Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

UNIVERSITY OF CAMBRIDGE

Next Generation Sequencing data analysis

27/11/2018

Pierpaolo Maisano Delser
mail: maisanop@tcd.ie ; pm604@cam.ac.uk

Outline:

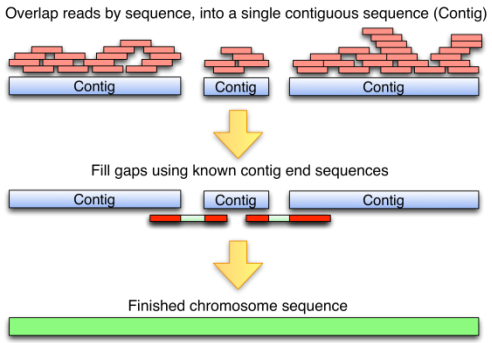
- Example of a pipeline to analyse NGS data;
- Overview of file formats and tools;
- Examples of filtering and validation

NGS pipeline(s)

sequencing

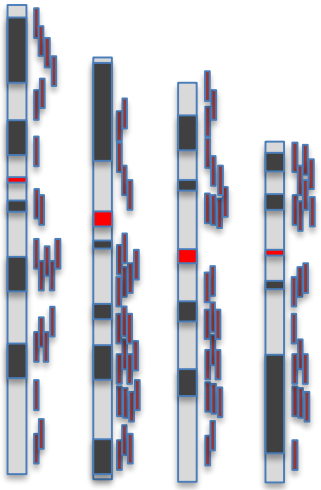
unaligned reads QC

reads trimming



de-novo assembly

mapping to a reference genome



mapping refinement

assembly QC

mapping QC

DNA-seq	whole genome	capture: exome/custom/cancer	amplicon sequencing
RNA-seq	whole transcriptome	amplicon sequencing: fixed/custom	

Filtering

Validation

an example of NGS data analysis

- **project**

an example of NGS data analysis

- **project**

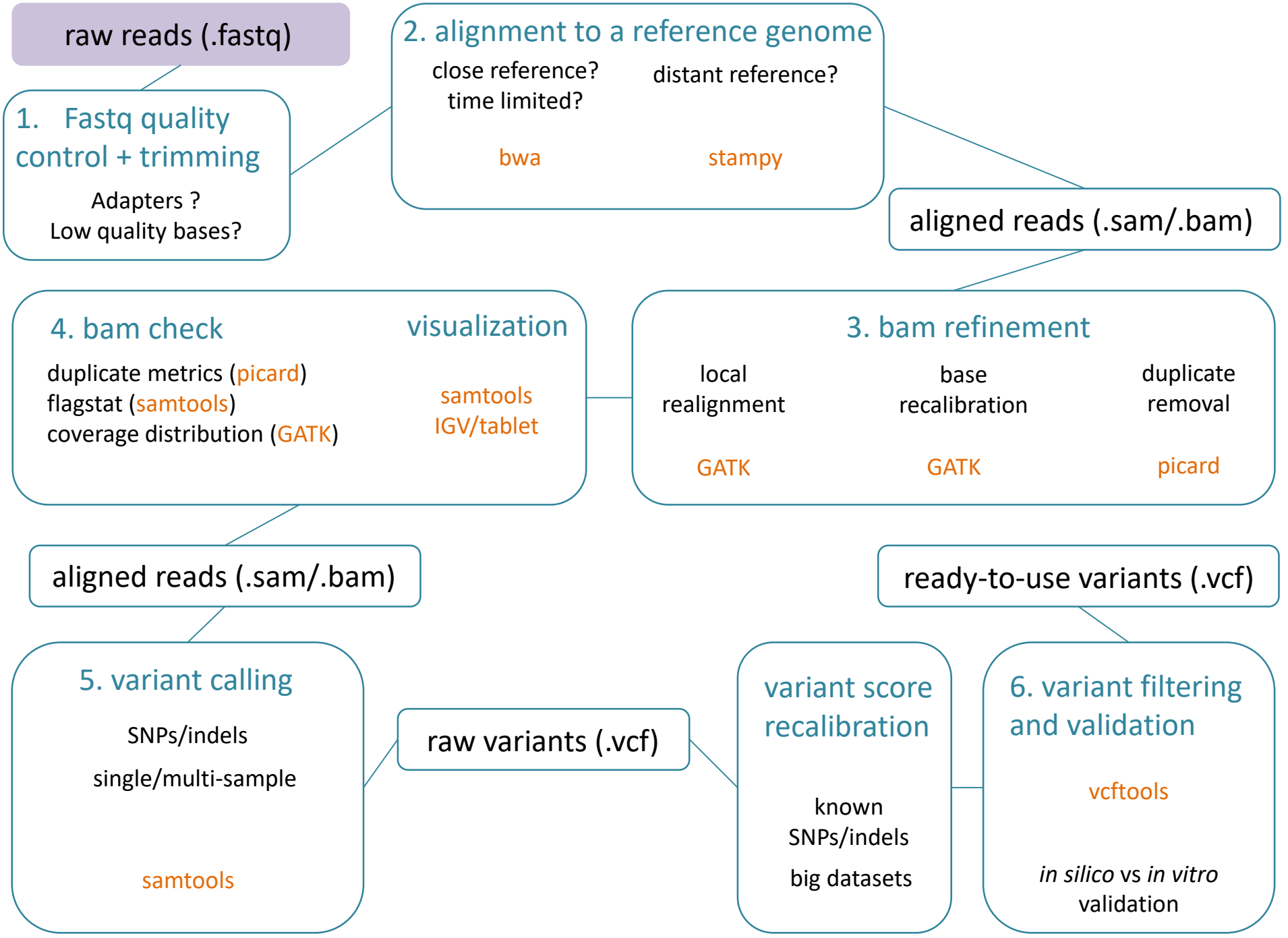
- **project:application** (whole genomes? Exomes? Target capture? Amplicon sequencing?)

an example of NGS data analysis

- **project**
- **project:application** (whole genomes? Exomes? Target capture? Amplicon sequencing?)
- **project:application:aim** (SNPs, indels, repeated elements, CNVs…)

an example of NGS data analysis

- **project**
- **project:application** (whole genomes? Exomes? Target capture? Amplicon sequencing?)
- **project:application:aim** (SNPs, indels, repeated elements, CNVs…)
- **project:application:aim:coverage**(SNPs, indels, repeated elements, CNVs…)

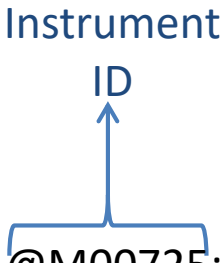

raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

.fa/.fasta	sequences
.fastq	read data
.sam (.sai)	mapped reads
.bam (.bai)	mapped reads (binary)
.vcf	variant information

raw reads (.fastq)

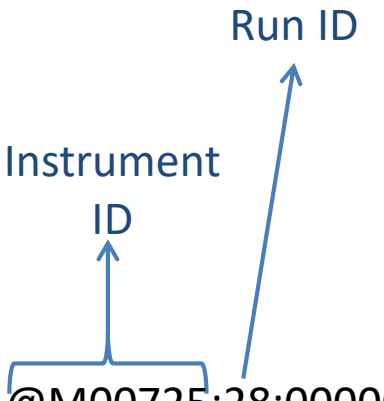
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



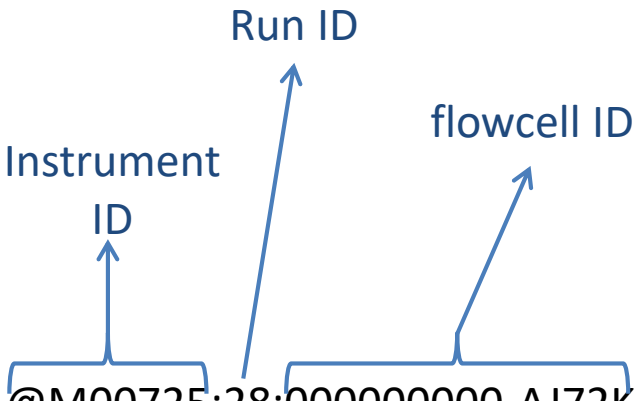
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



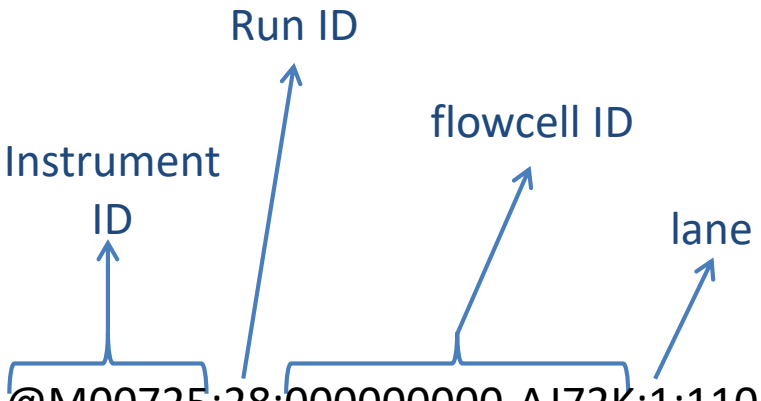
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



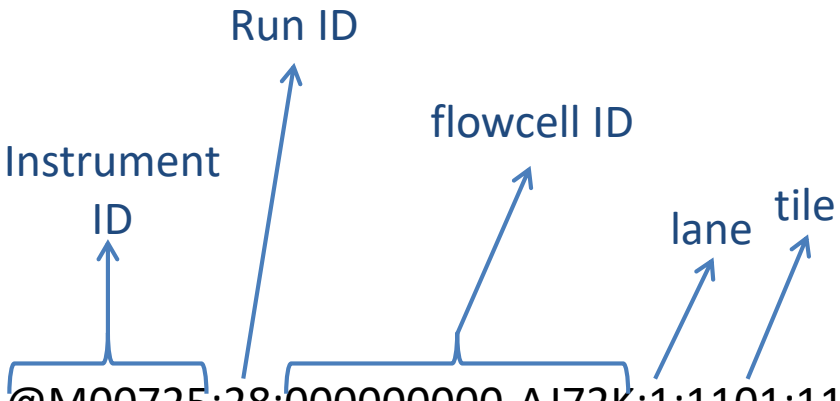
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



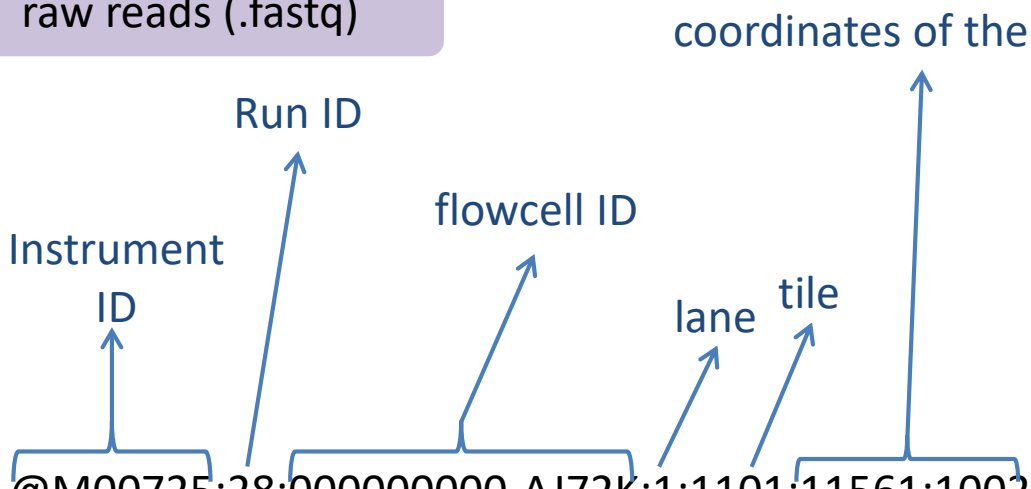
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



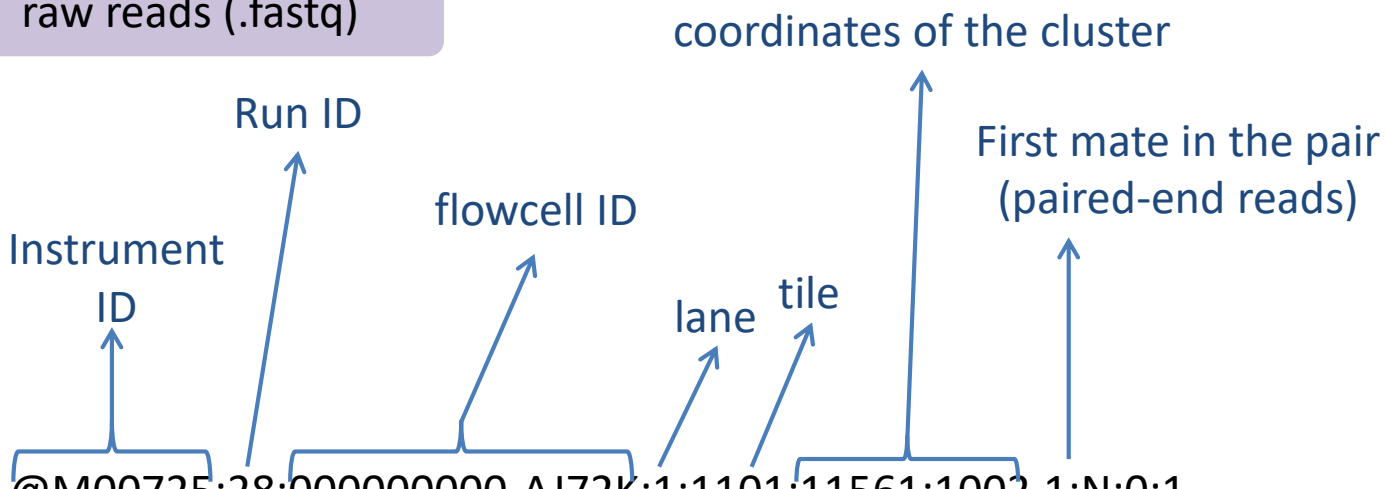
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



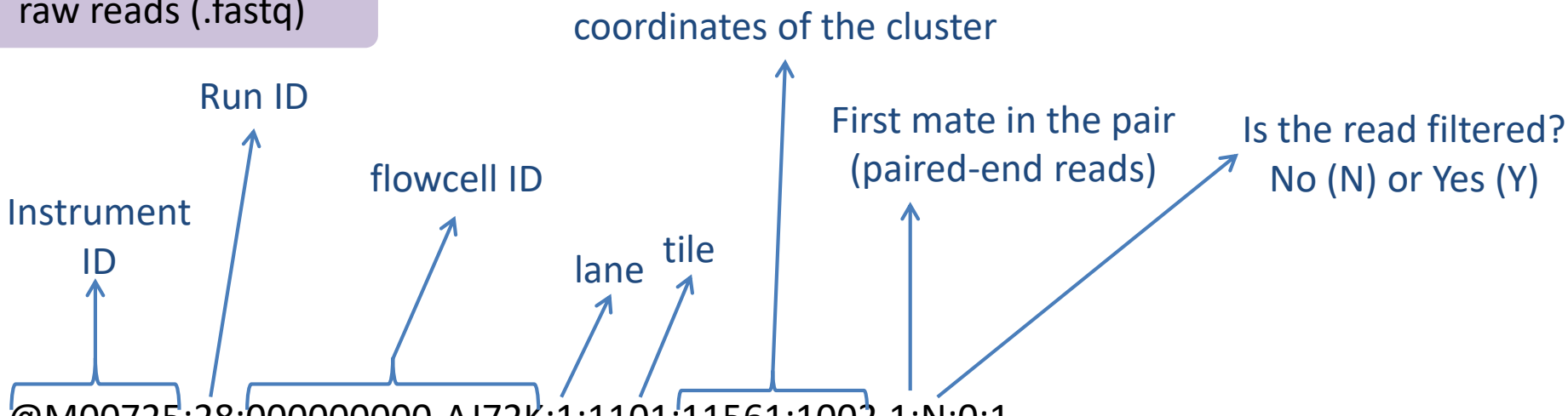
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



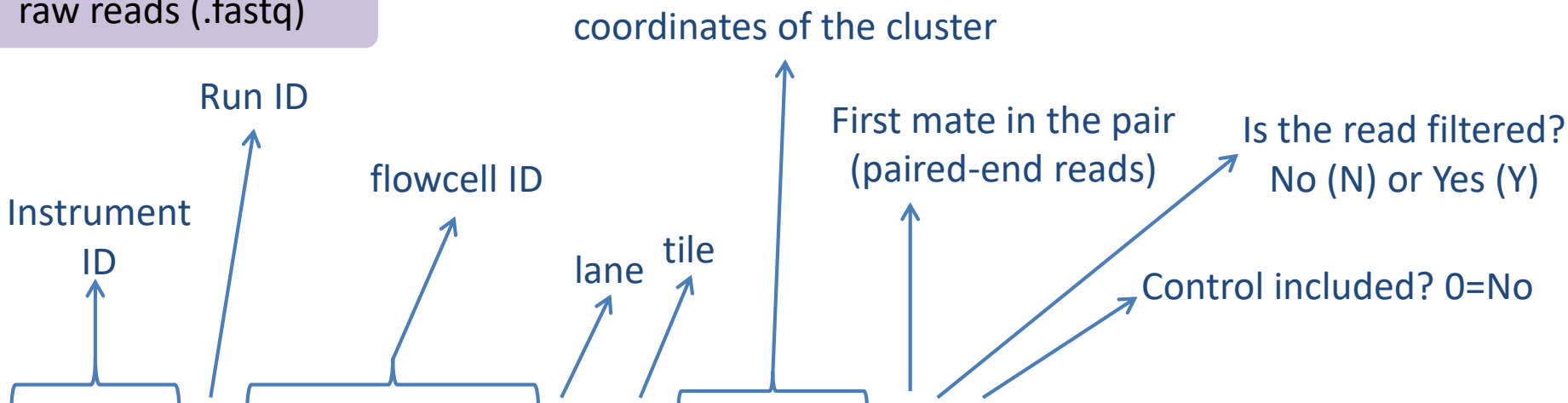
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



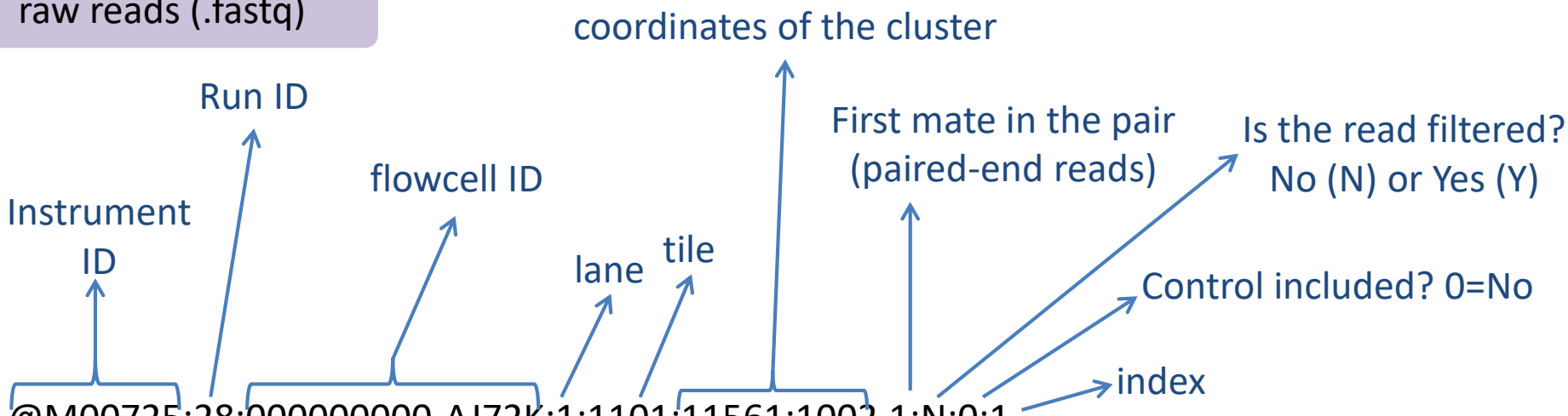
@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1

AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA

TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN

GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN

GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN

NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG

+

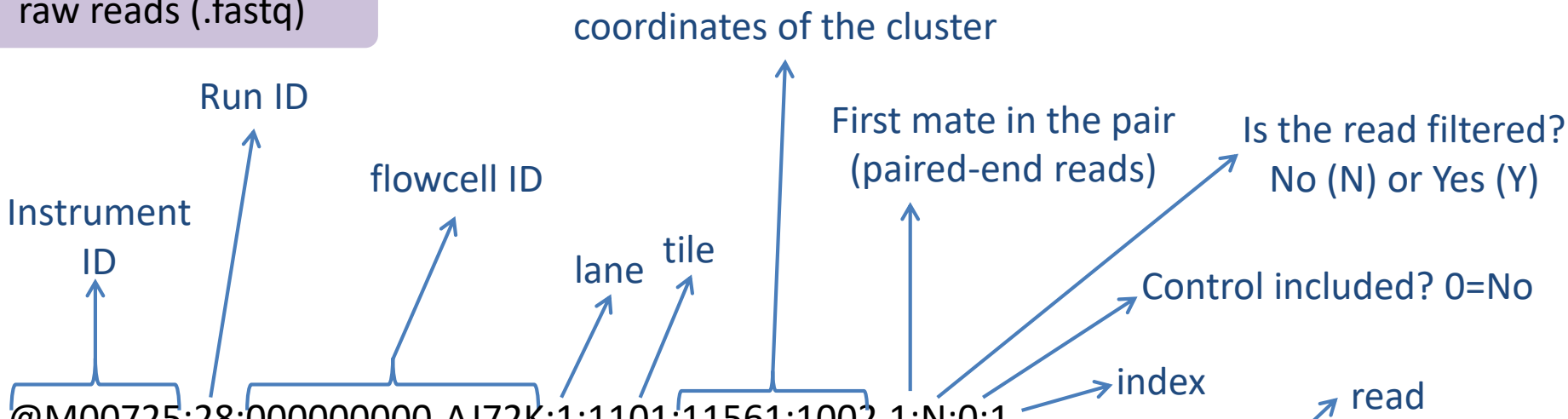
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@

FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF

FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*

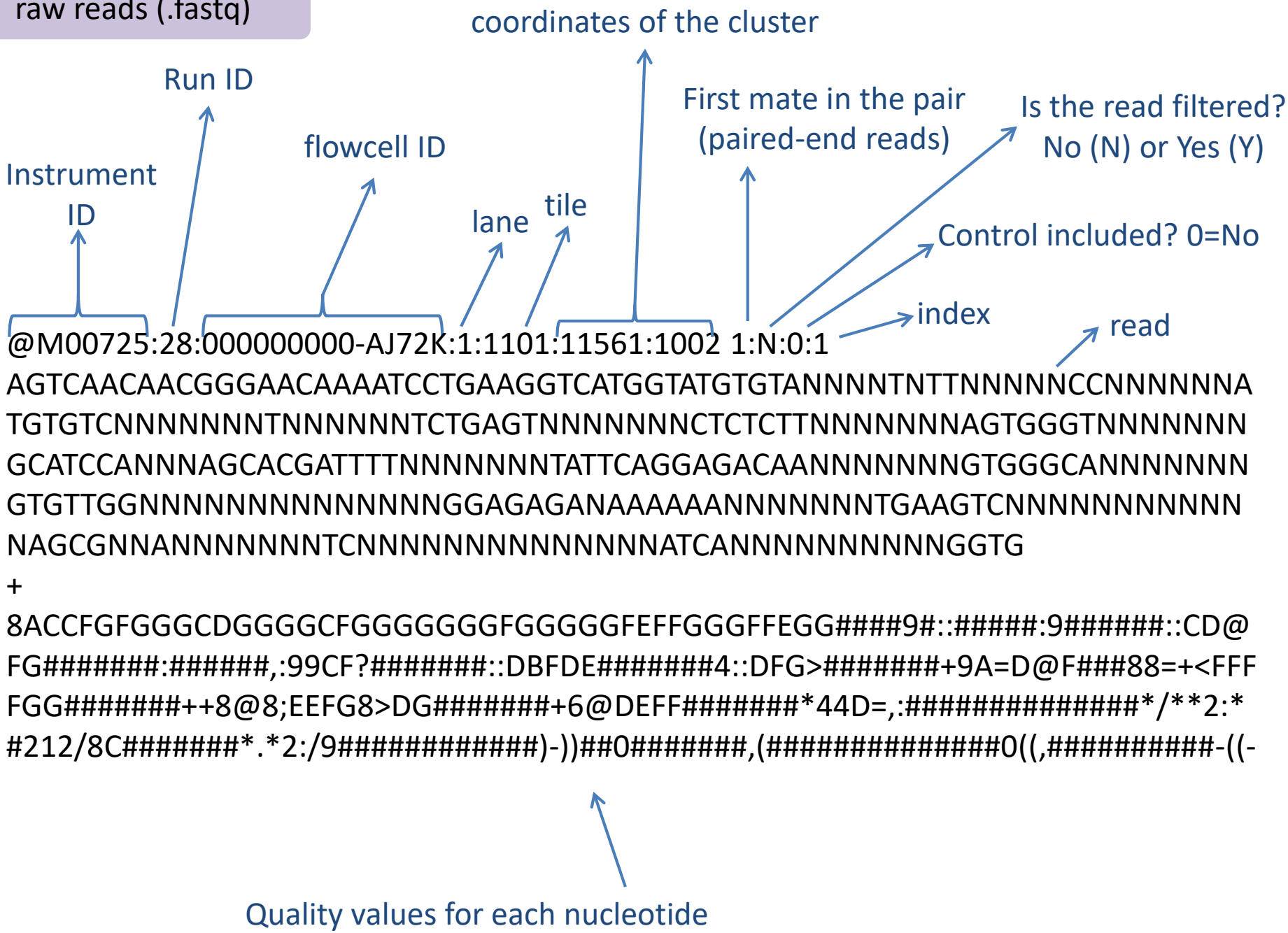
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

raw reads (.fastq)



raw reads (.fastq)

