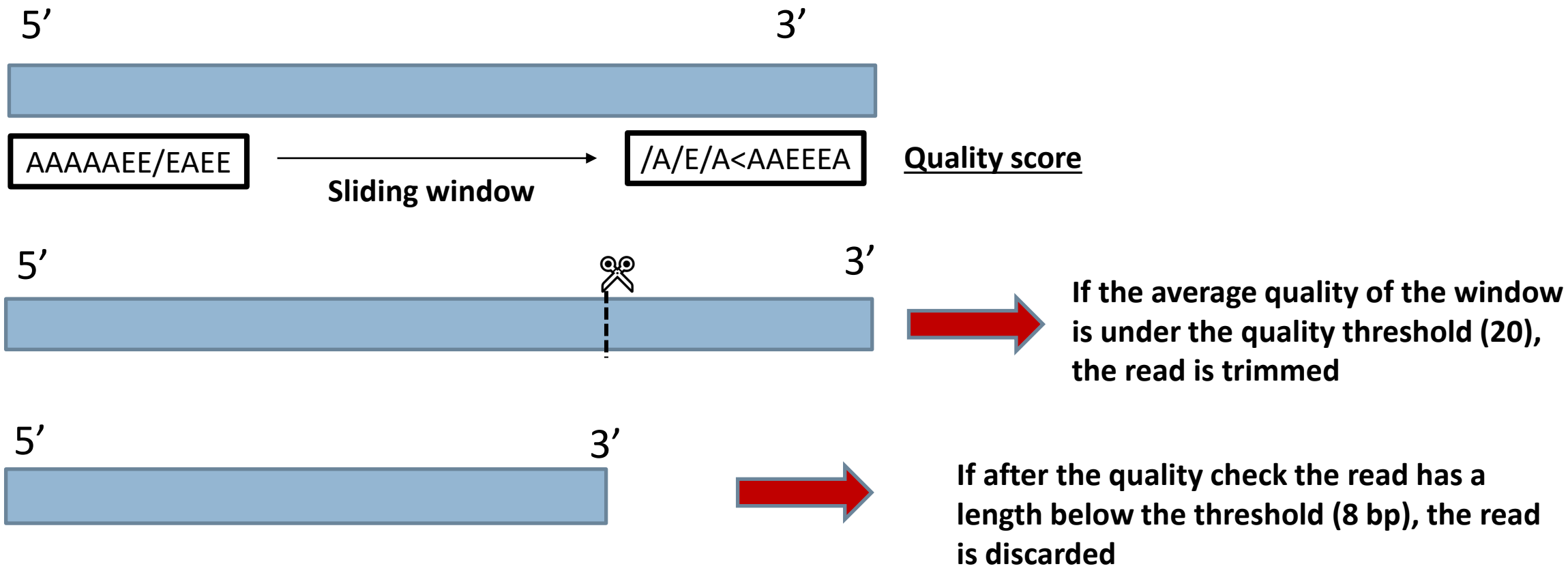
Adapter Trimming: scythe

<u>Scythe</u>: uses a **Naive Bayesian approach** to classify contaminant substrings in sequence reads. **It considers quality information**, which can make it robust in picking out 3'-end adapters, which often include poor quality bases.



Low quality base trimming: Sickle

<u>Sickle</u>: a tool that uses **sliding windows along with quality and length thresholds** to determine when quality is sufficiently low to trim the 3'-end of reads



Trimming command

Remove adapters from both reads and trimm reads:

```
sickle pe -g -t sanger  -f <( scythe -a ../ref/illumina_adapters.fa -q sanger R1.fastq.gz)  -r <( scythe -a ../ref/illumina_adapters.fa -q sanger R2.fastq.gz ) -o trimmed1.fastq.gz -p trimmed2.fastq.gz -s /dev/null
```

Control the quality of trimmed reads with FASTQC

```
fastqc trimmed*.fastq.gz -o fastqc
```

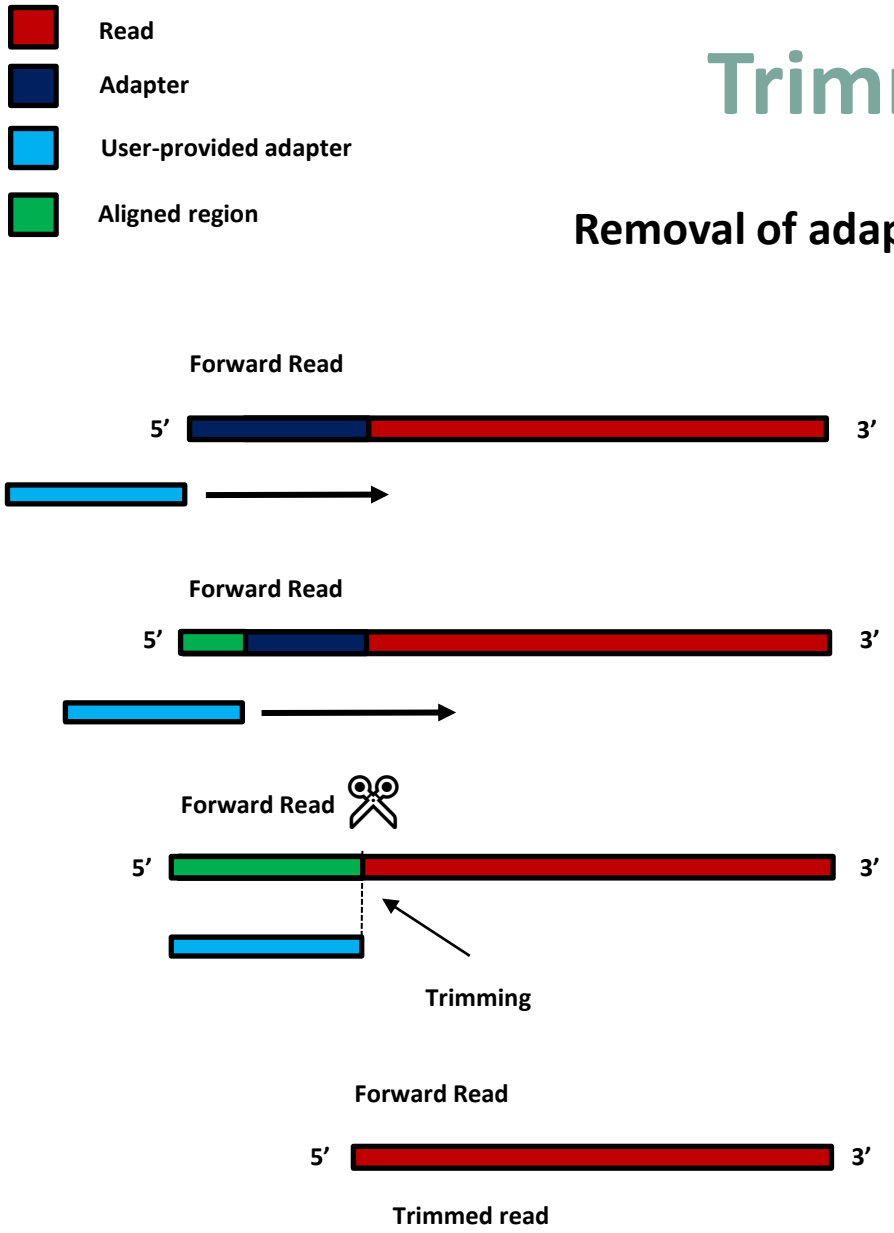
Adapter and low quality base trimming: Trimmomatic

Trimmomatic includes a variety of processing steps for read trimming and filtering, but the main algorithmic innovations are related to identification of adapter sequences and quality filtering

- Trimmomatic uses two approaches to detect technical sequences within the reads:
  - Simple mode
  - Palindrome mode

Trimmomatic: simple mode

Removal of adapters and polymerase chain reaction (PCR) primers

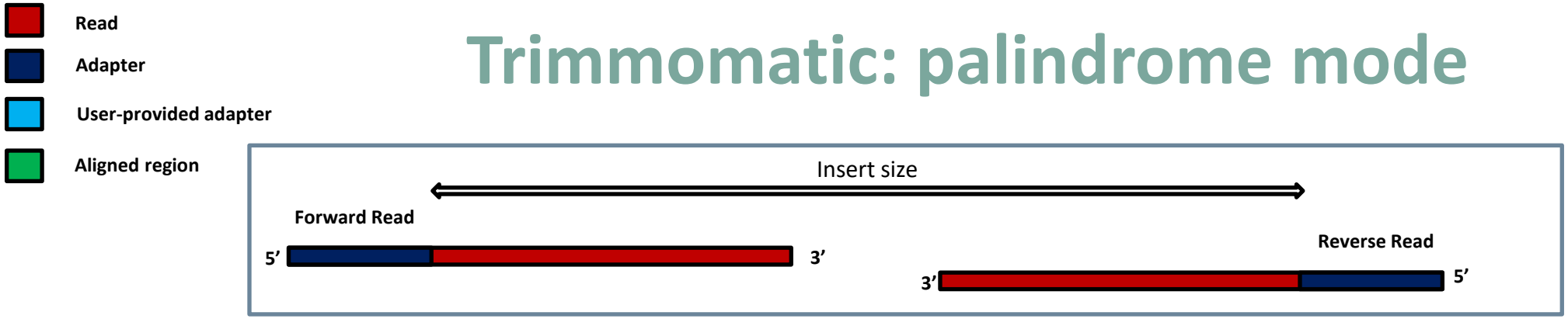


Each read is scanned from the 5' end to the 3' end to determine if any of the user-provided adapters are present