@M00725:28:000000000-AJ72K:1:1101:11561:1002 1:N:0:1
AGTCAACAACGGGAACAAAATCCTGAAGGTCATGGTATGTGTANNNNTNTTNNNNNCCNNNNNNA
TGTGTCNNNNNNNTNNNNNNTCTGAGTNNNNNNNCTCTCTTNNNNNNNAGTGGGTNNNNNNN
GCATCCANNNAGCACGATTTTNNNNNNNTATTCAGGAGACAANNNNNNNGTGGGCANNNNNNN
GTGTTGGNNNNNNNNNNNNNNGGAGAGANAAAAAANNNNNNNTGAAGTCNNNNNNNNNNN
NAGCGNNANNNNNNNTCNNNNNNNNNNNNNNATCANNNNNNNNNGGTG
+
8ACCFGFGGGCDGGGGCFGGGGGGGFGGGGGFEFFGGGFFEGG####9#::#####:9######::CD@
FG#######:######,:99CF?#######::DBFDE#######4::DFG>#######+9A=D@F###88=+<FFF
FGG#######++8@8;EEFG8>DG#######+6@DEFF#######*44D=,:###############*/**2:*
#212/8C#######*.*2:/9############)-))##0#######,(###############0((,##########-((-

Quality values for each nucleotide

Lowest	ASCII	Highest
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~		
33		126

0.2.....26...31.....41

Illumina 1.8+ Phred+33, raw reads typically (0, 41)

raw reads (.fastq)

Lowest ASCII Highest

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

33 126

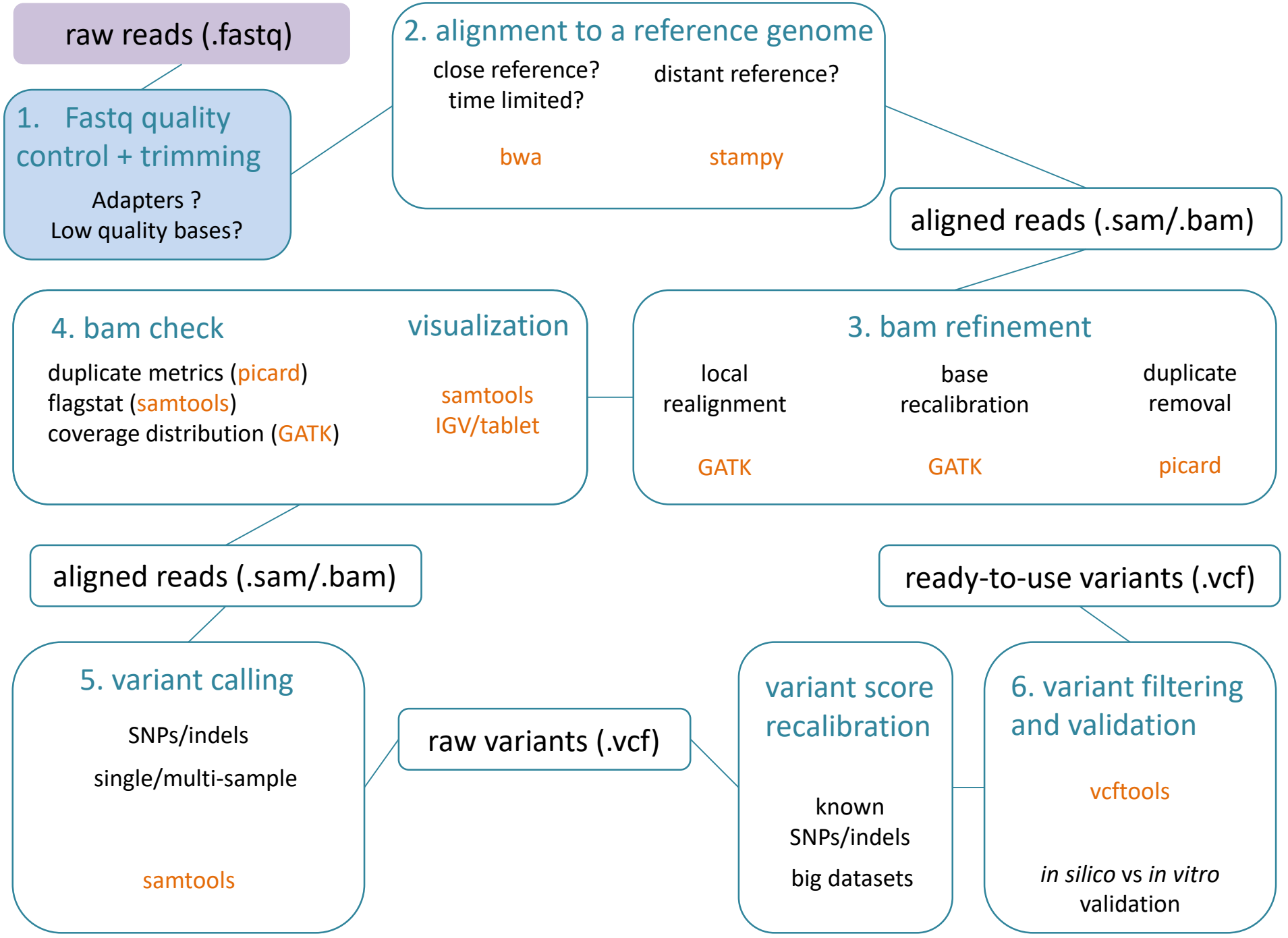
```
0.2.....................26...31........41
```

```
Illumina 1.8+ Phred+33, raw reads typically (0, 41)
```

Phred-scale value:

$$Q = -10*\log_{10}P \quad \rightarrow \quad P = 10^{-Q/10}$$

Phred Quality Score (Q)	Probability of incorrect base call (P)	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%
50	1 in 100000	99.999%


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

Fastqc: quality control of the raw data coming out from the sequencer

Fastqc: quality control of the raw data coming out from the sequencer

- Evaluation of the quality of the generated data;

Fastqc: quality control of the raw data coming out from the sequencer

- Evaluation of the quality of the generated data;
- Basic summary statistics of the raw data;

Fastqc: quality control of the raw data coming out from the sequencer

- Evaluation of the quality of the generated data;
- Basic summary statistics of the raw data;
- Several modules to evaluate different features (i.e. adapters; base quality, etc…)

Fastqc: quality control of the raw data coming out from the sequencer

- Evaluation of the quality of the generated data;
- Basic summary statistics of the raw data;
- Several modules to evaluate different features (i.e. adapters; base quality, etc…)
- Feedback (green, orange, red): do not fully rely on that, think what does it mean!!

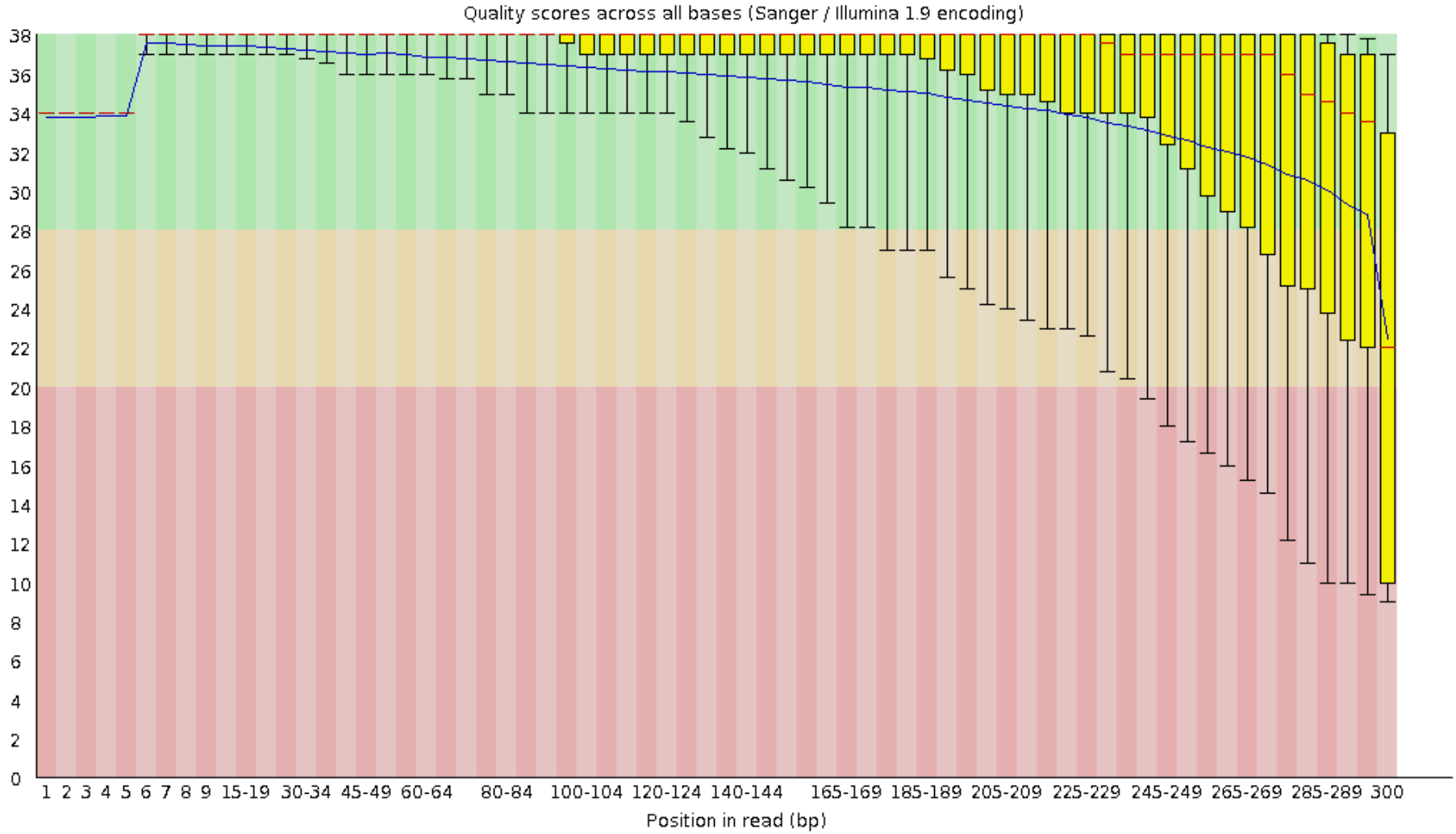
1- Fastq quality control + trimming

Basic Statistics

Measure	Value
Filename	GN12738_original_galeus_L001_R1_001.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	1266231
Sequences flagged as poor quality	0
Sequence length	300
%GC	45

1- Fastq quality control + trimming

Per base sequence quality: **warning**



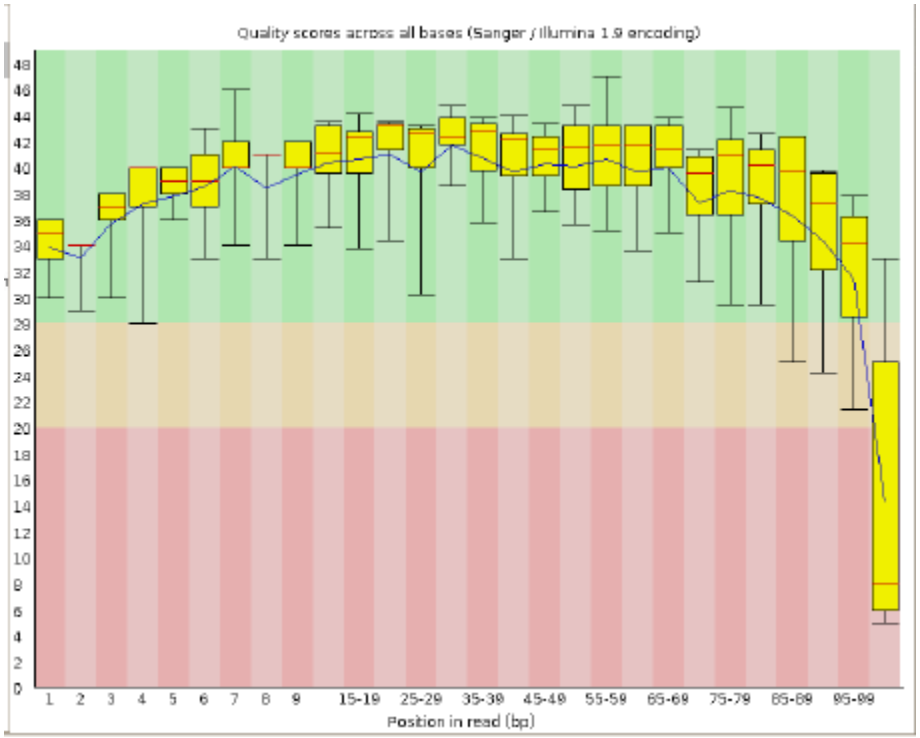
What can we do to improve the quality at the end of the reads?

Read Trimming: removal of lower-quality 3' Ends with Low Quality Scores

1- Fastq quality control + trimming

What can we do to improve the quality at the end of the reads?

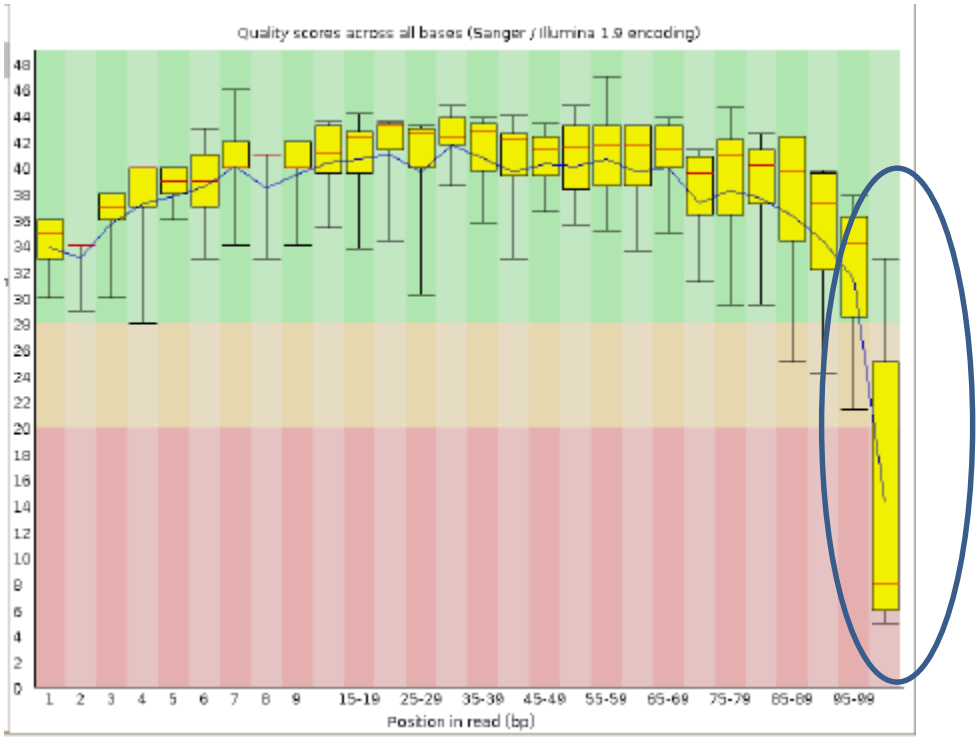
Read Trimming: removal of lower-quality 3' Ends with Low Quality Scores



1- Fastq quality control + trimming

What can we do to improve the quality at the end of the reads?

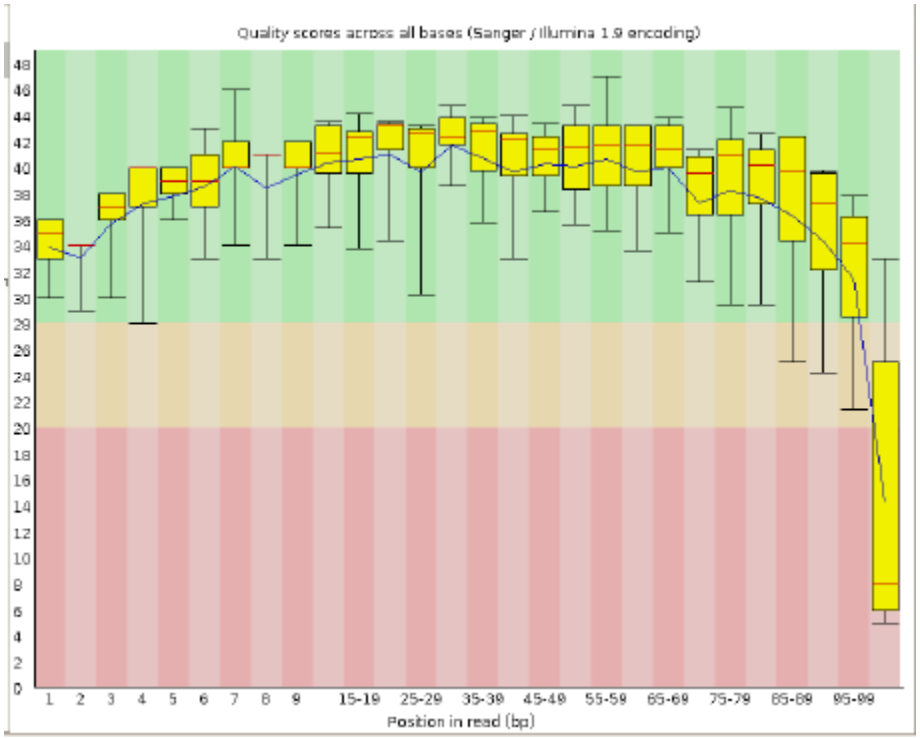
Read Trimming: removal of lower-quality 3' Ends with Low Quality Scores

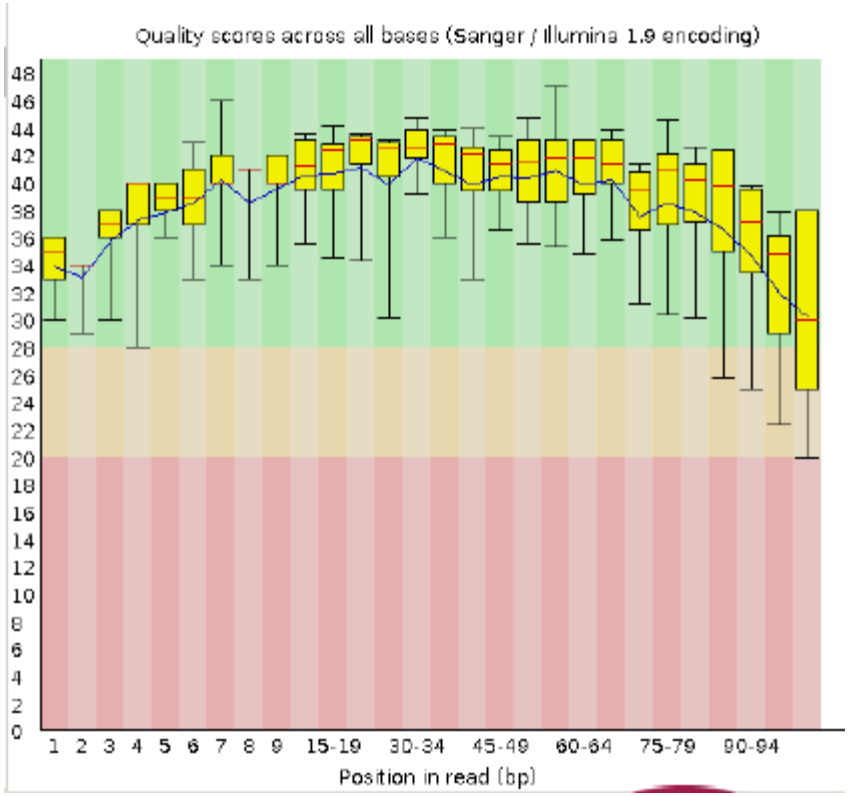


1- Fastq quality control + trimming

What can we do to improve the quality at the end of the reads?

Read Trimming: removal of lower-quality 3' Ends with Low Quality Scores

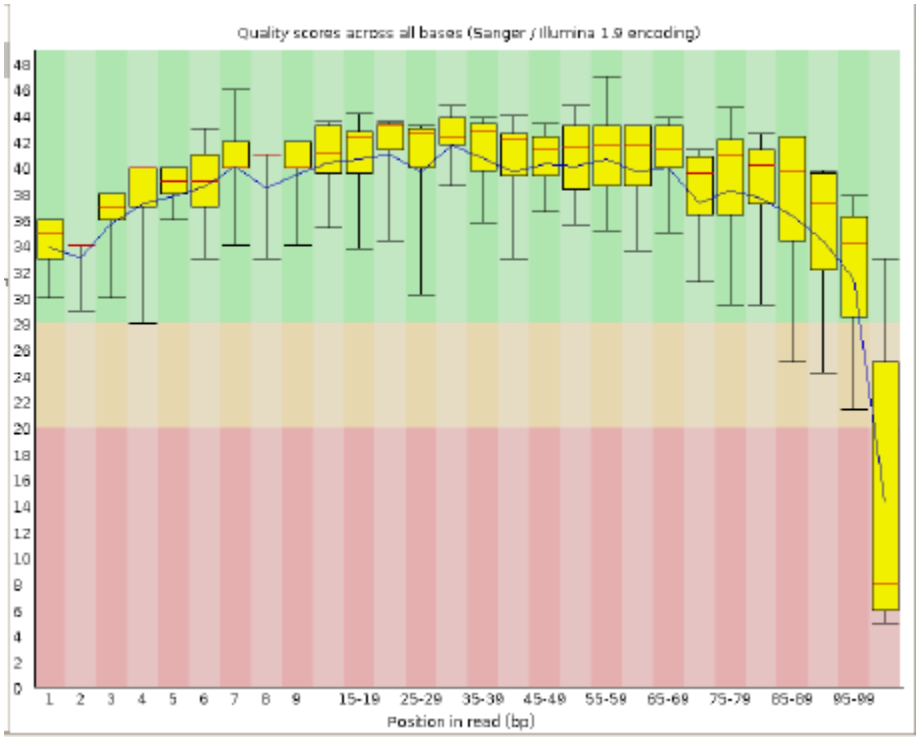




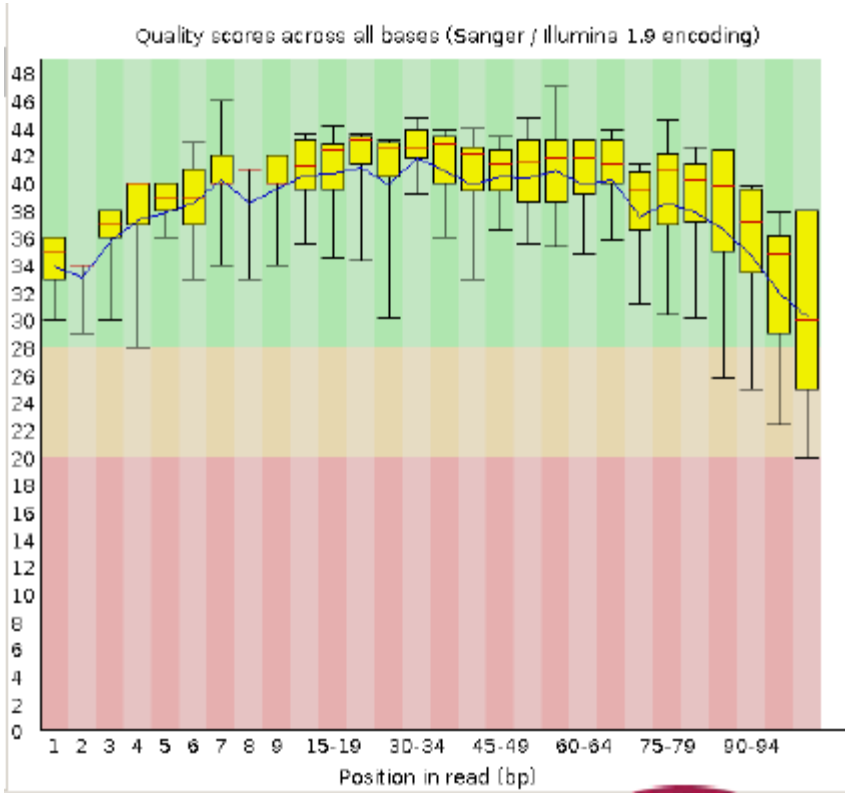
1- Fastq quality control + trimming

What can we do to improve the quality at the end of the reads?