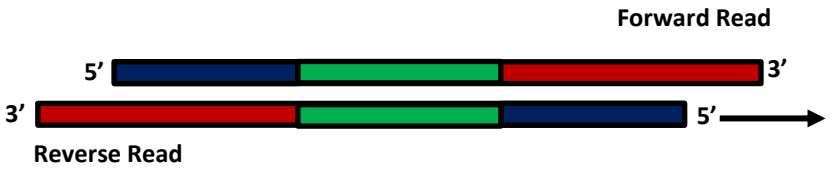
A local alignment is performed

If a sufficiently accurate match is detected, the read is clipped appropriately

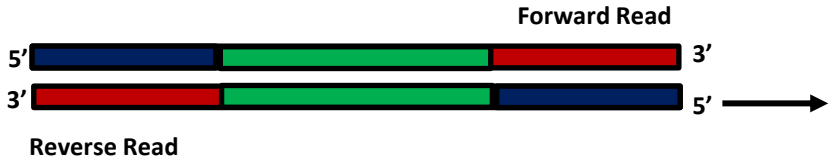
Trimmomatic: palindrome mode



In some cases, the sequenced DNA fragment is shorter than the read length, and results in adapter contamination on the end of the reads.



The alignment process begins with the adapters completely overlapping the reads

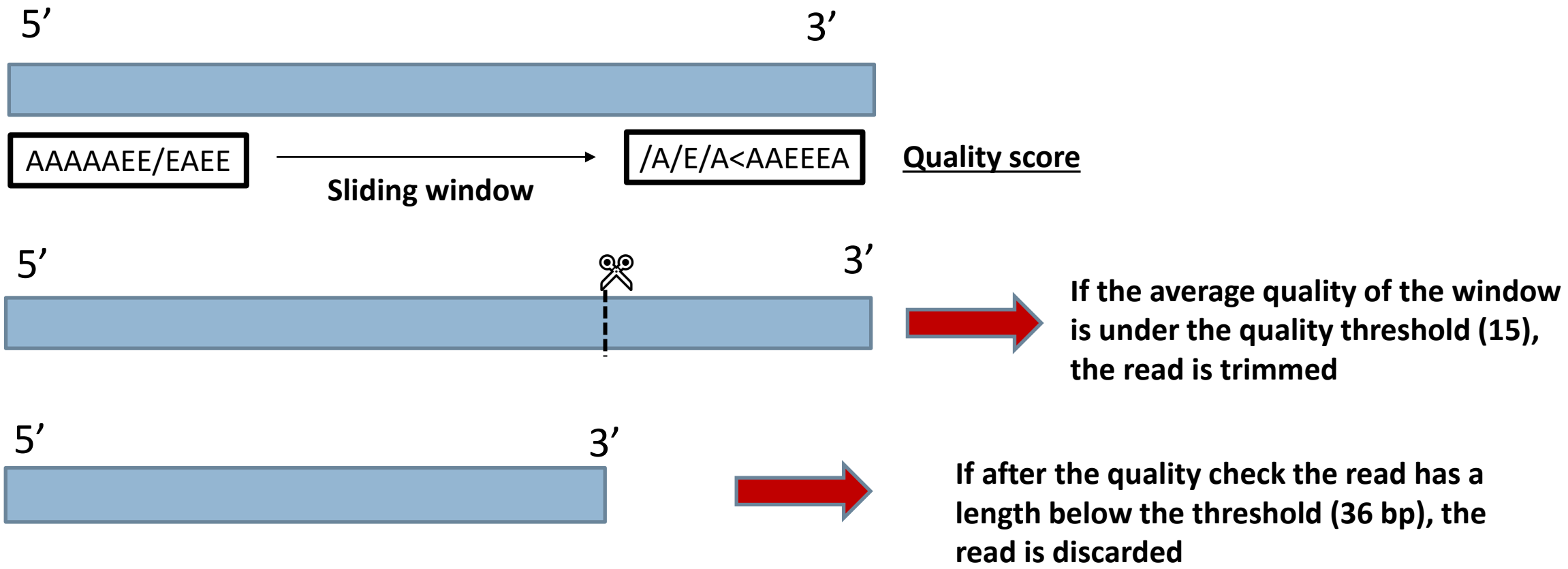


Then it proceeds by checking for later overlap

If they align in a manner which indicates 'read-through', the forward read is clipped and the reverse read dropped

Trimmomatic quality filtering

Trimmomatic performs a quality filtering which exploit the Illumina quality score of each base position to determine where the read should be cut, resulting in the retention of the 5’ portion, while the sequence on the 3’ of the cut point is discarded. It also discards reads by ‘length threshold’



Trimming command

Remove adapters from both reads and trim:

```
java -jar /opt/Trimmomatic-0.39/trimmomatic-0.39.jar PE -phred33 R1.fastq.gz R2.fastq.gz
trimmed1_trimmomatic.fastq.gz undetermined1_trimmomatic.fastq.gz
trimmed2_trimmomatic.fastq.gz undetermined2_trimmomatic.fastq.gz
ILLUMINACLIP:../ref/illumina_adapters.fa:2:30:10 LEADING:5 TRAILING:5 SLIDINGWINDOW:4:20
```

Control the quality of trimmed reads with FASTQC

```
fastqc trimmed*_trimmomatic.fastq.gz -o fastqc
```

Download the Fastqc files (trimmed reads)

On the server, we don't have a graphical visualization, therefore:

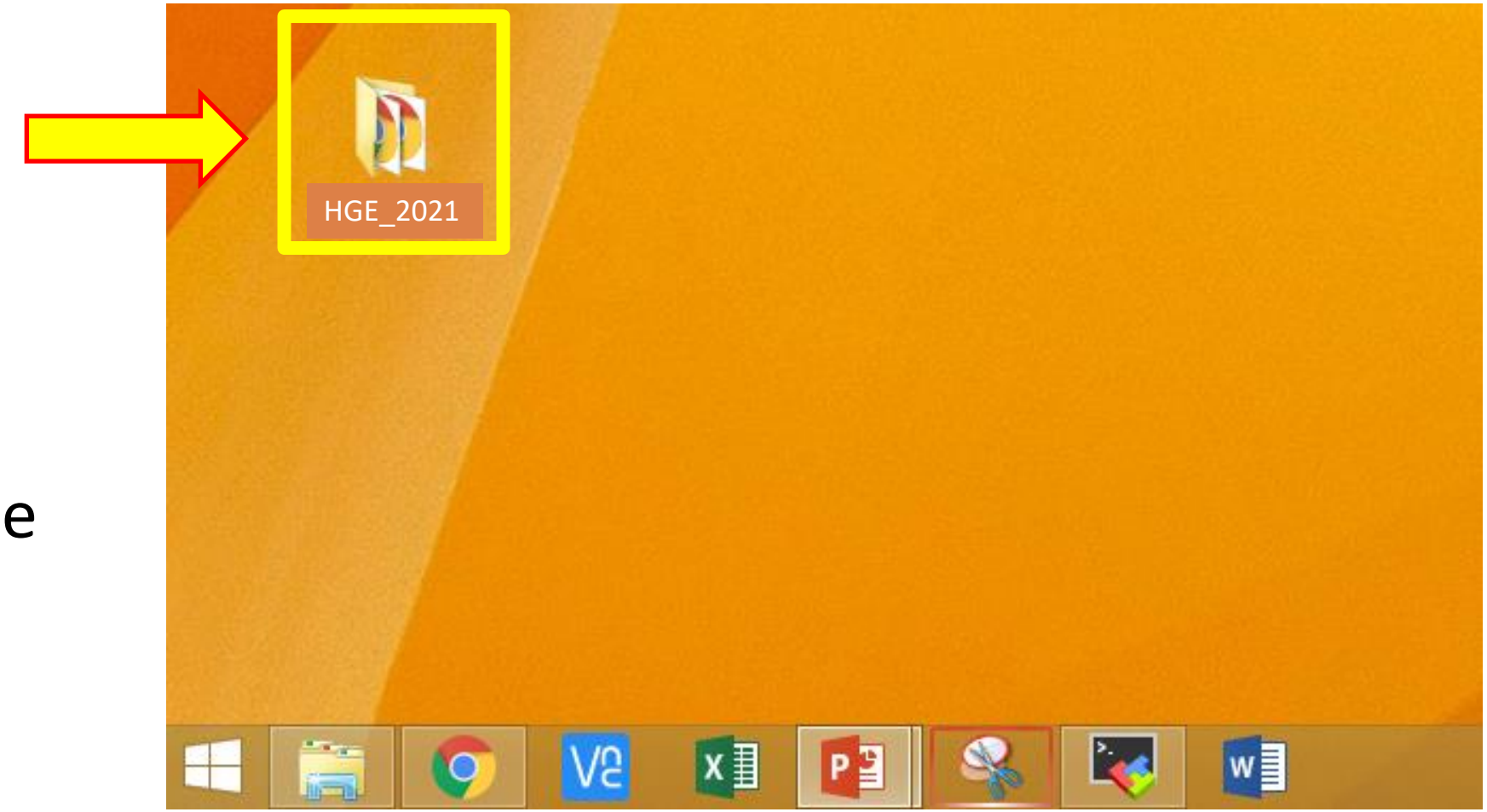
1. Open a new terminal
2. Enter in the folder on your PC: cd Desktop/HGE_2021
3. Download the results from here:

   rsync -auv lessons@157.27.80.26:/home/lessons/HGE_2021/denise/fastqc/trimmed*_fastqc.html .

   Pass: lez2021
4. Check what you have downloaded: ls
5. Close the shell

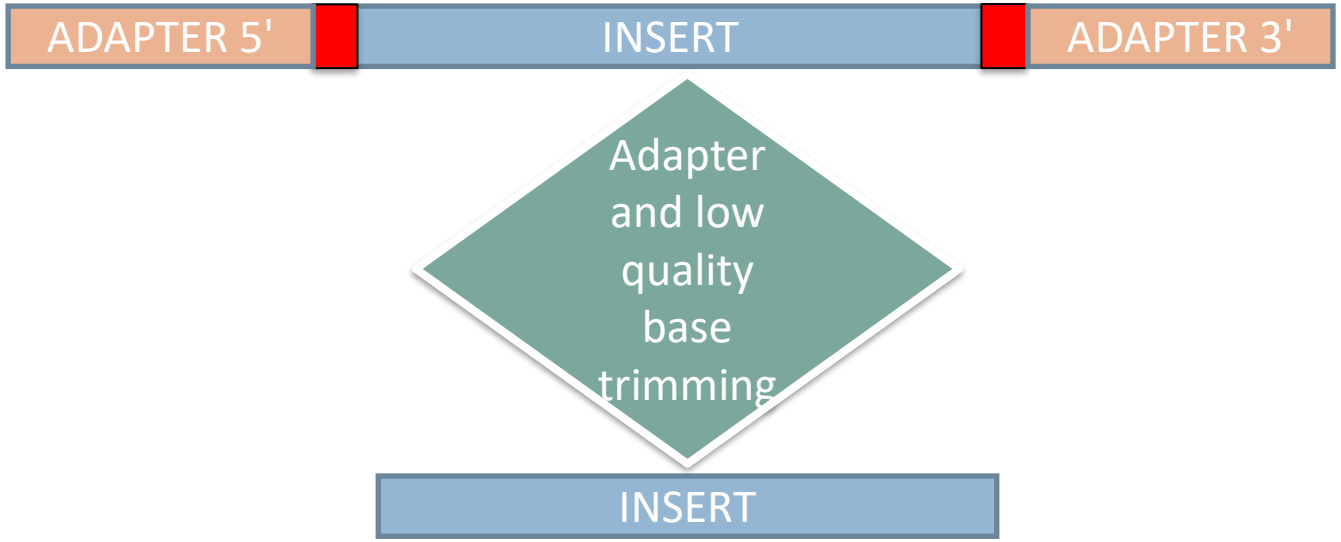
Open the Fastqc files (trimmed reads)