Read Trimming: removal of lower-quality 3' Ends with Low Quality Scores



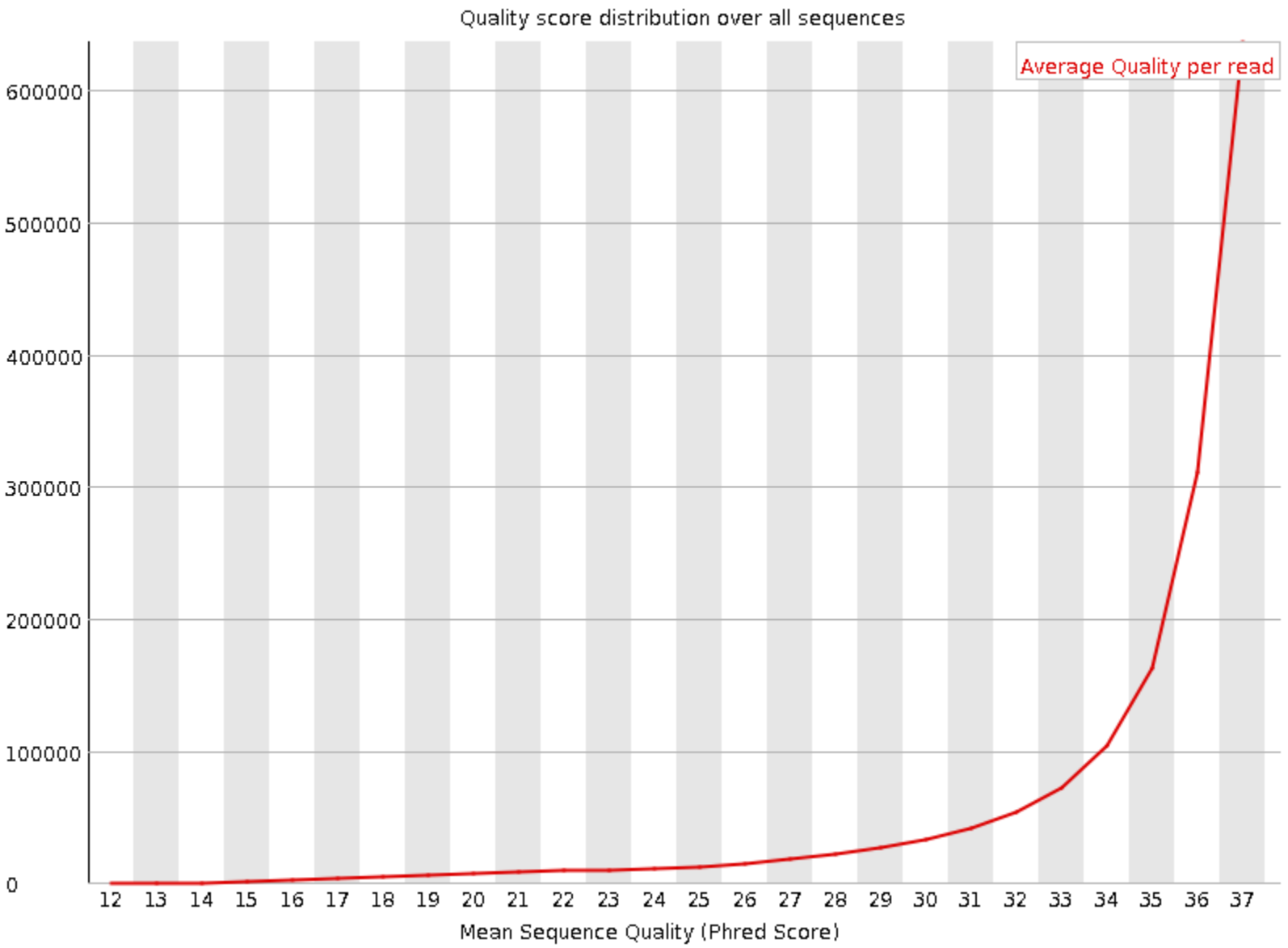
95-99 bp



90-94 bp

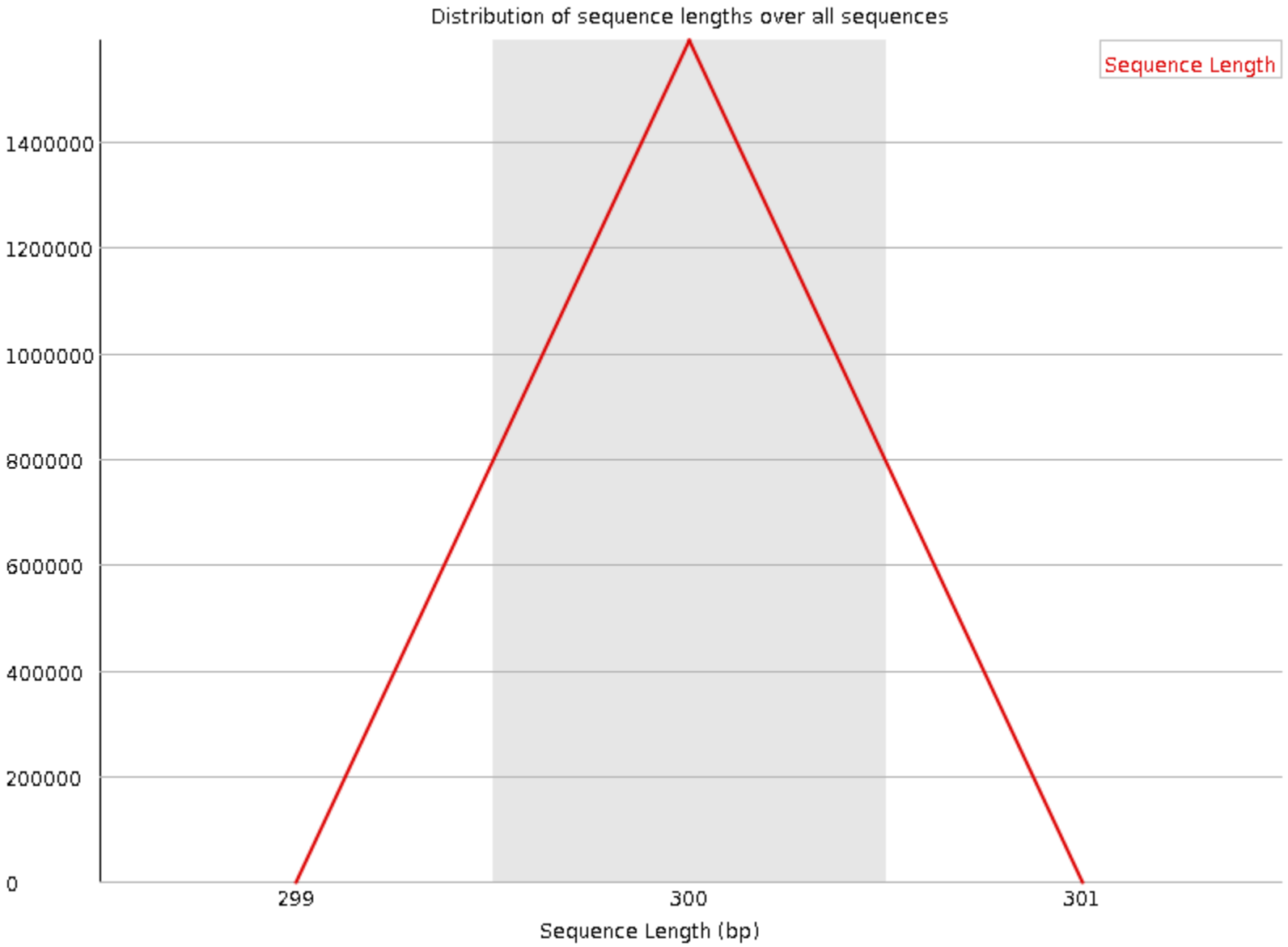
1- Fastq quality control + trimming

Per sequence quality score: **pass**



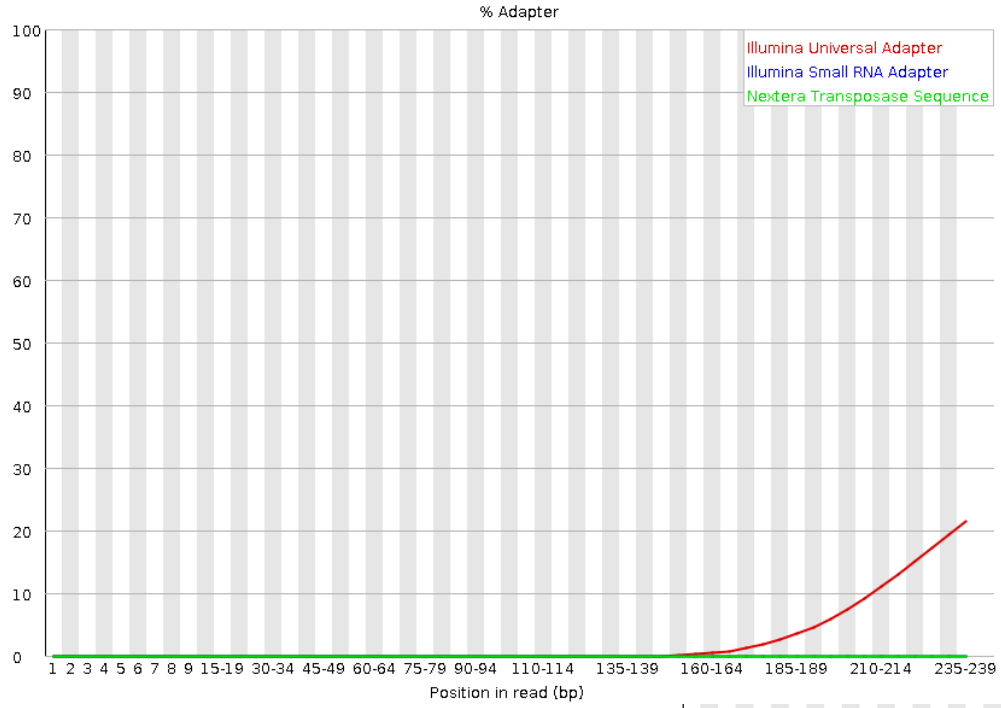
1- Fastq quality control + trimming

Sequence length: **pass**

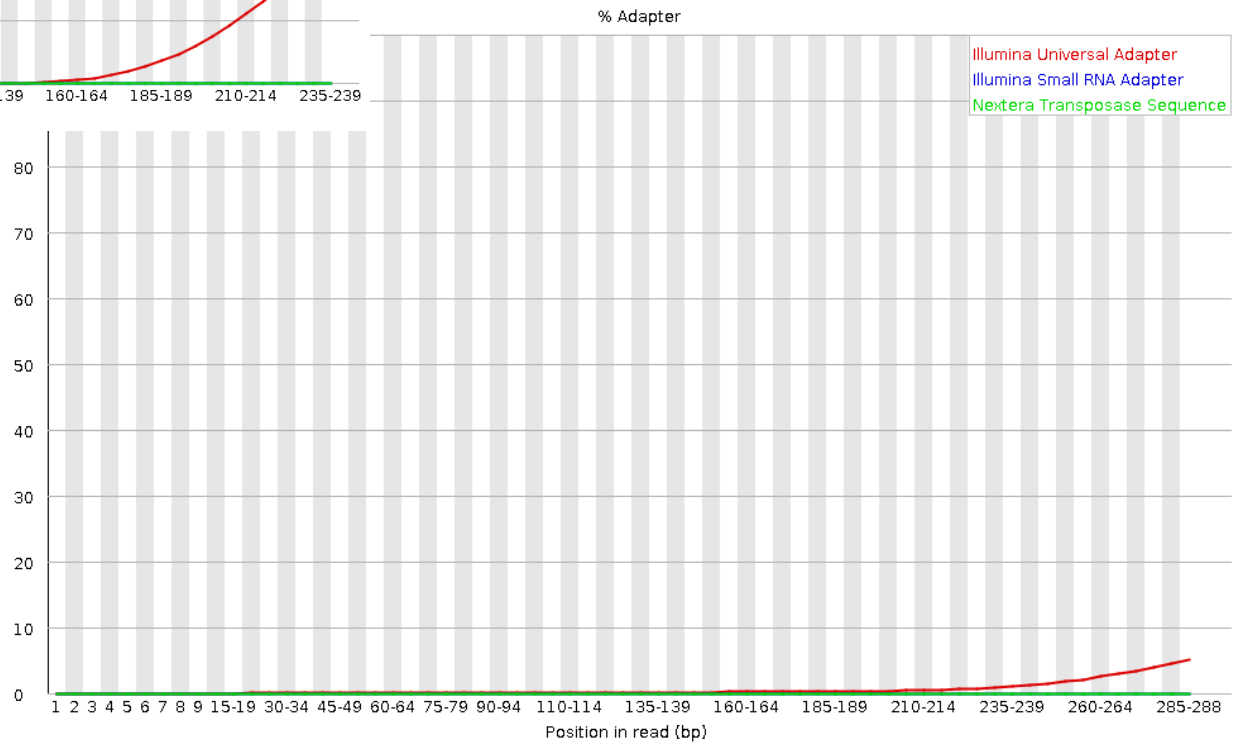


1- Fastq quality control + trimming

Adapters removal



Failed

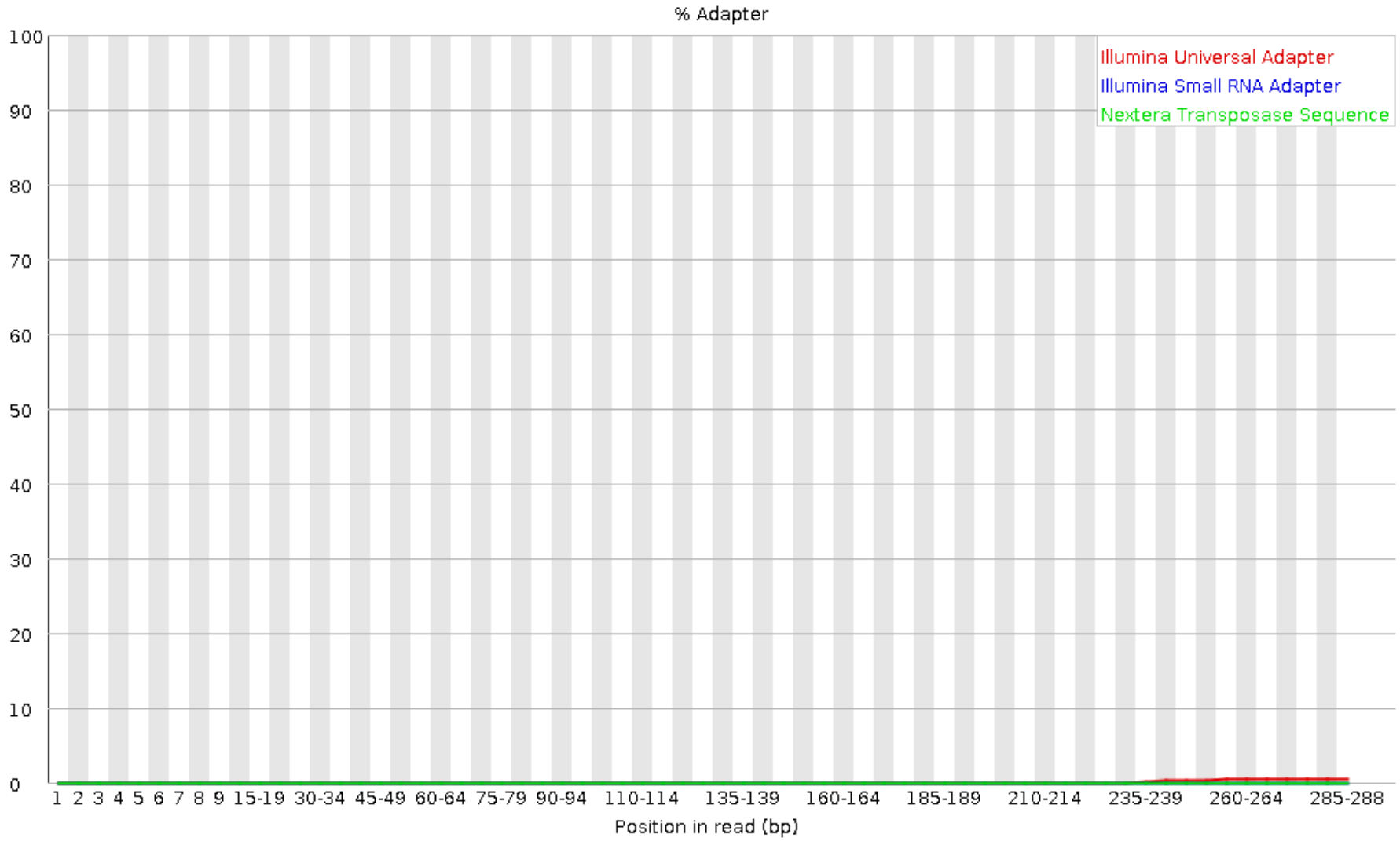


Warning

1- Fastq quality control + trimming

Adapters removal

Pass



Overrepresented sequences

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CGGAAGAGCGTTCAGCAGGAATGCCGAGACCGATCTCCGATGTTTATCTC	461	0.23194618446010878	Illumina Paired End PCR Primer 2 (96% over 29bp)
GGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCCGATGTTTATCTC	213	0.10716819368764244	Illumina Paired End PCR Primer 2 (97% over 37bp)

Removal of overrepresented sequences (PCR primers).

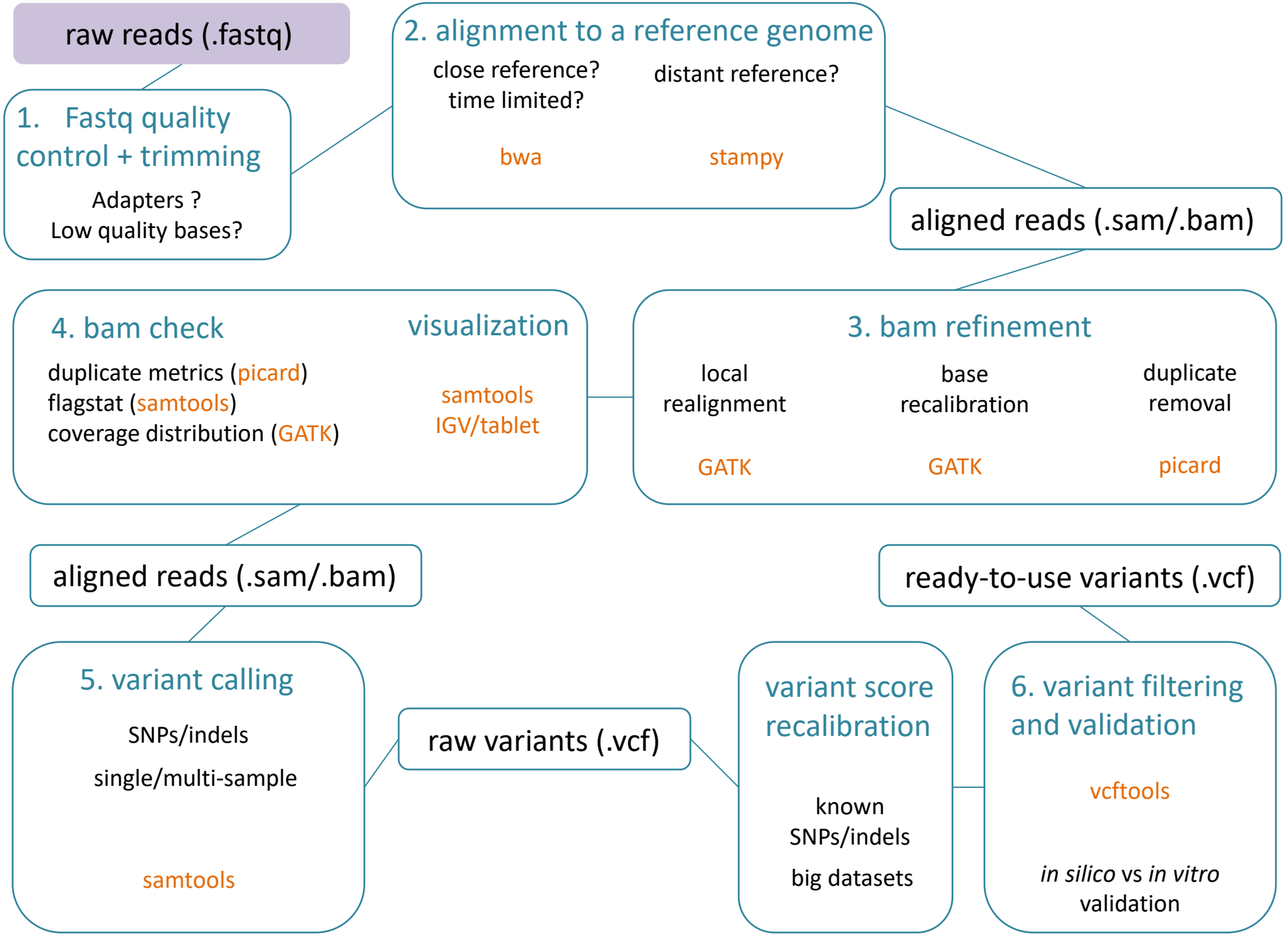
FASTQC references:

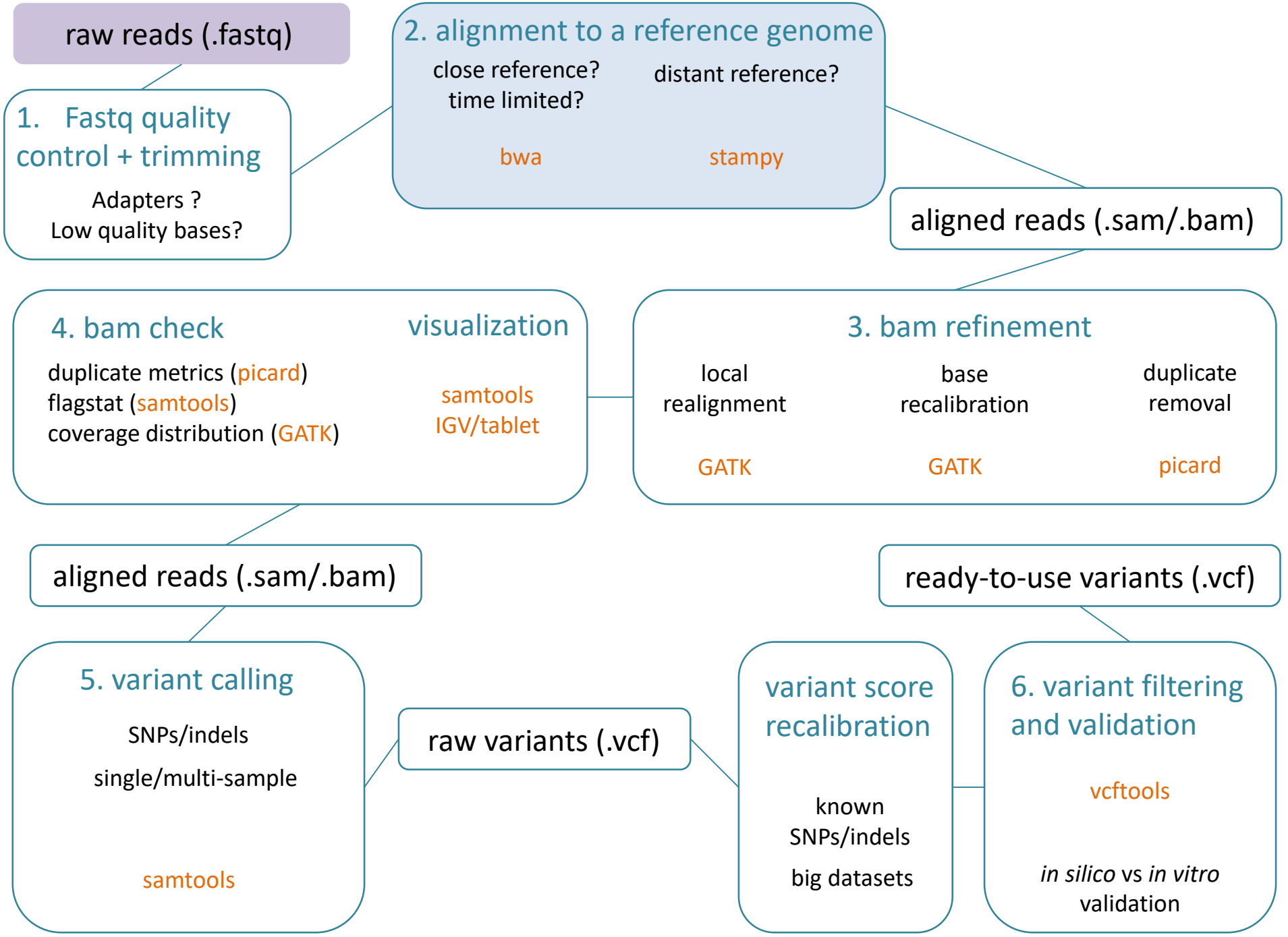
- Software website:

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

- Manual:

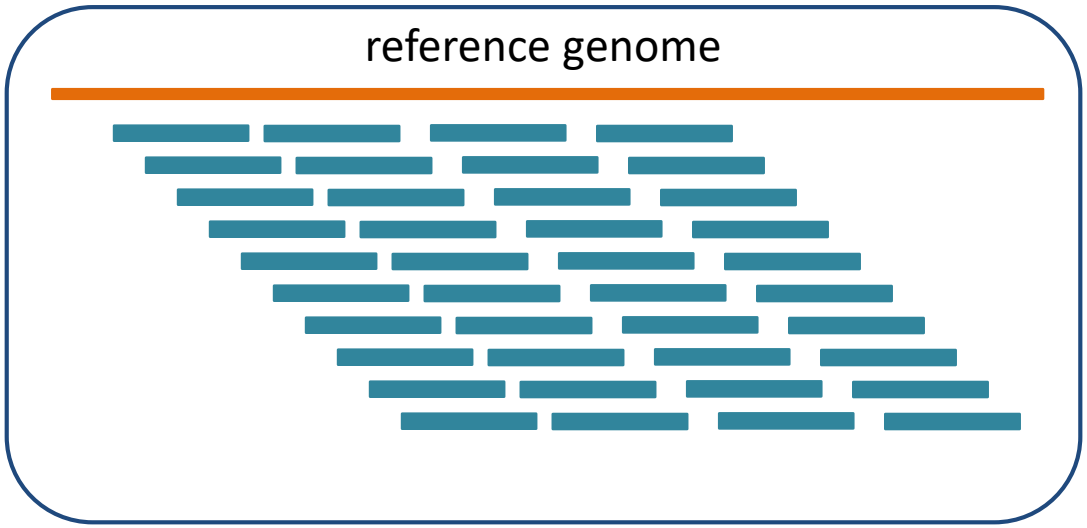
https://insidedna.me/tool_page_assets/pdf_manual/fastqc.pdf


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

2- Alignment to a reference genome

Alignment : process of determining the most likely location within the genome for the observed DNA read



2- Alignment to a reference genome

the shorter the read the harder is to find its location in the genome

big amount of data: computationally challenging for memory and speed

trade-off: speed vs sensitivity – the higher the accuracy the longer the alignment run

two classes of methods:

Burrows-Wheeler

- Fast
- less robust at high divergence with reference genome
- e.g. bwa

Hashing

- slow (needs more memory)
- robust at high divergence with reference genome
- e.g. stampy

2- Alignment to a reference genome

What if there are several possible places to align your sequencing read?

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

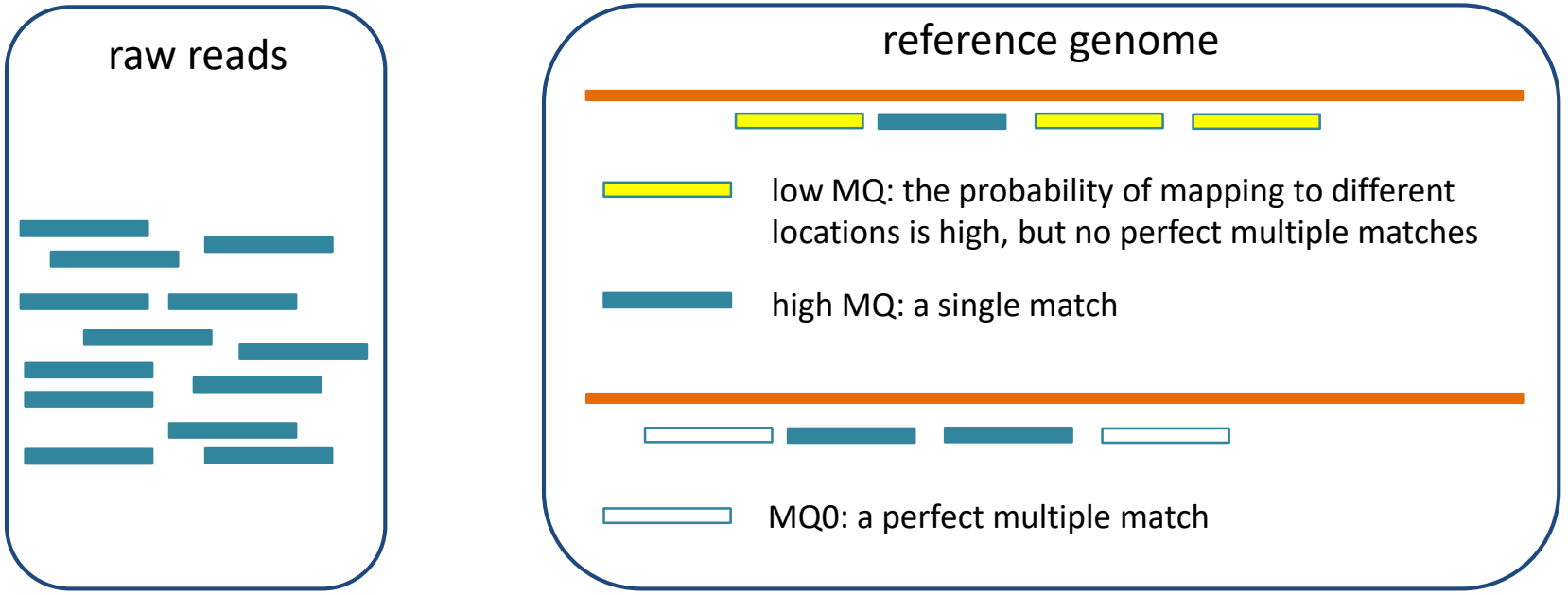
2- Alignment to a reference genome

What if there are several possible places to align your sequencing read?

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

MQ is a phred-score of the quality of the alignment



2- Alignment to a reference genome

This may be due to:

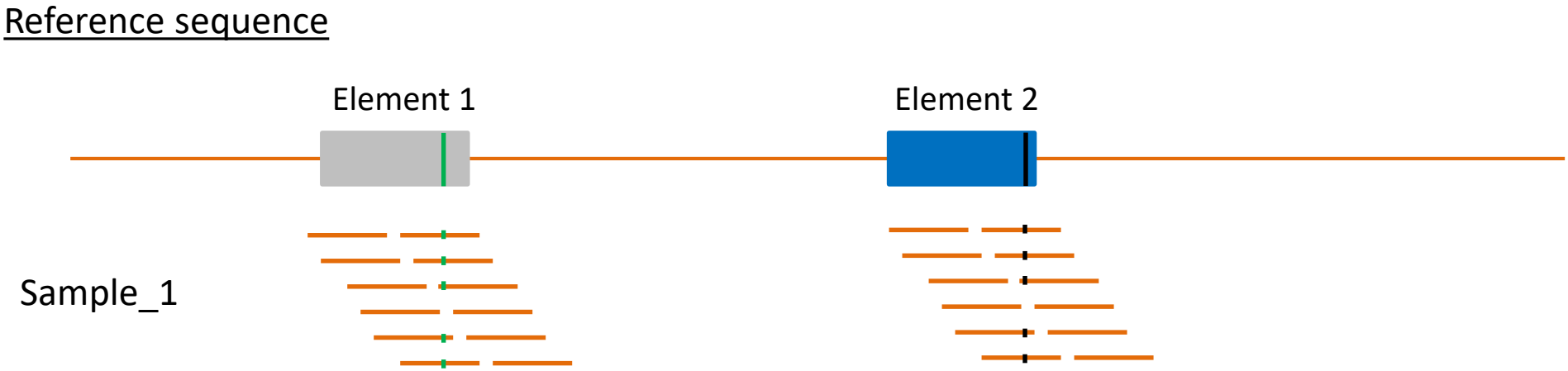
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

Reference sequence

2- Alignment to a reference genome

This may be due to:

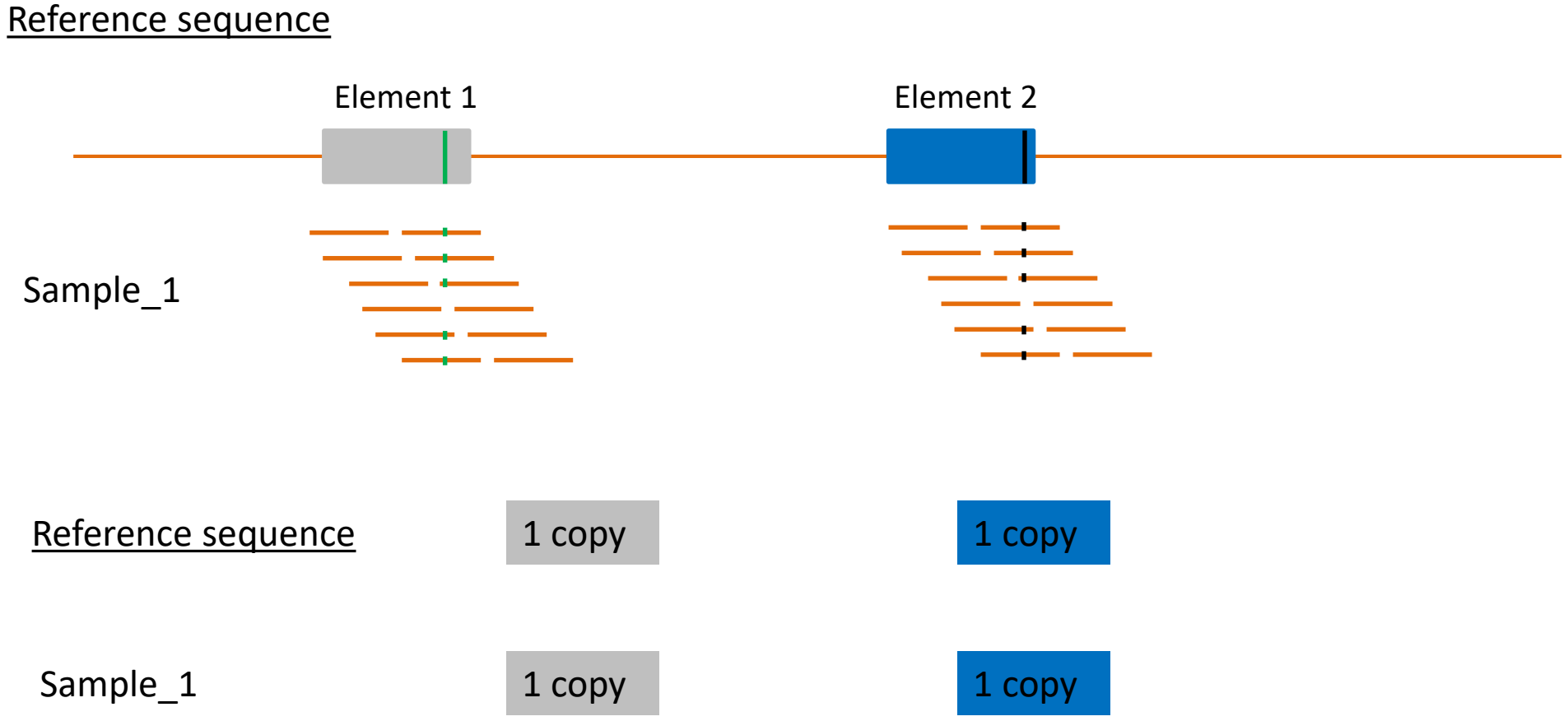
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

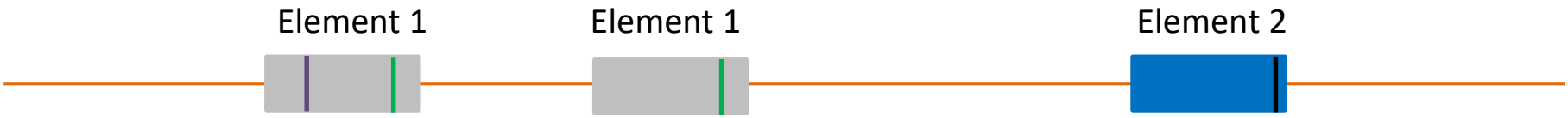


2- Alignment to a reference genome

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

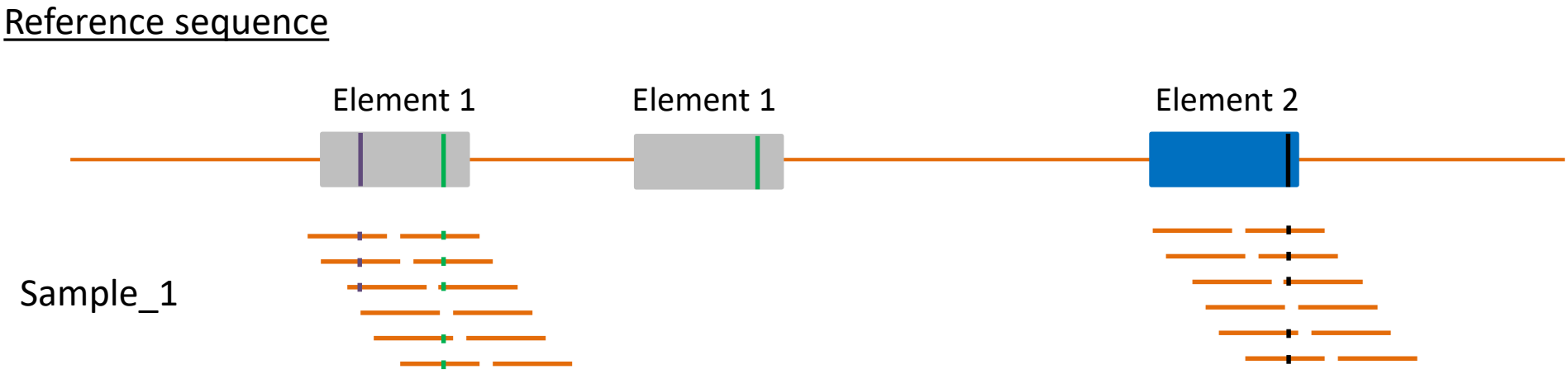
<u>Reference sequence</u>



2- Alignment to a reference genome

This may be due to:

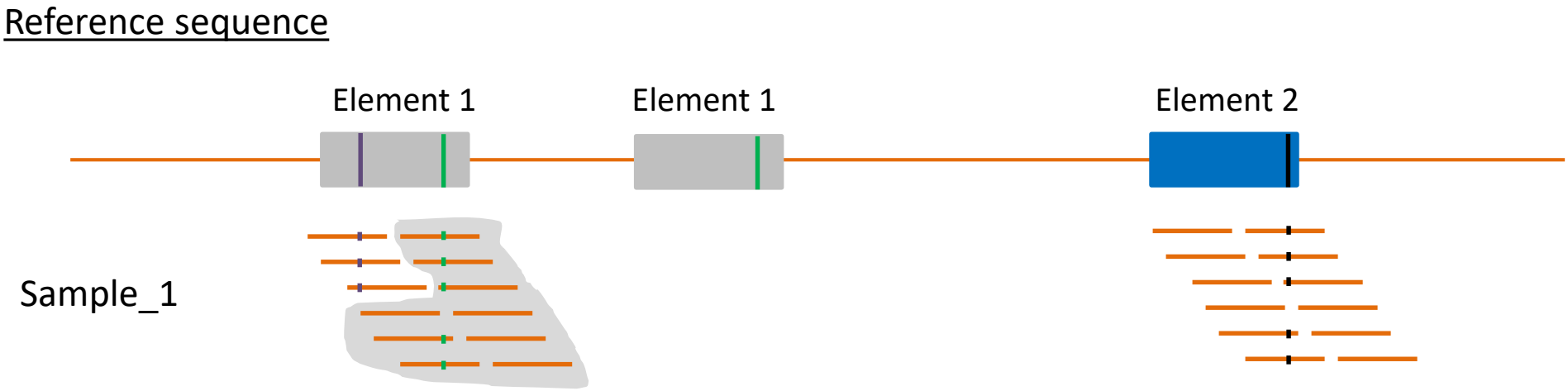
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

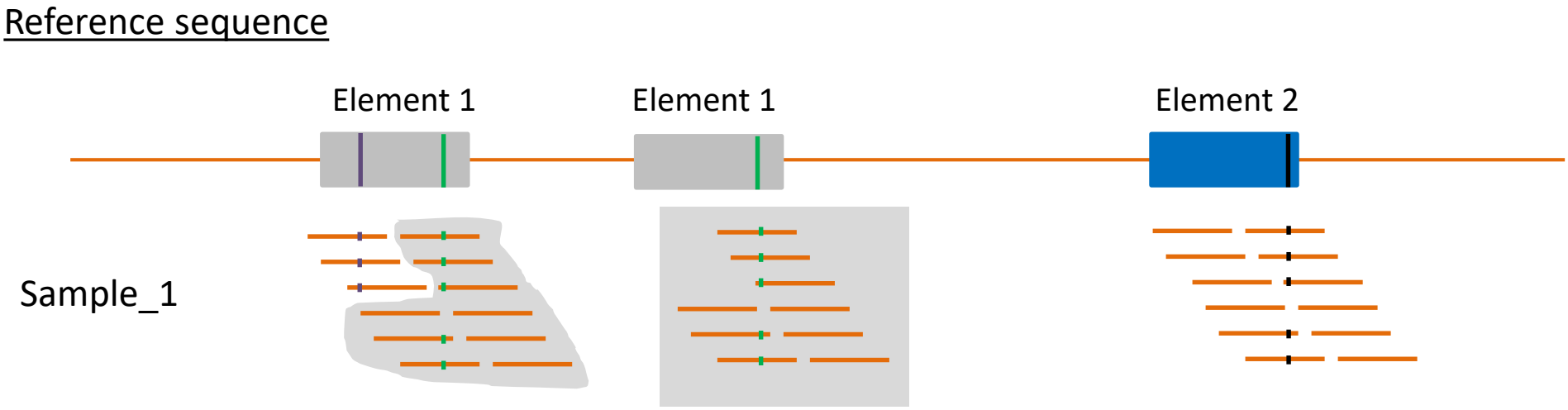
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

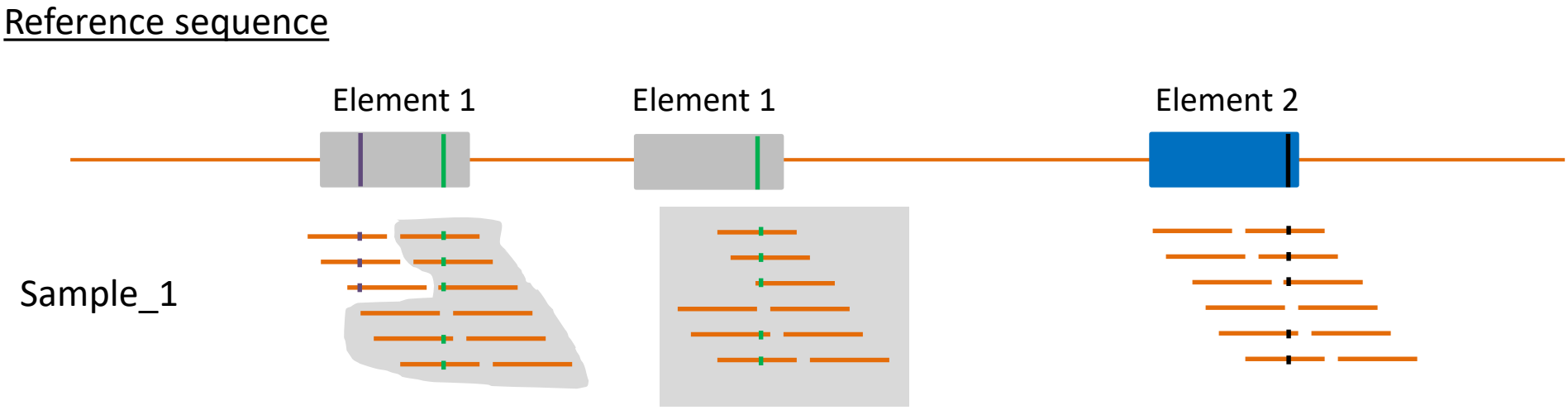
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