1. On your desktop, open the folder «HGE_2021»



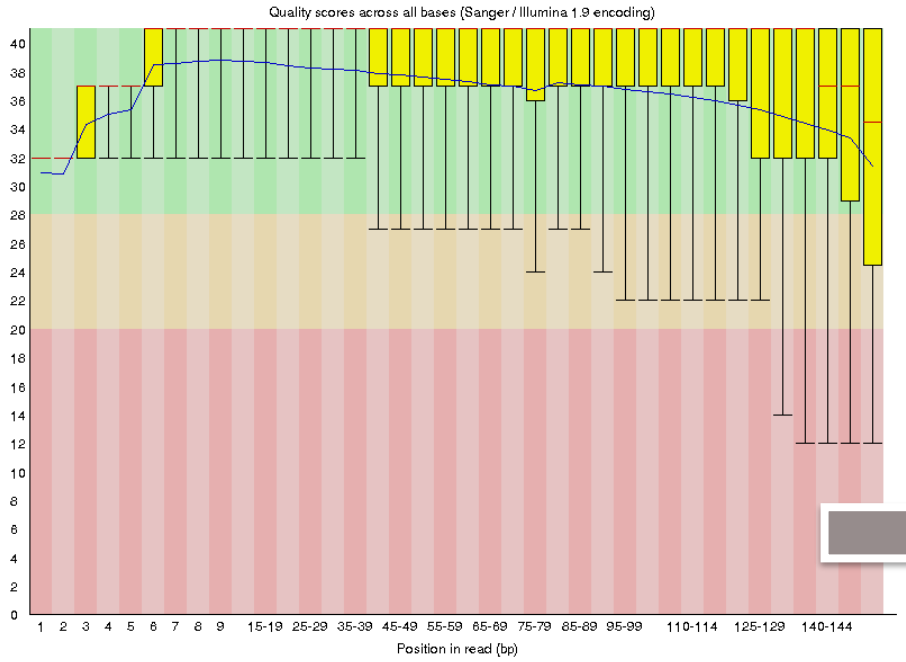
2. Open the html file

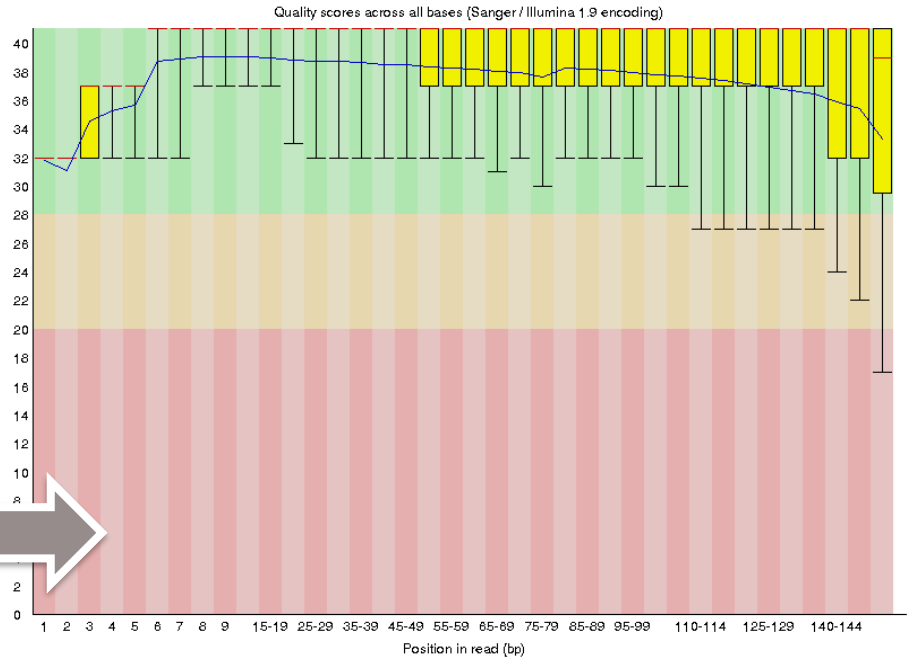
FastQC of trimmed reads



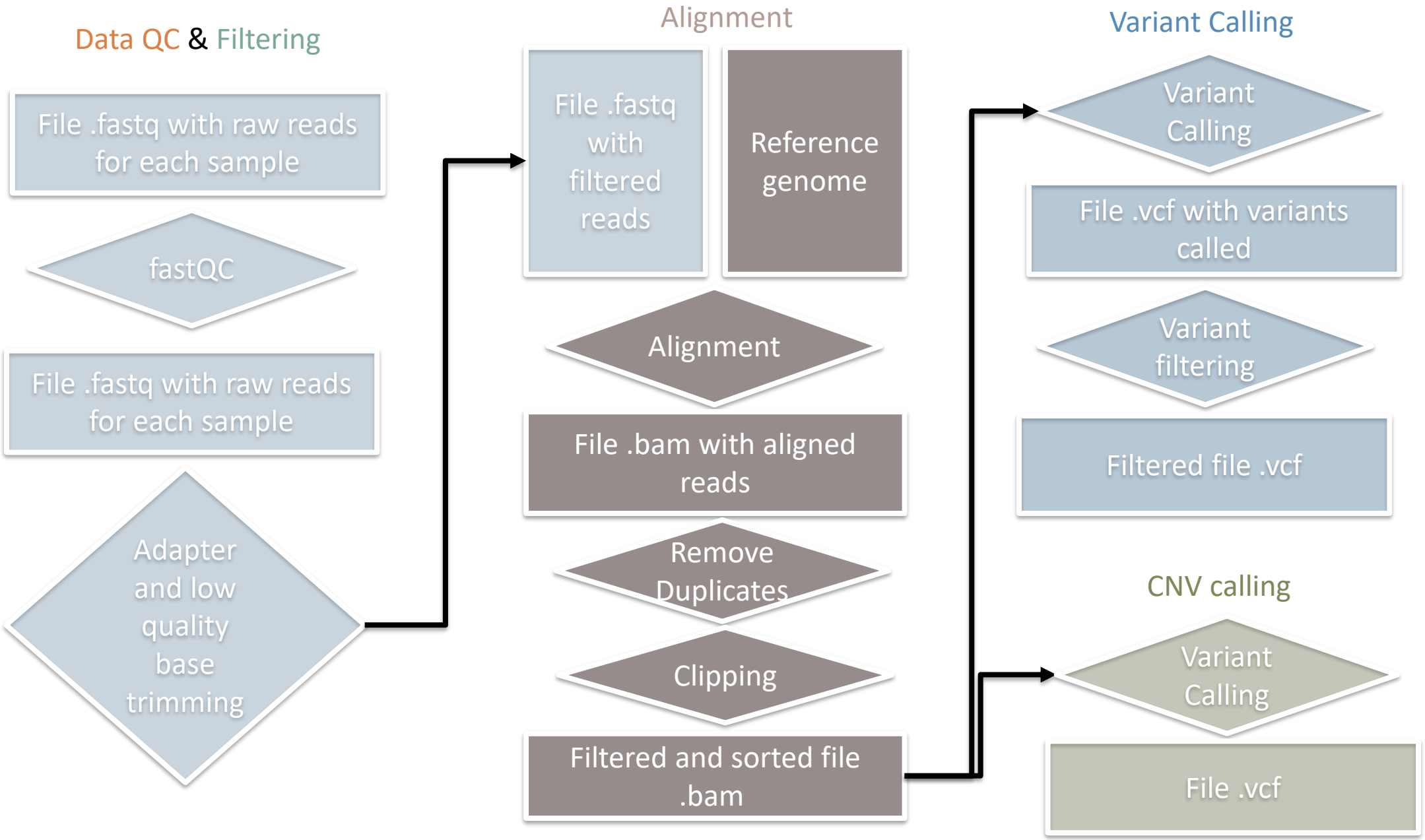
Low quality bases

Input: RAW fastQ read without adapters





Pipeline


Data QC & Filtering
File .fastq with raw reads for each sample
fastQC
File .fastq with raw reads for each sample
Adapter and low quality base trimming
Alignment
File .fastq with filtered reads
Reference genome
Alignment
File .bam with aligned reads
Remove Duplicates
Clipping
Filtered and sorted file .bam
Variant Calling
Variant Calling
File .vcf with variants called
Variant filtering
Filtered file .vcf
CNV calling
Variant Calling
File .vcf

Alignment

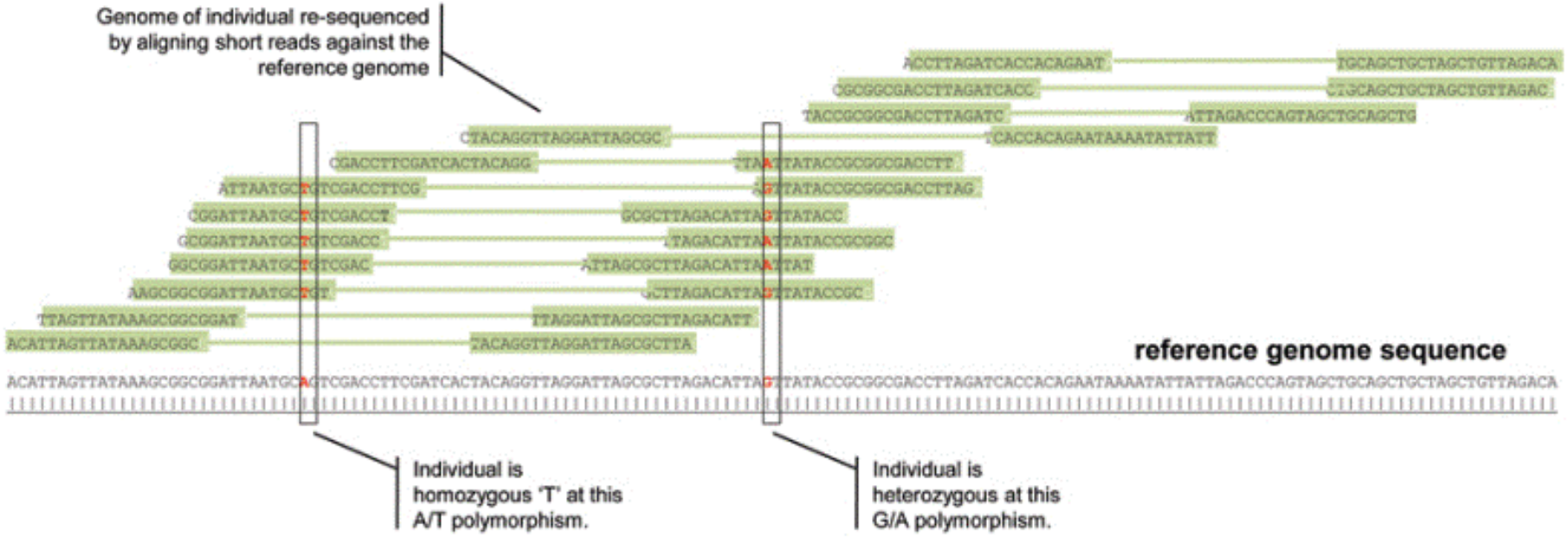
Once high-quality data are obtained from pre-processing, the next step is the read mapping or alignment. Many of the next-generation sequencing projects begin with a known, or so-called ‘reference’, genome. In this case, to make sense of the reads, their positions within the reference sequence must be determined. This process is known as aligning or ‘mapping’ the read to the reference.