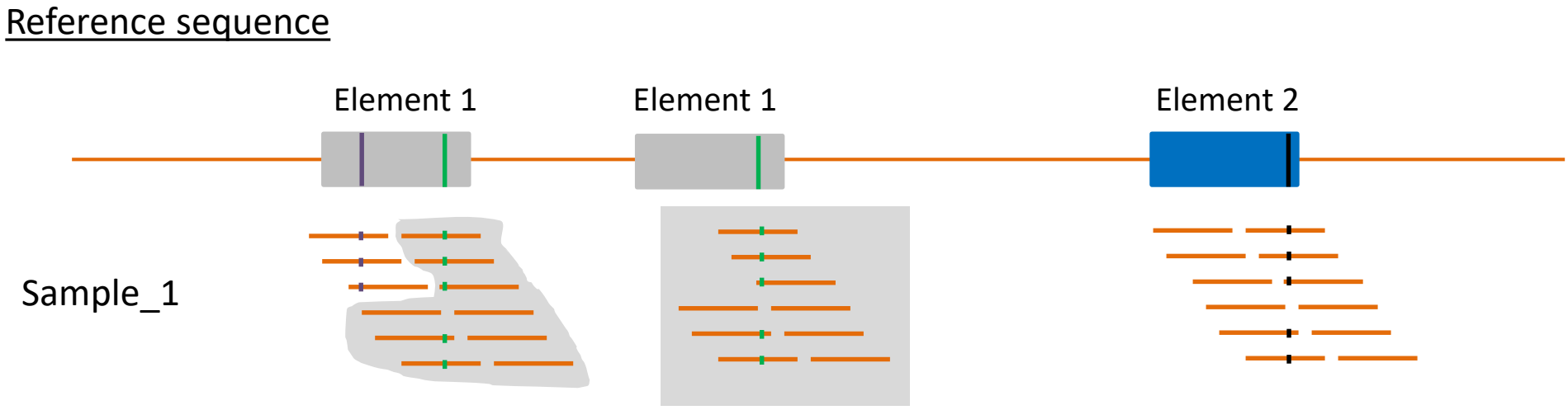


Perfect multiple matches → MQ0
Not a perfect match → Low MQ

2- Alignment to a reference genome

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



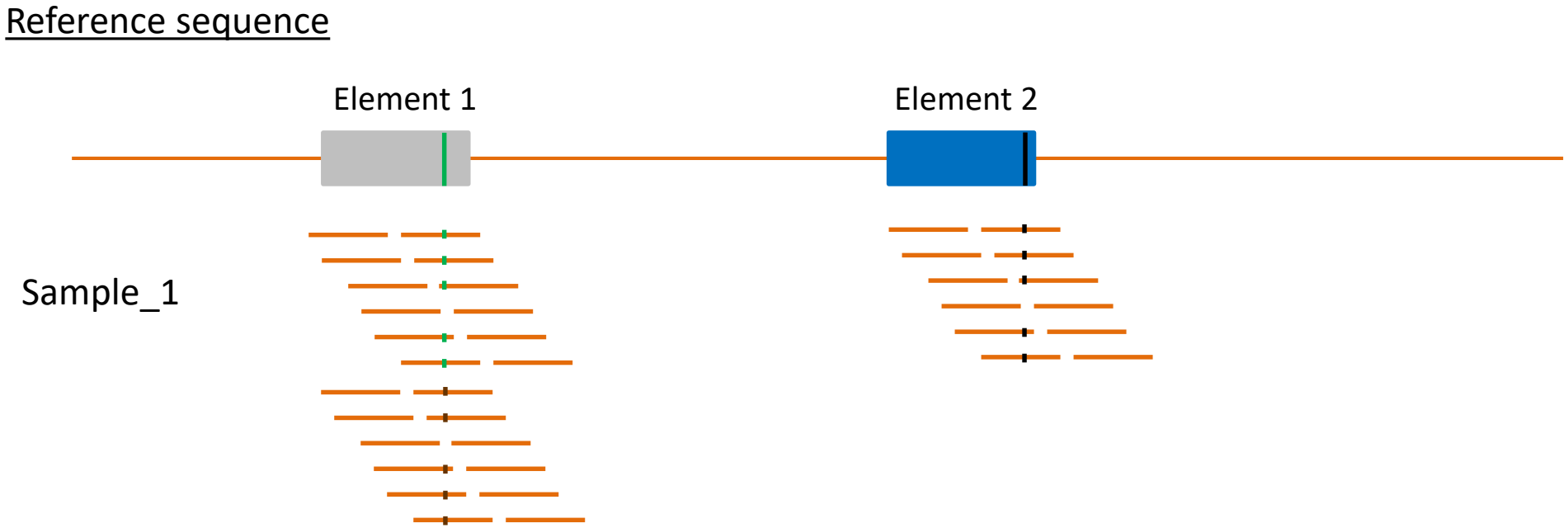
Perfect multiple matches → MQ0
Not a perfect match → Low MQ

Reference sequence	2 copies	1 copy
Sample_1	1 copy	1 copy

2- Alignment to a reference genome

This may be due to:

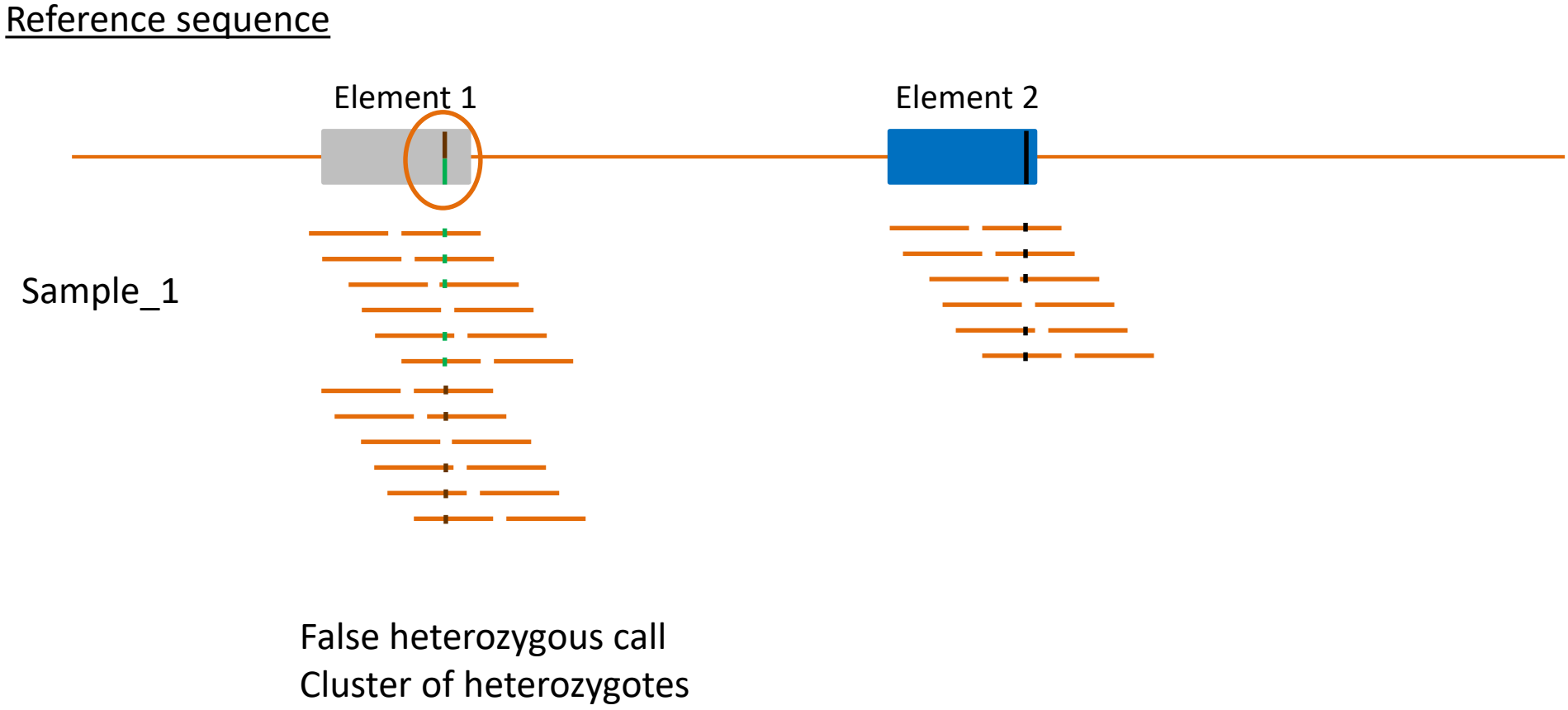
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

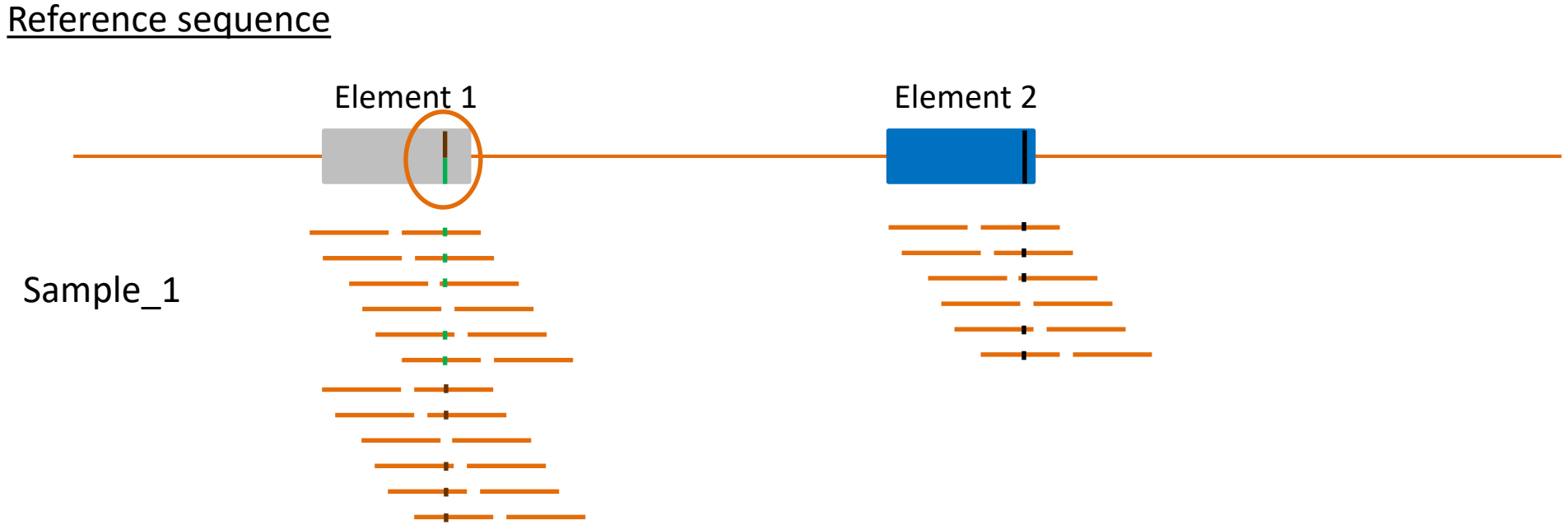
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

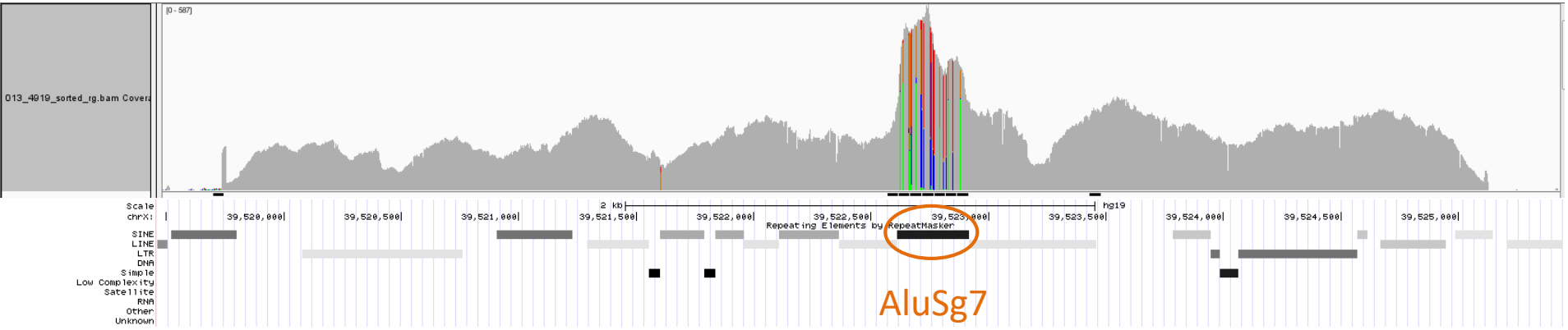
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

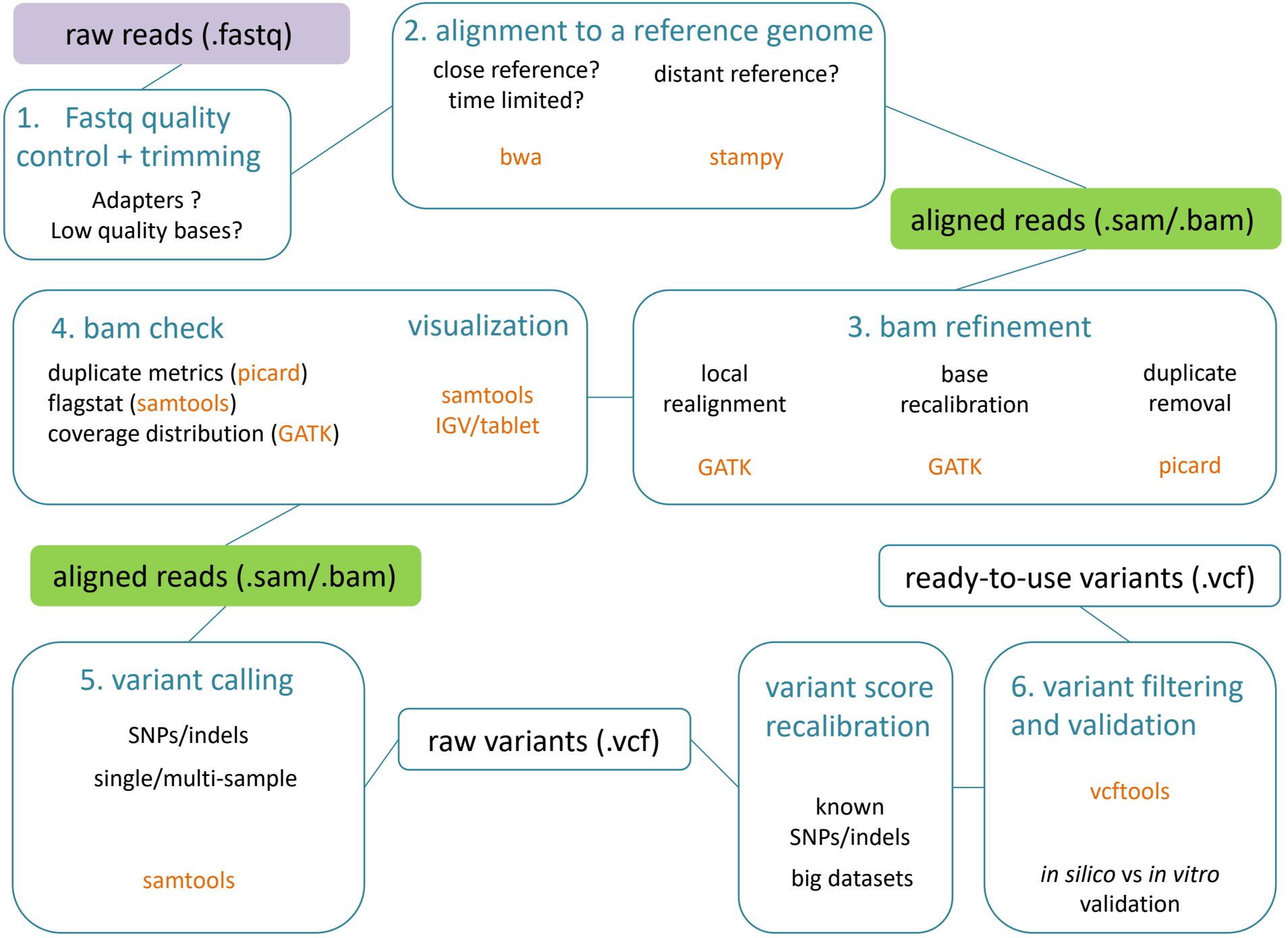
- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps



2- Alignment to a reference genome

This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

aligned reads (.sam/.bam)

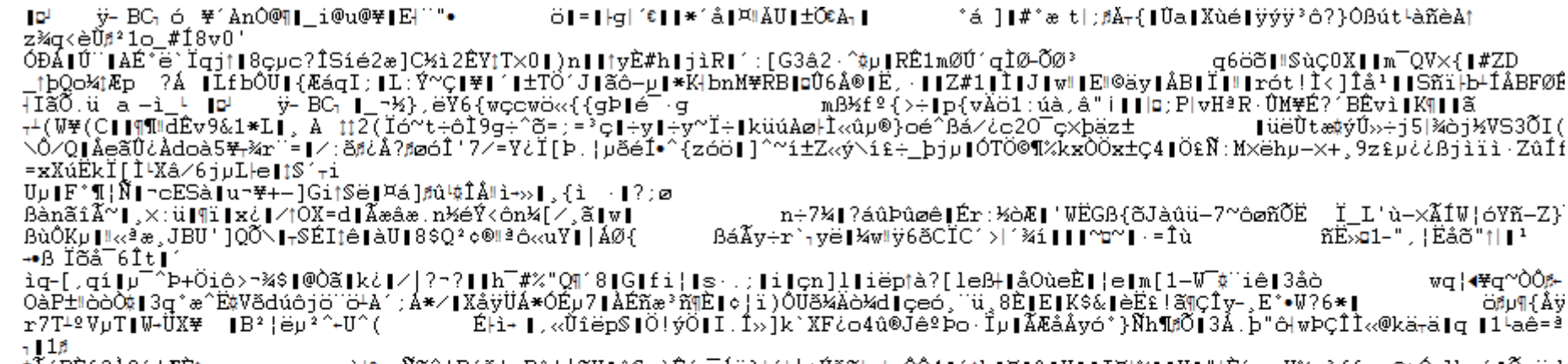
SAM/BAM format

SAM – sequence alignment map
BAM – binary alignment map

Standard formats for alignment
BAM is the binary version of SAM – reduced size, easier to store and to access but the full information is not readable by human eye

aligned reads (.sam/.bam)

BAM format



aligned reads (.sam/.bam)

We can “read” the header of the BAM file…

aligned reads (.sam/.bam)

We can “read” the header of the BAM file…

use samtools to check the header of the BAM

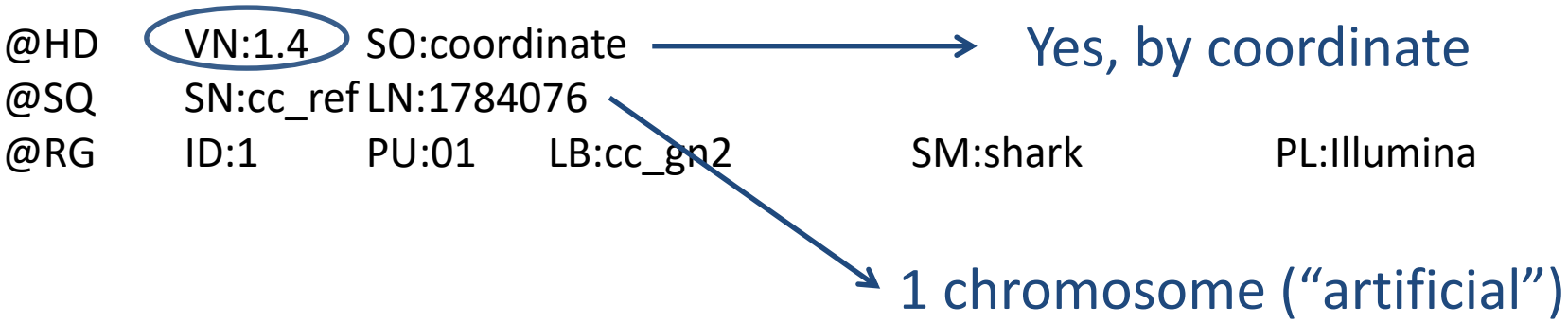
```
samtools view -H cc_gn2_R12_rg_sorted.bam
```

1. How many chromosomes are present in your header?
2. Which version of the BAM is it?
3. Is it sorted?

aligned reads (.sam/.bam)

```
@HD     VN:1.4     SO:coordinate
@SQ     SN:cc_ref LN:1784076
@RG     ID:1       PU:01      LB:cc_gn2          SM:shark          PL:Illumina
```

aligned reads (.sam/.bam)

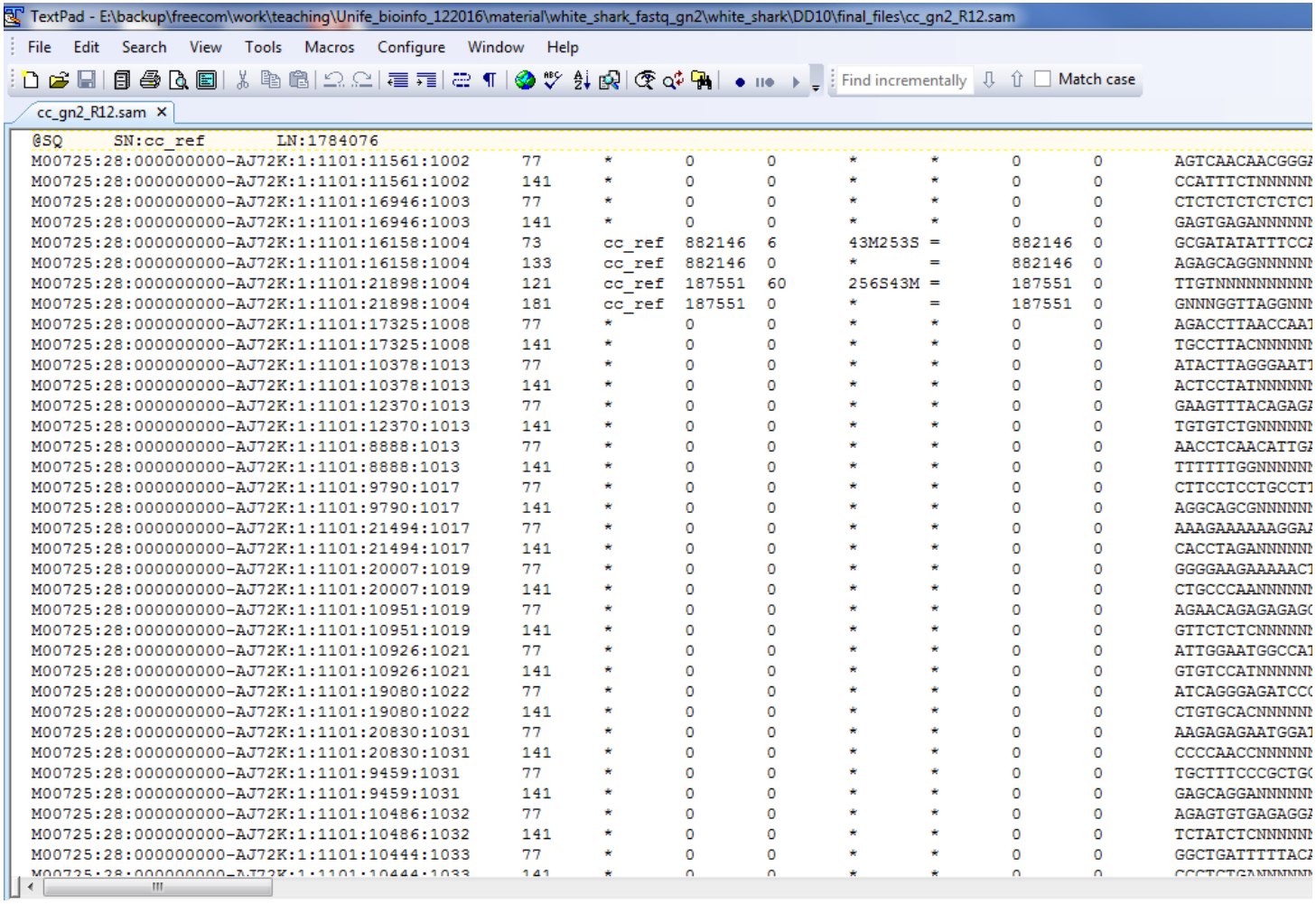


aligned reads (.sam/.bam)

```
@HD     VN:1.3  SO:coordinate
@SQ     SN:I    LN:230218
@SQ     SN:II   LN:813184
@SQ     SN:III  LN:316620
@SQ     SN:IV   LN:1531933
@SQ     SN:IX   LN:439888
@SQ     SN:Mito LN:85779
@SQ     SN:V    LN:576874
@SQ     SN:VI   LN:270161
@SQ     SN:VII  LN:1090940
@SQ     SN:VIII LN:562643
@SQ     SN:X    LN:745751
@SQ     SN:XI   LN:666816
@SQ     SN:XII  LN:1078177
@SQ     SN:XIII LN:924431
@SQ     SN:XIV  LN:784333
@SQ     SN:XV   LN:1091291
@SQ     SN:XVI  LN:948066
@PG     ID:bwa  PN:bwa  VN:0.7.10-r789  CL:bwa mem -M
```

aligned reads (.sam/.bam)