Computationally difficult

- Short Reads
- Lots of repeats
- Presence of mismatch

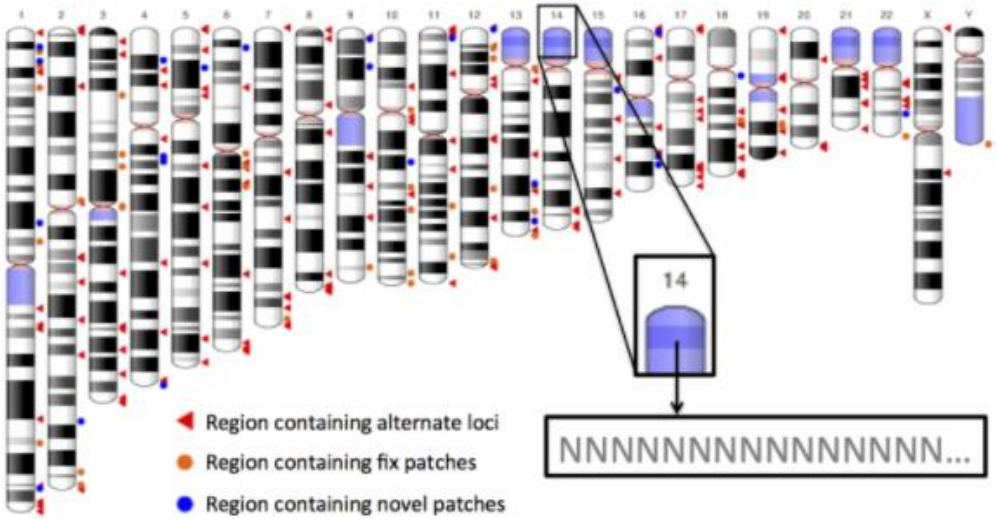
Different algorithm solution:

- Bowtie
- BWA
- ISAAC Aligner



Reference Genome

Latest release of the Human Genome is hg38 (GRCh38).



GRCh37:

Total bases:
- 3.23 Billion
- 2.99 Billion (without N)

N50:
- 46 Million

Number of alternative loci:
- 9

Non-nuclear genome:
- No

GRCh38.p12:

Total bases:
- 3.24 Billion
- 3.08 Billion (without N)

N50:
- 68 Million

Number of alternative loci :
- 261

Non-nuclear genome:
- Yes

GRCh38.p13 Release date 2019-03-01
- 185 Patches (113 Fix, 72 Novel): 38 Mb
- 178 Alternate loci: 129 Mb

To visualize the human genome reference FASTA content (chr6 example):

less -S ../ref/chr6.hg38.fa

SAM/BAM file

SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments. BAM is the compressed binary version of the SAM format that represents the standard format for sequence alignment.

HEADER

```
@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       * 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
```

(UN)ALIGNED READS

Columns:

1. **Read name**
2. **Flag**
3. **Reference sequence name**
4. **1- based leftmost mapping position**
5. **Mapping quality (MAPQ)**
5. **Cigar string**