SAM format



aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

2. **Alignment**: contains location and qualities for all the reads

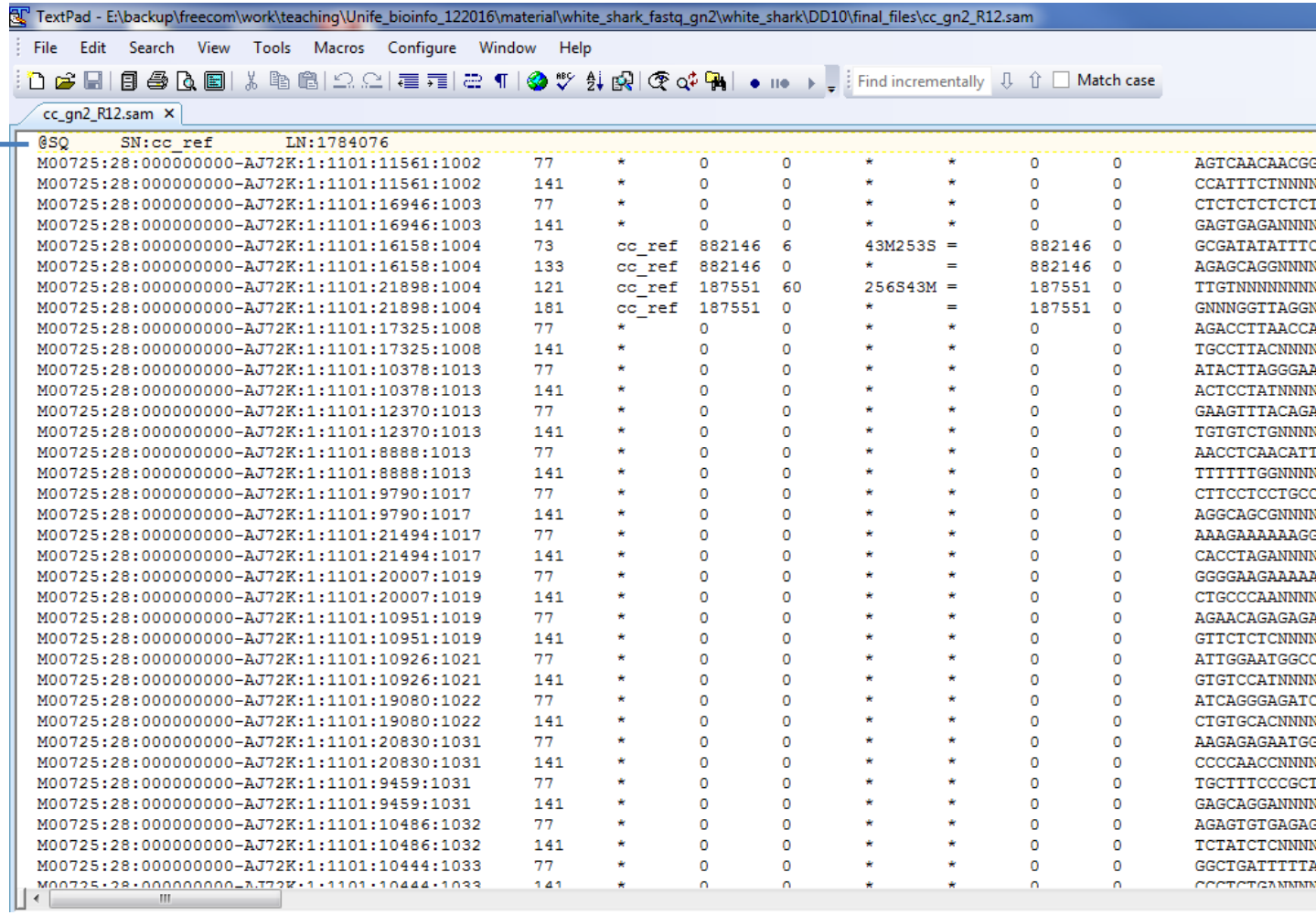
You can find a detailed explanation in the sam/bam format specification (http://samtools.sourceforge.net/SAMv1.pdf).

aligned reads (.sam/.bam)

SAM format

Header

Alignment



aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

Header contains:
@HD – header line
@SQ – Reference sequence dictionary, one per chromosome,
SN (reference sequence name) and **LN** (reference sequence length)
@RG – Read group
@PG – Program, **ID** (identifier)
@CO – comment

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

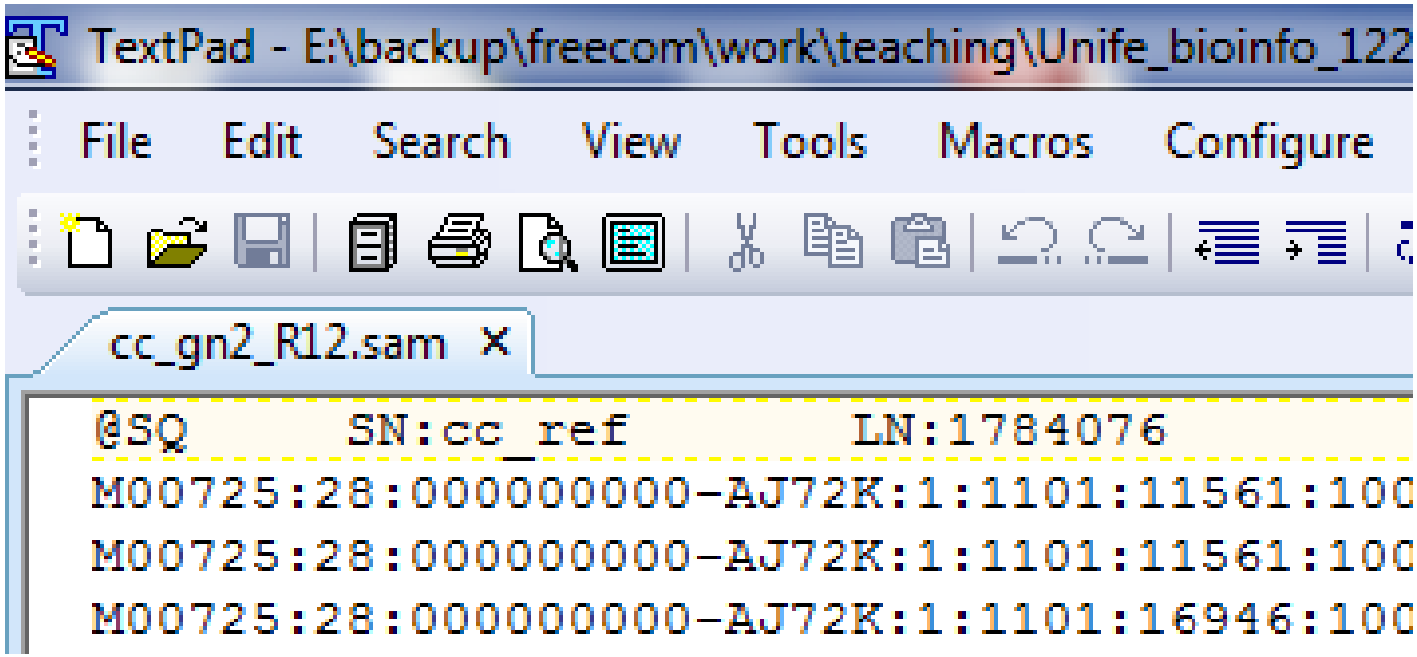
Header contains:
@HD – header line
@SQ – Reference sequence dictionary, one per chromosome,
SN (reference sequence name) and LN (reference sequence length)
@RG – Read group
@PG – Program, **ID** (identifier)
@CO – comment

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

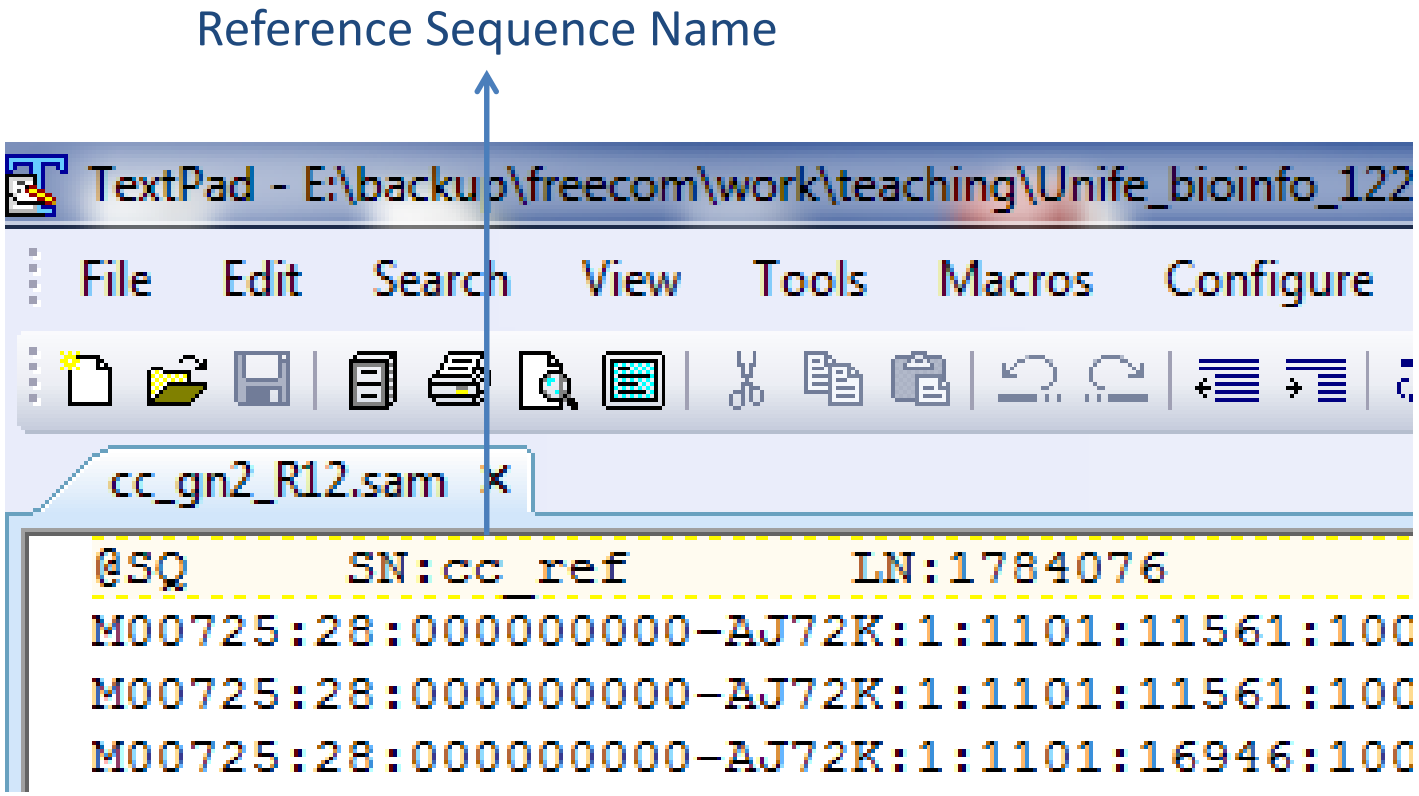


aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

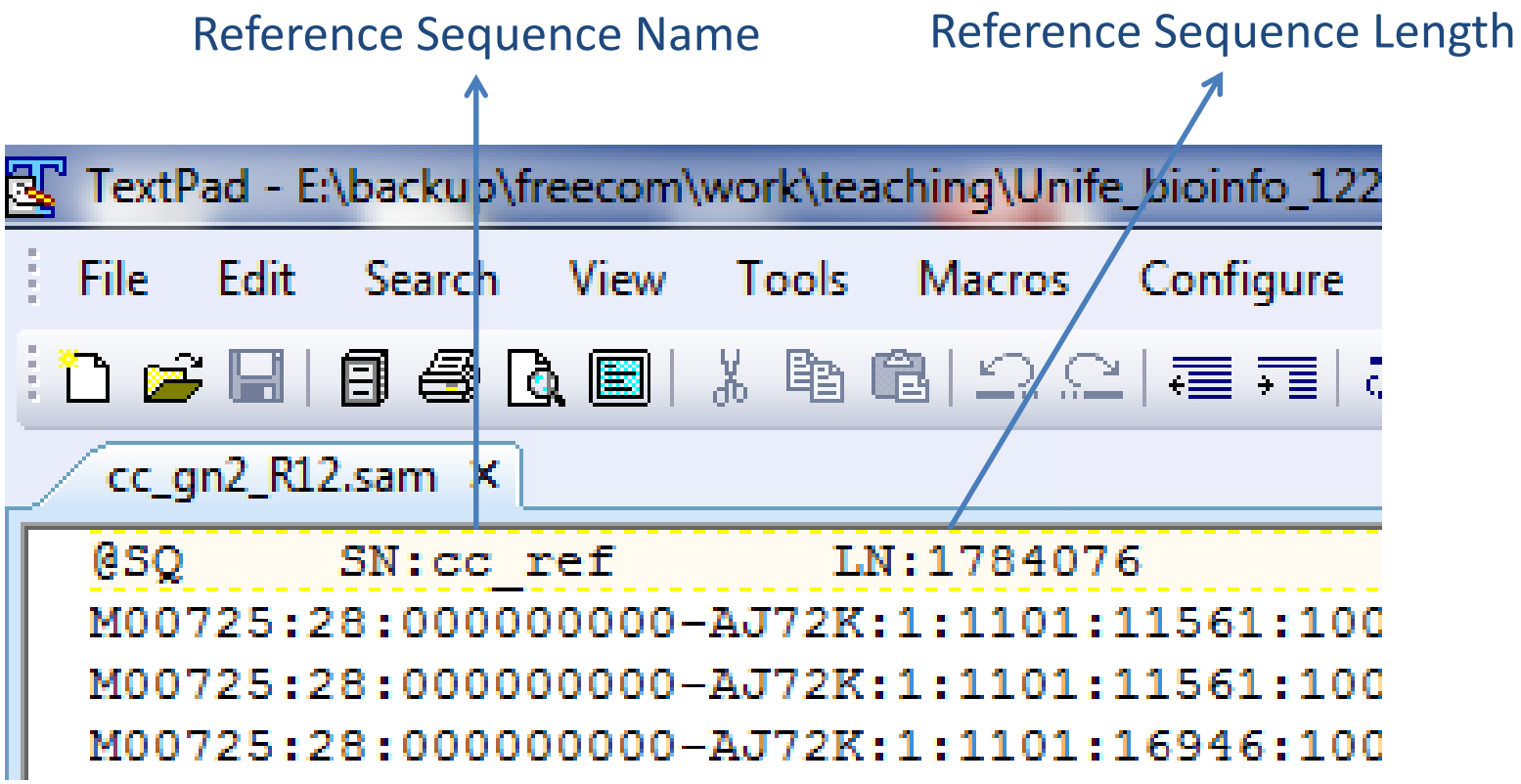


aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

1. **Header**: contains information about the sample

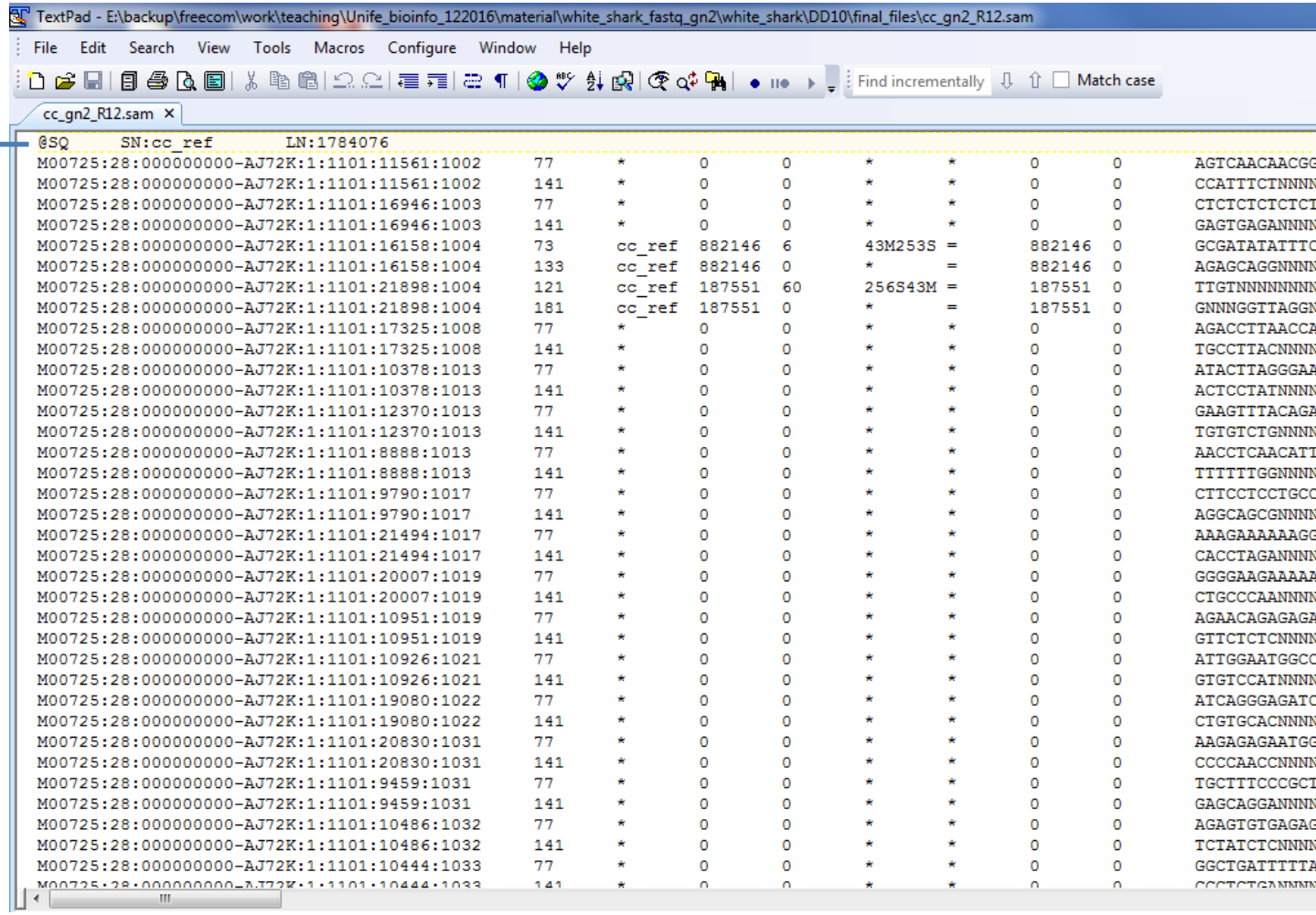


aligned reads (.sam/.bam)

SAM format

Header

Alignment



aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:

2. **Alignment**: contains location and qualities for all the reads

Alignment contains one line per read, and each line contains 12 columns:

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

M00725:28:000000000-AJ72K:1:1101:18215:1102 99 cc_ref 1754677 60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME

M00725:28:000000000-AJ72K:1:1101:18215:1102 99 cc_ref 1754677 60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME **FLAG**

M00725:28:000000000-AJ72K:1:1101:18215:1102 99 cc_ref 1754677 60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

bitwise FLAG

It is an integer, but it represents the sum of different values.

QNAME	FLAG
M00725:28:000000000-AJ72K:1:1101:18215:1102	99

aligned reads (.sam/.bam)

bitwise FLAG

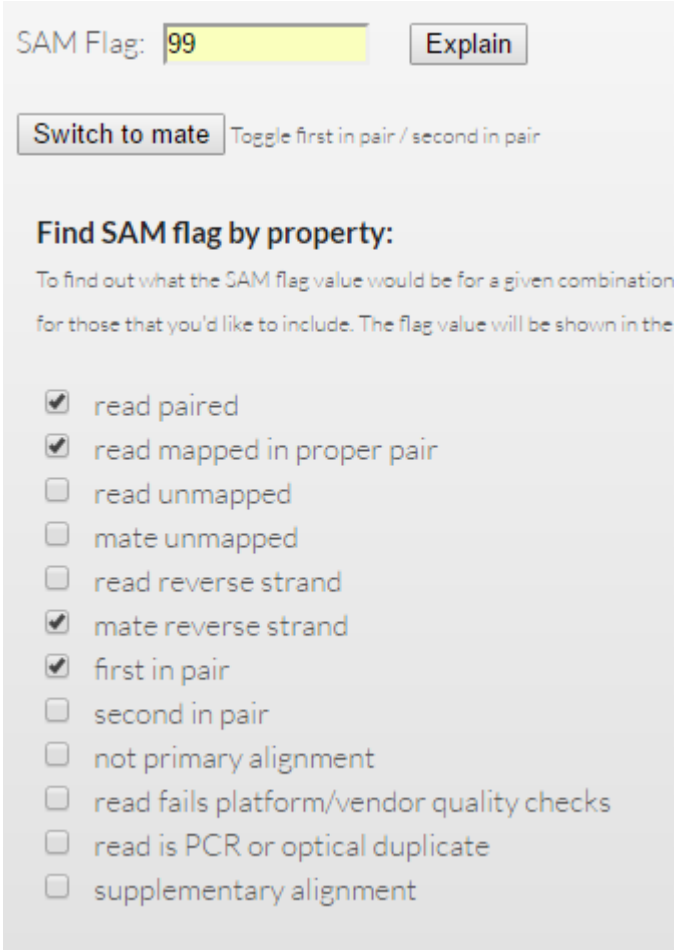
It is an integer, but it represents the sum of different values.

QNAME	FLAG
M00725:28:000000000-AJ72K:1:1101:18215:1102	99

Open Firefox > google.co.uk > Type “bitwise flag broad”

There is a tool online which provides a quick “translation” (https://broadinstitute.github.io/picard/explain-flags.html)

aligned reads (.sam/.bam)



bitwise FLAG

It is an integer, but it represents the sum of different values.

QNAME	FLAG
M00725:28:000000000-AJ72K:1:1101:18215:1102	99

There is a tool online which provides a quick “translation” (https://broadinstitute.github.io/picard/explain-flags.html)

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME		
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

300M = 1754780 309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

CIGAR			
300M	=	1754780	309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

CIGAR	MRNM		
300M	=	1754780	309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

CIGAR	MRNM	MPOS	
300M	=	1754780	309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

CIGAR	MRNM	MPOS	ISIZE
300M	=	1754780	309

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

CIGAR	MRNM	MPOS	ISIZE
300M	=	1754780	309

SEQ

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+

aligned reads (.sam/.bam)

SAM – sequence alignment map
BAM – binary alignment map

2. **Alignment**: contains location and qualities for all the reads

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

QNAME	FLAG	RNAME	POS	MAPQ
M00725:28:000000000-AJ72K:1:1101:18215:1102	99	cc_ref	1754677	60

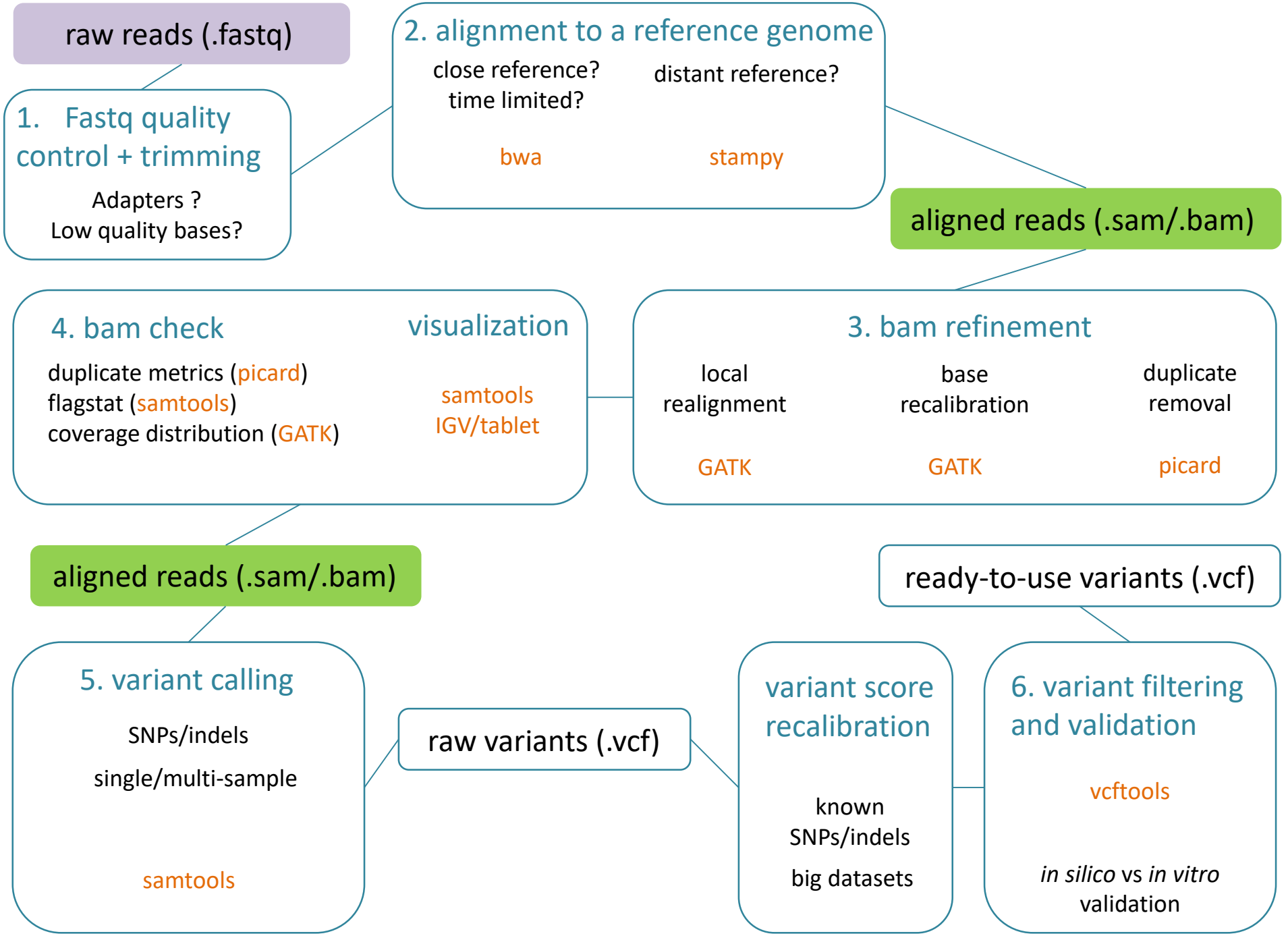
CIGAR	MRNM	MPOS	ISIZE
300M	=	1754780	309

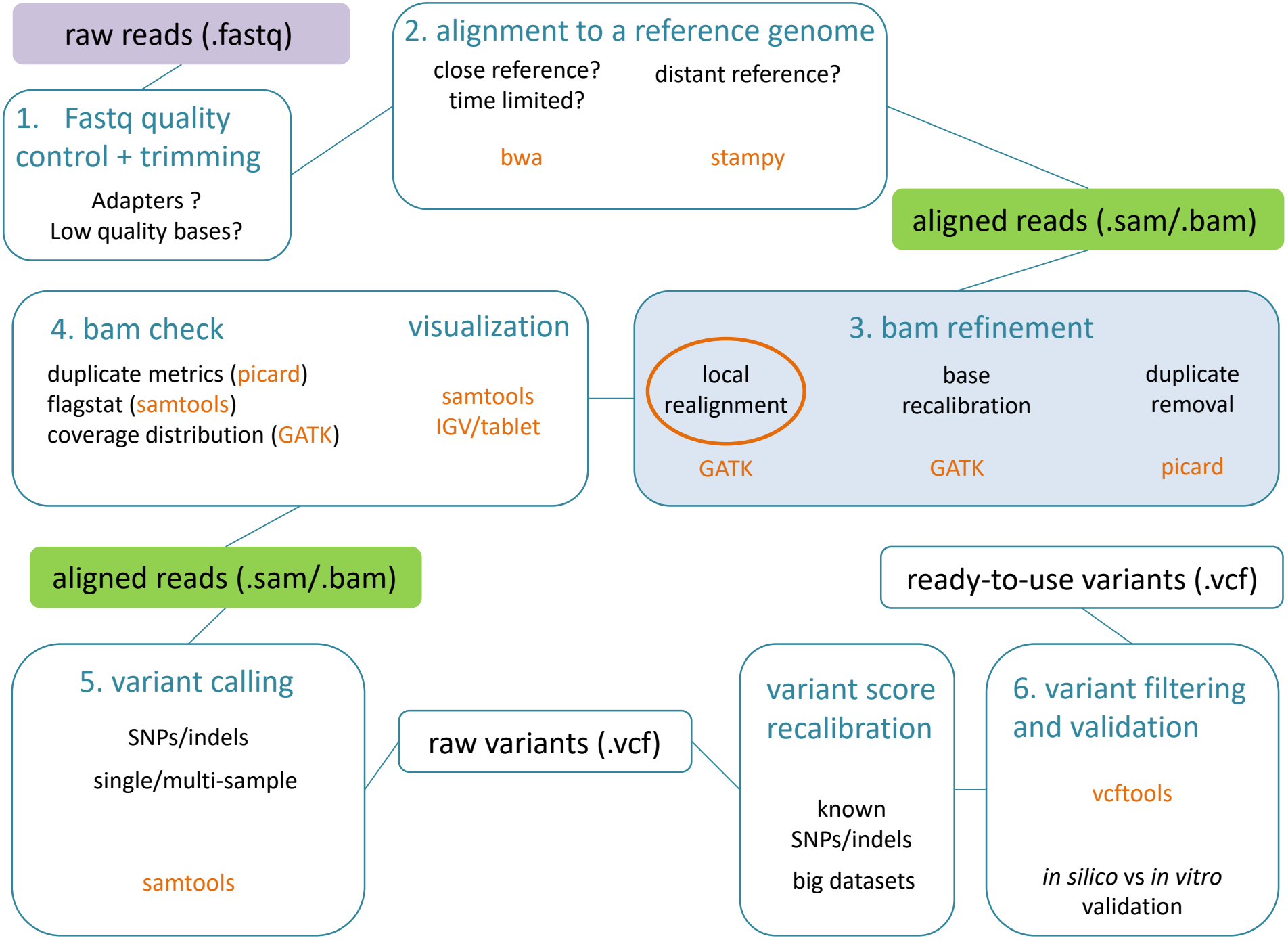
SEQ

CTCCTTCACCAGATGGATTCTCGCCTTACAGTCCTGAGGAAACTAACCGCAGAGTCAACAAAG
TAATGCGAGNNNNNNNGTACTTGCTACAGCNANNNGGTCCAAATNNNTNNNTTATTGGNNNAGATGTT

QUAL

CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGG#######::CFGGGGGGGGGG#:###:9BFGGGGG###:###::DFGDG###4+


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

3- Bam refinement

Input: BAM

Three main steps:

1. Local realignment
2. Base quality recalibration
3. Duplicate removal

3- Bam refinement

Input: BAM

Three main steps:

1. Local realignment
2. Base quality recalibration
3. Duplicate removal

Output: BAM

3- Bam refinement – Local realignment

Problem: Short indels in the sample relative to the reference sequence can pose difficulties for alignment programs. Indels occuring towards the ends of the reads are often not aligned correctly, introducing an excess of SNPs

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATCGGGATGCTTTAGCTGA
          TCGGATGCTGATCGGGATGCTTTAGCTGATGCT
                    CTGATCGGGATGCTTTAGCTGATGCTGATGG
```

3- Bam refinement – Local realignment

Problem: Short indels in the sample relative to the reference sequence can pose difficulties for alignment programs. Indels occuring towards the ends of the reads are often not aligned correctly, introducing an excess of SNPs

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATCGGGATGCTTTAGCTGA
          TCGGATGCTGATCGGGATGCTTTAGCTGATGCT
                    CTGATCGGGATGCTTTAGCTGATGCTGATGG

Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATC____ATGCTTTAGCTGA
          TCGGATGCTGATC____T_GCTTTAGCTGATGCT
                    CTGATC____ATGCTTTAGCTGATGCTGATGG
                         TC_____T_GCTTTAGCTGATGCTGATGGGCTT
```

3- Bam refinement – Local realignment

Problem: Short indels in the sample relative to the reference sequence can pose difficulties for alignment programs. Indels occuring towards the ends of the reads are often not aligned correctly, introducing an excess of SNPs

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATCGGGATGCTTTAGCTGA
          TCGGATGCTGATCGGGATGCTTTAGCTGATGCT
                    CTGATCGGGATGCTTTAGCTGATGCTGATGG
```

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATC____ATGCTTTAGCTGA
          TCGGATGCTGATC____T_GCTTTAGCTGATGCT
                    CTGATC____ATGCTTTAGCTGATGCTGATGG
                         TC_____T_GCTTTAGCTGATGCTGATGGGCTT
```

3- Bam refinement – Local realignment

Problem: Short indels in the sample relative to the reference sequence can pose difficulties for alignment programs. Indels occuring towards the ends of the reads are often not aligned correctly, introducing an excess of SNPs

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATCGGGATGCTTTAGCTGA
          TCGGATGCTGATCGGGATGCTTTAGCTGATGCT
                    CTGATCGGGATGCTTTAGCTGATGCTGATGG
```

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATC____ATGCTTTAGCTGA
          TCGGATGCTGATC____T_GCTTTAGCTGATGCT
                    CTGATC____ATGCTTTAGCTGATGCTGATGG
                         TC_____T_GCTTTAGCTGATGCTGATGGGCTT
```

```
Ref: ACTTTCGGATGCTGATCGGGATGCTTTAGCTGATGCTGATGGGCTTTCGATCGATTTAAAAGCT
     ACTTTCGGATGCTGATC____ATGCTTTAGCTGA
          TCGGATGCTGATC____ATGCTTTAGCTGATGCT
                    CTGATC____ATGCTTTAGCTGATGCTGATGG
                         TC____ATGCTTTAGCTGATGCTGATGGGCTT
```

3- Bam refinement – Local realignment

It uses the full alignment context to determine whether the indel exists.

Two-step process:

3- Bam refinement – Local realignment

It uses the full alignment context to determine whether the indel exists.

Two-step process:

1. **RealignerTargetCreator:** it determines the small suspicious intervals which are likely in need of realignment;

3- Bam refinement – Local realignment

It uses the full alignment context to determine whether the indel exists.

Two-step process:

1. **RealignerTargetCreator:** it determines the small suspicious intervals which are likely in need of realignment

2. **IndelRealigner:** it runs the realignment on those intervals

3- Bam refinement – Local realignment

It uses the full alignment context to determine whether the indel exists.

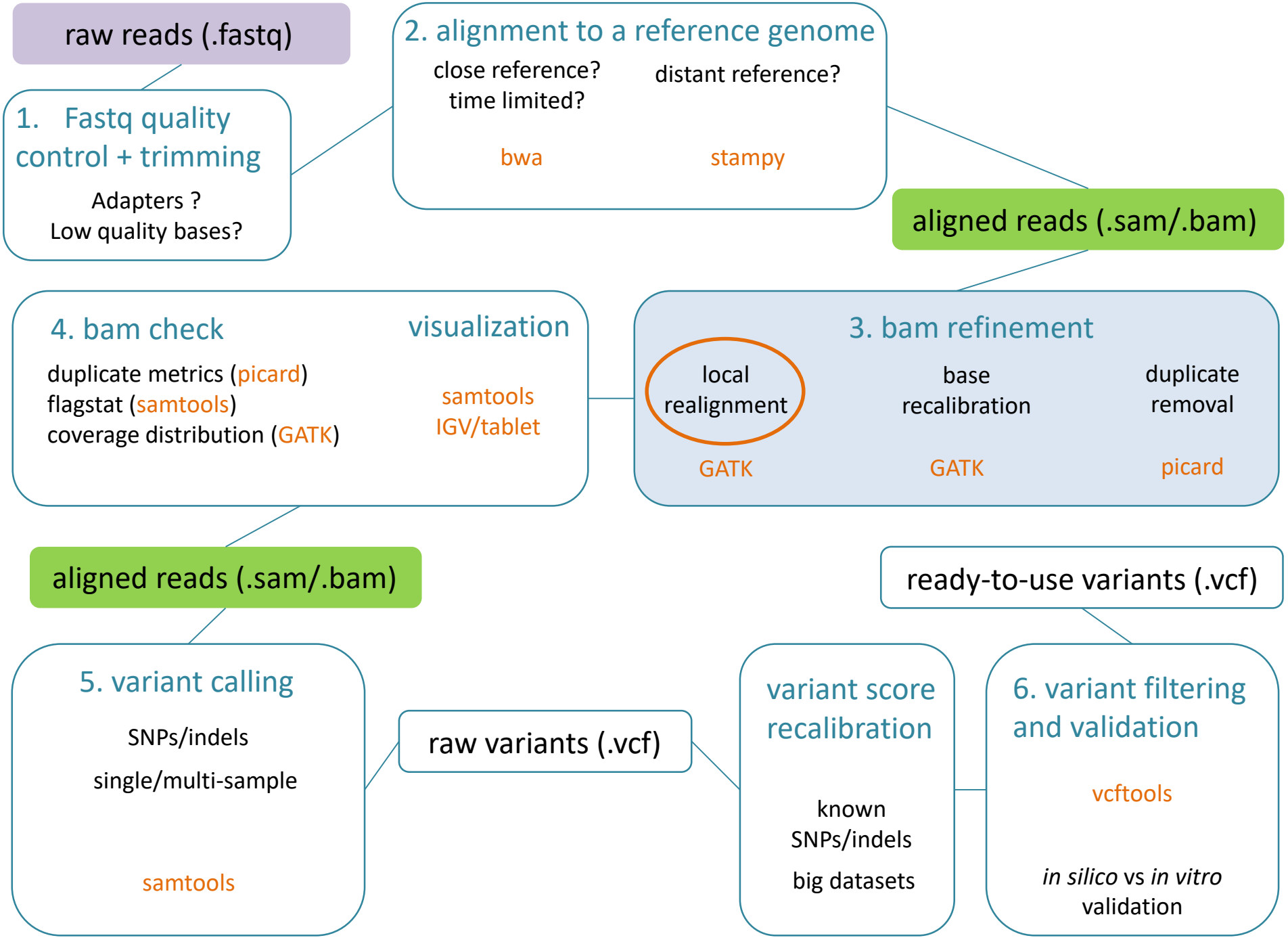
Two-step process:

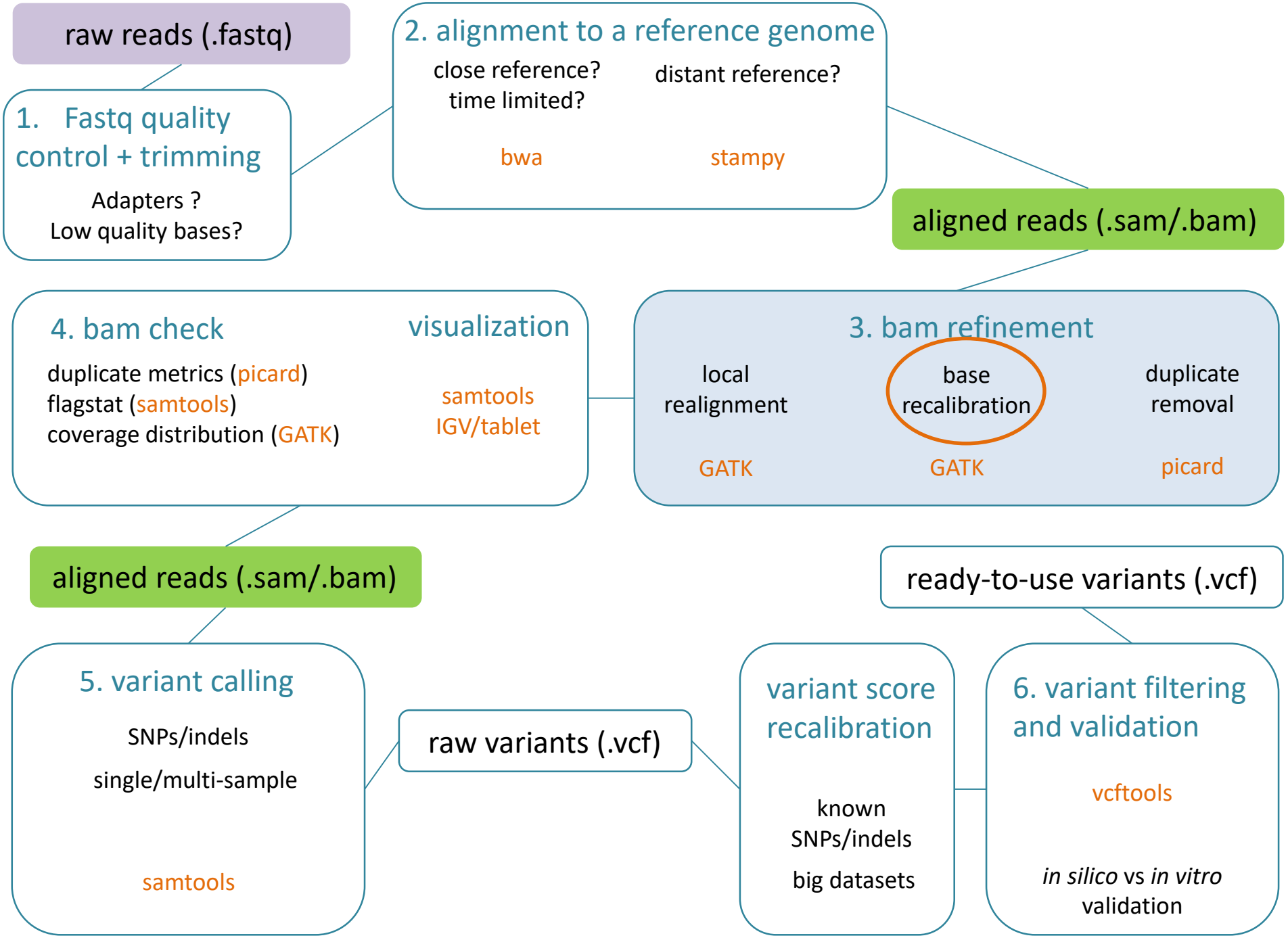
1. **RealignerTargetCreator:** it determines the small suspicious intervals which are likely in need of realignment

2. **IndelRealigner:** it runs the realignment on those intervals

notes:

- having a list of known indels helps


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

3- Bam refinement – Base Quality Recalibration

Each base call has an associated base call quality (phred-scale).
Rule of thumb: anything less than Q20 is not useful data.

3- Bam refinement – Base Quality Recalibration

Each base call has an associated base call quality (phred-scale).
Rule of thumb: anything less than Q20 is not useful data.

The quality of a call depends on multiple factors (e.g. position in the read, sequence context).

3- Bam refinement – Base Quality Recalibration

Each base call has an associated base call quality (phred-scale).
Rule of thumb: anything less than Q20 is not useful data.

The quality of a call depends on multiple factors (e.g. position in the read, sequence context).

In addition, the alignment can provide useful information.
Mismatches to the reference are considered errors (unless they are described polymoprhisms).

It requires a catalogue of variable sites!

How does it work?

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

BQ: base quality

Covariates: position in the read, sequencing cycle, dinucleotide, …

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

List of know variants

123	C/A
145	G/A
1345	C/T

BQ: base quality

Covariates: position in the read, sequencing cycle, dinucleotide, …

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

List of know variants

123	C/A
145	G/A
1345	C/T

Considered as real variants

BQ: base quality

Covariates: position in the read, sequencing cycle, dinucleotide, …

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

List of know variants

123	C/A
145	G/A
1345	C/T

Considered as real variants

Recalibrated BQ using different covariates

BQ: base quality

Covariates: position in the read, sequencing cycle, dinucleotide, …

3- Bam refinement – Base Quality Recalibration

Considered as real variants

Recalibrated BQ using different covariates

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

3- Bam refinement – Base Quality Recalibration

BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

Considered as real variants

Recalibrated BQ using different covariates

High quality >>>>>>>>>>>>> Low Quality

BQ: base quality

Covariates: position in the read, end of read with worse calls!

3- Bam refinement – Base Quality Recalibration

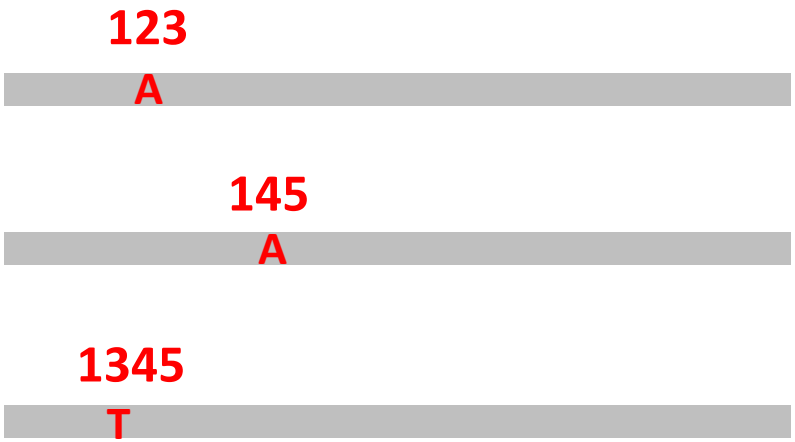
BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

Considered as real variants

Recalibrated BQ using different covariates

High quality >>>>>>>>>>>>> Low Quality



BQ: base quality

Covariates: position in the read, end of read with worse calls!

3- Bam refinement – Base Quality Recalibration

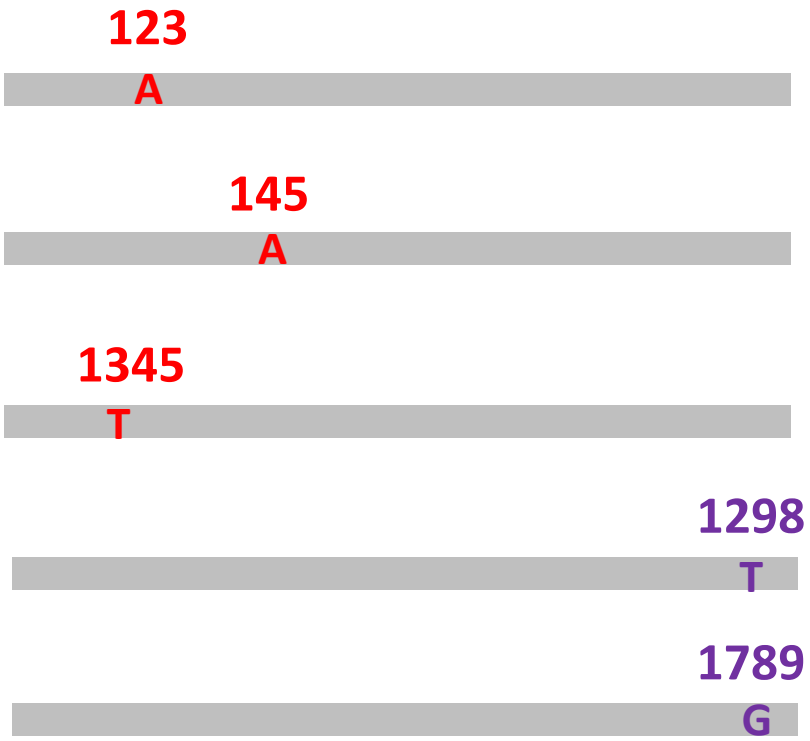
BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ	cov1 cov2 cov3…

Considered as real variants

Recalibrated BQ using different covariates

High quality >>>>>>>>>>>>> Low Quality



BQ: base quality

Covariates: position in the read, end of read with worse calls!

3- Bam refinement – Base Quality Recalibration

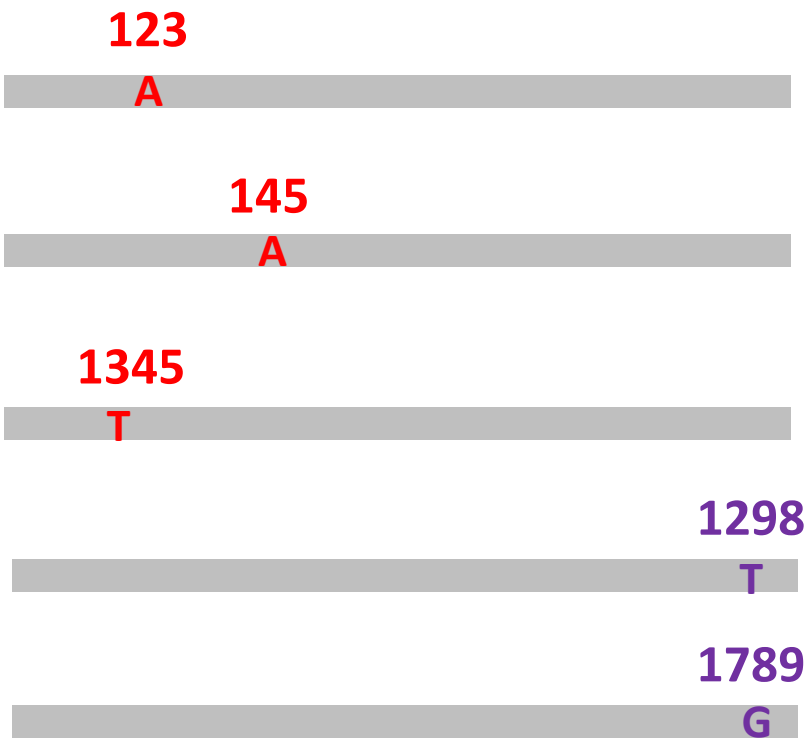
BAM files with variants

123	C/A	BQ	cov1 cov2 cov3…
145	G/A	BQ	cov1 cov2 cov3…
1298	G/T	BQ_1	cov1 cov2 cov3…
1345	C/T	BQ	cov1 cov2 cov3…
1789	C/G	BQ_1	cov1 cov2 cov3…

Considered as real variants

Recalibrated BQ using different covariates

High quality >>>>>>>>>>>>> Low Quality



BQ: base quality

Covariates: position in the read, end of read with worse calls!

3- Bam refinement – Base Quality Recalibration

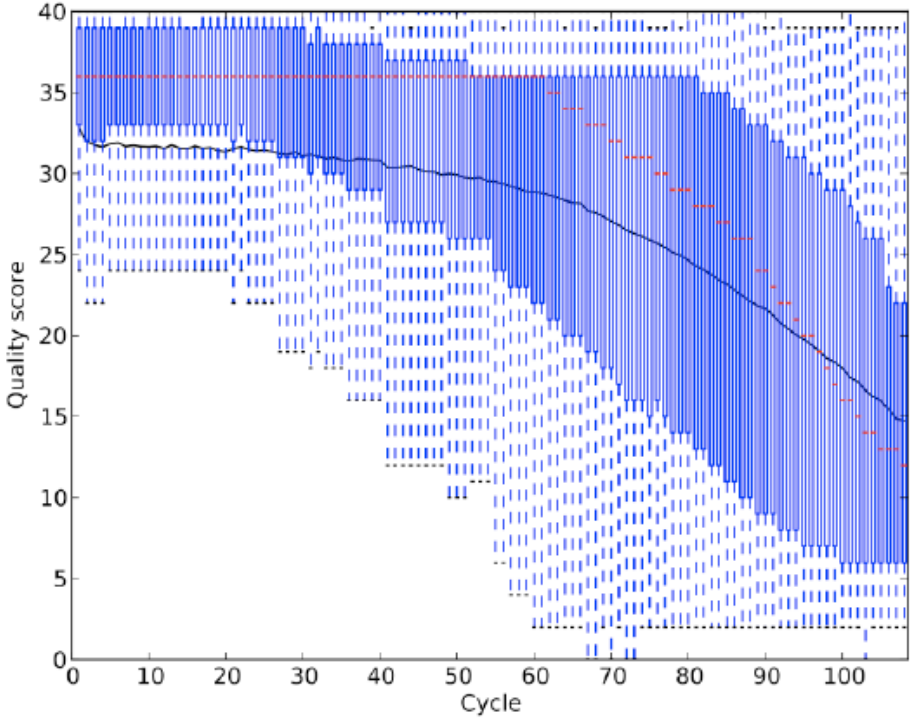
Covariate: cycle number

First cycle: higher quality
Last cycles: lower quality

3- Bam refinement – Base Quality Recalibration

Covariate: cycle number

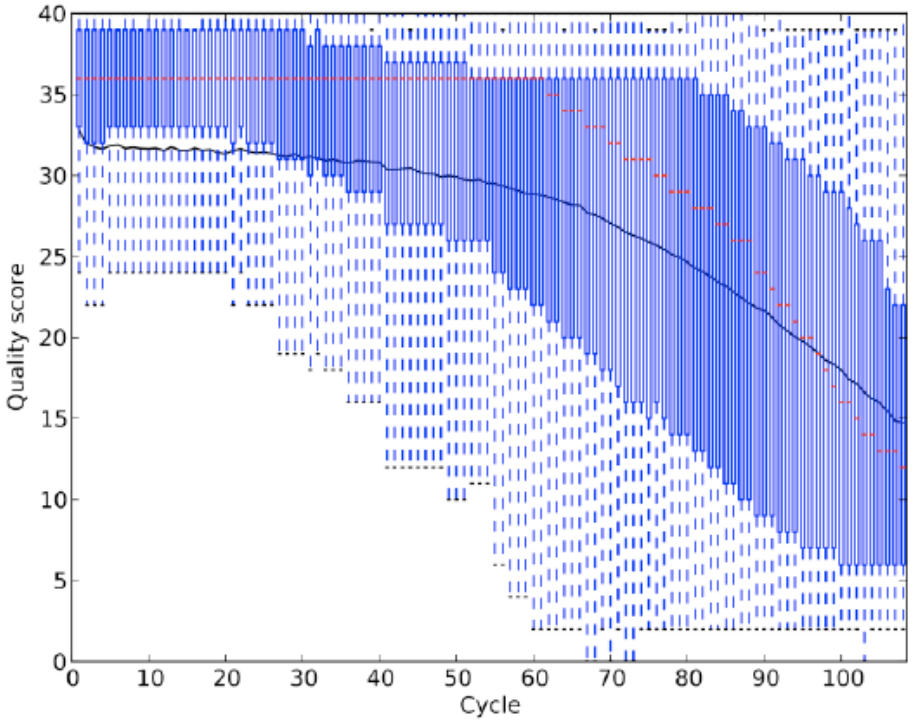
First cycle: higher quality
Last cycles: lower quality

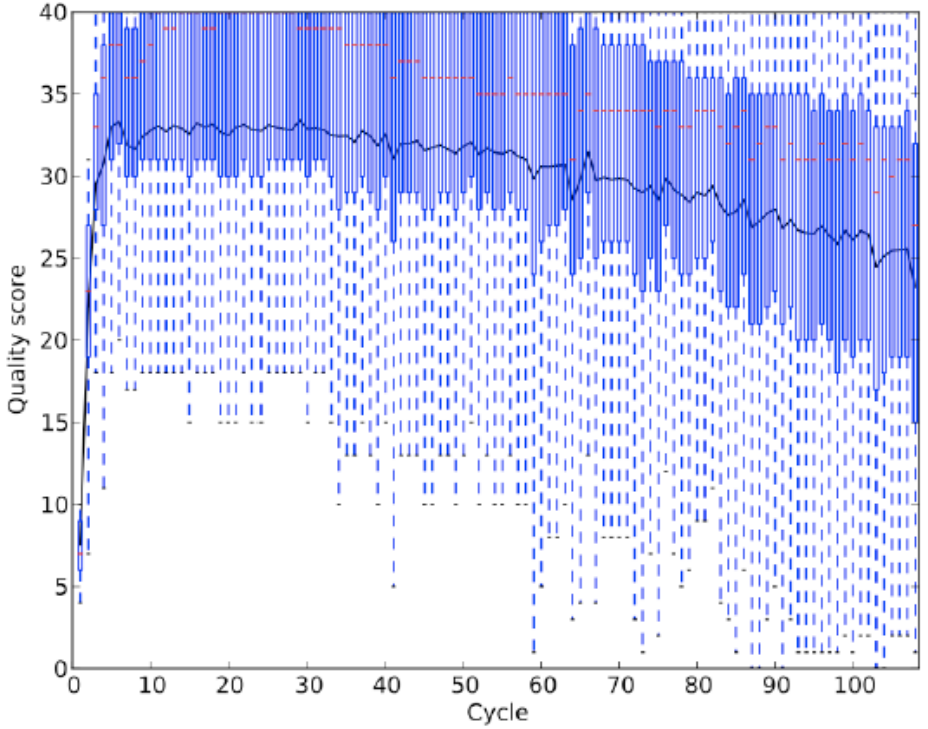


3- Bam refinement – Base Quality Recalibration

Covariate: cycle number

First cycle: higher quality
Last cycles: lower quality





3- Bam refinement – Base Quality Recalibration

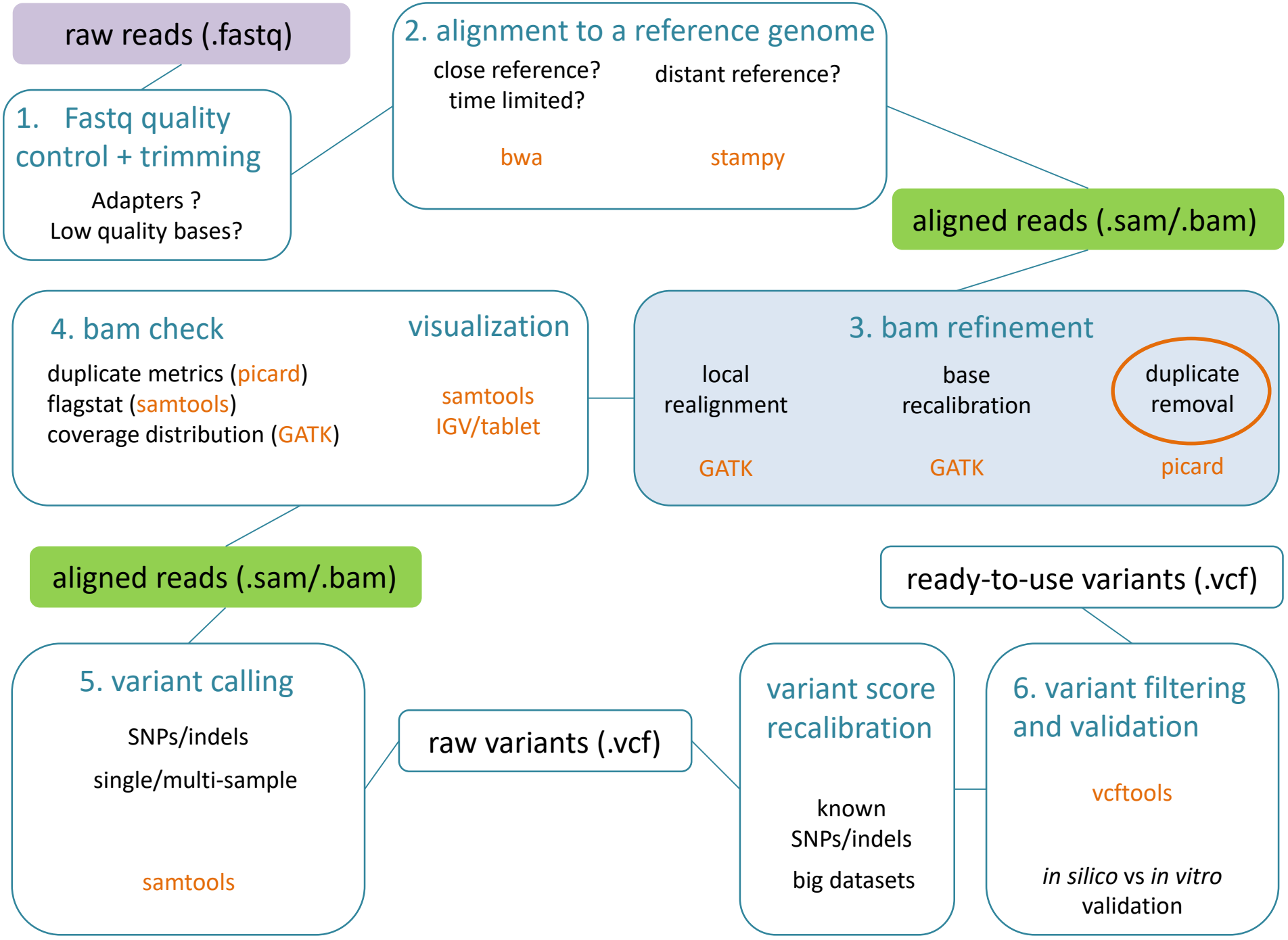
We will not run the Base Quality Recalibration because of time and list of variants available.

Few more details:

It supports several platforms: Illumina, SOLiD, 454, Complete Genomics, Pacific Biosciences (stated on the website) and IonTorrent (stated in the GATK forum).

You can find how to do it at:

https://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_gatk_tools_walkers_bqsr_BaseRecalibrator.php


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

3- Bam refinement – Duplicate removal

PCR is used during library preparation.

This can result in duplicate DNA fragments in the final library prep.

3- Bam refinement – Duplicate removal

PCR is used during library preparation.

This can result in duplicate DNA fragments in the final library prep.

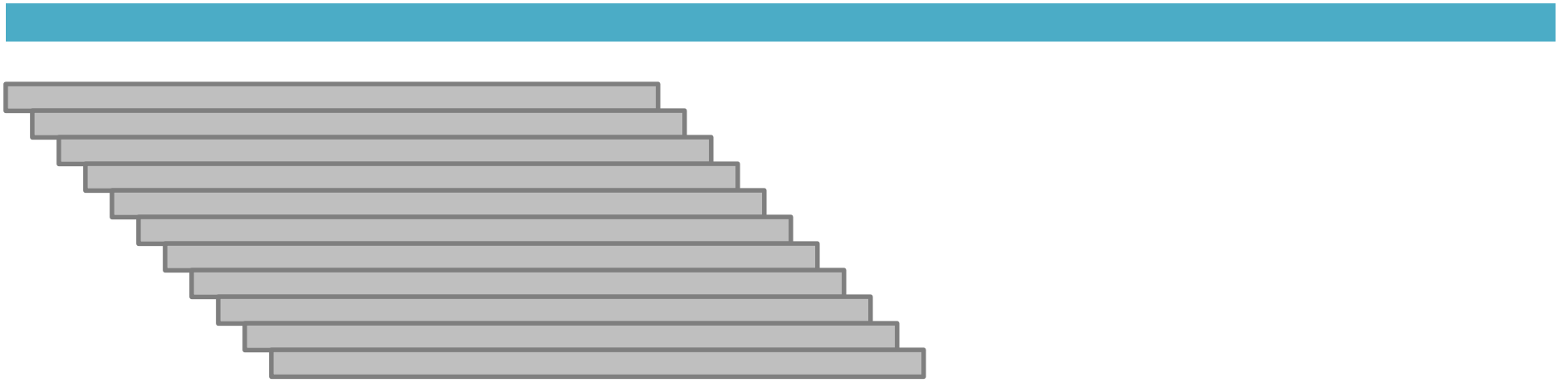
What we want: information from independent fragments
What we do not want: copies of the same information coming from one fragment

3- Bam refinement – Duplicate removal

PCR is used during library preparation.

This can result in duplicate DNA fragments in the final library prep.

What we want: information from independent fragments
What we do not want: copies of the same information coming from one fragment

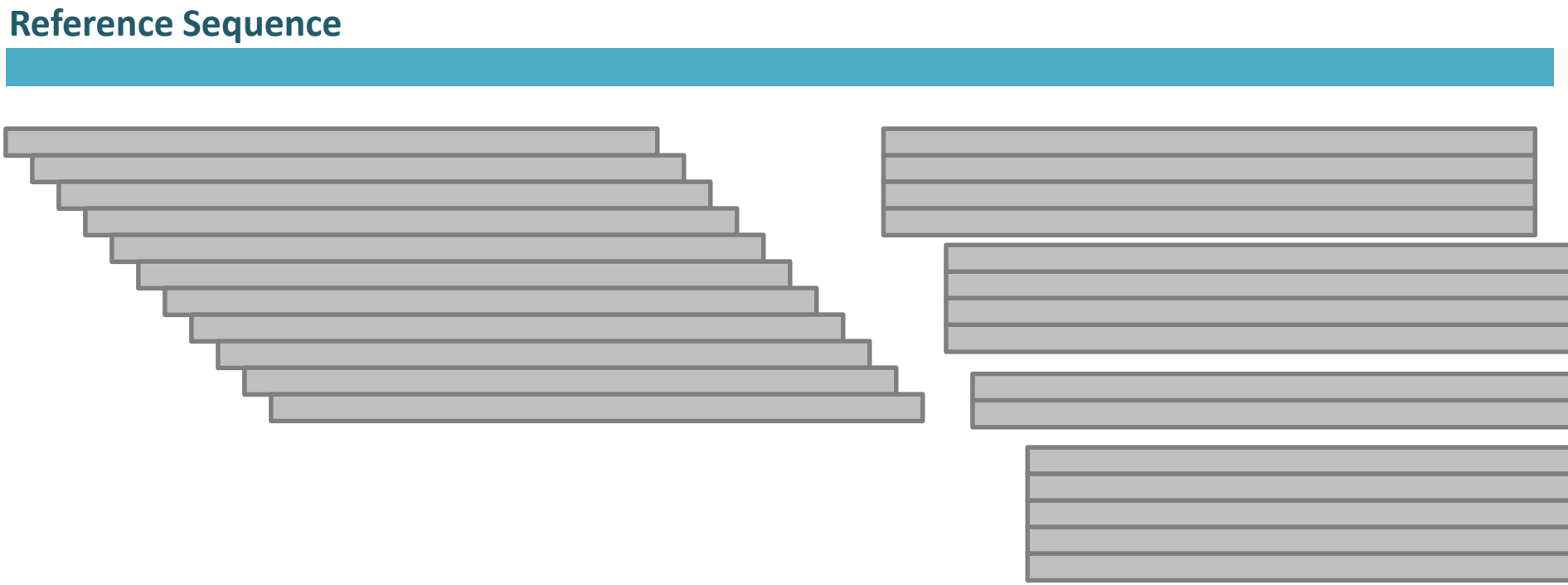


3- Bam refinement – Duplicate removal

PCR is used during library preparation.

This can result in duplicate DNA fragments in the final library prep.

What we want: information from independent fragments
What we do not want: copies of the same information coming from one fragment

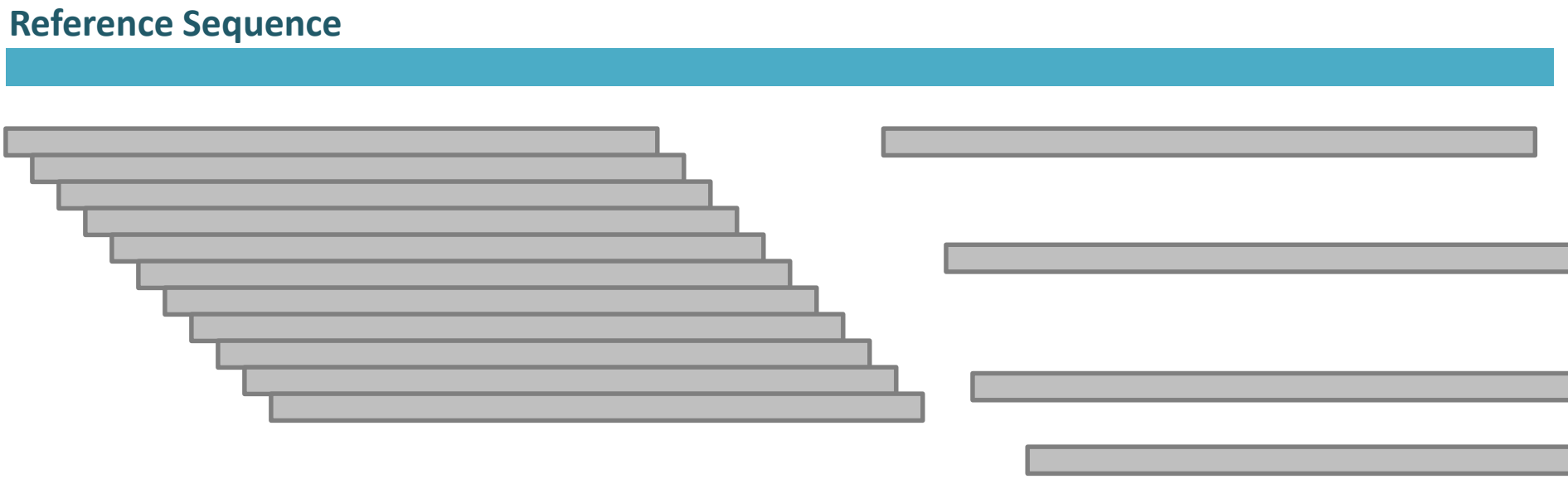


3- Bam refinement – Duplicate removal

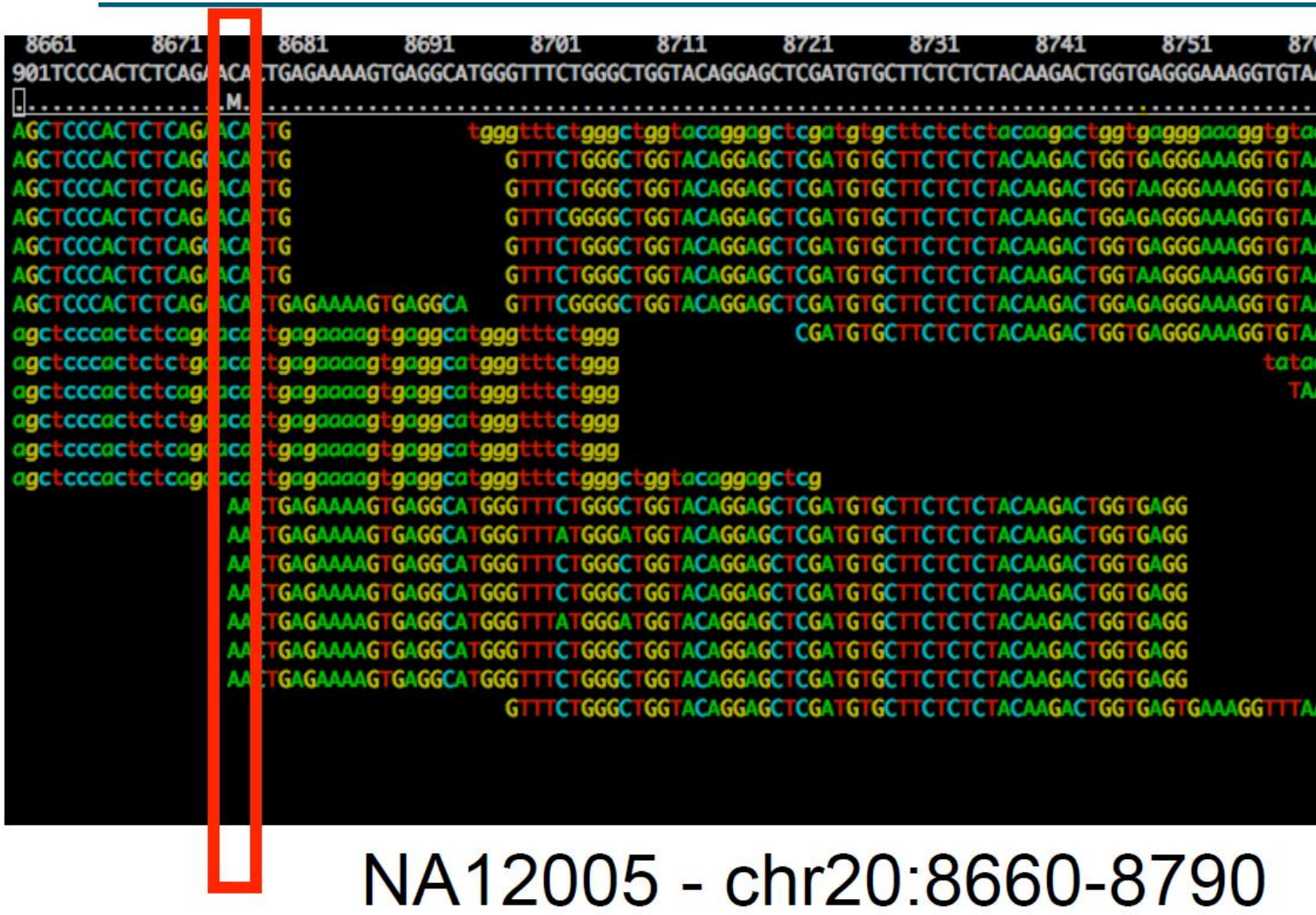
PCR is used during library preparation.

This can result in duplicate DNA fragments in the final library prep.

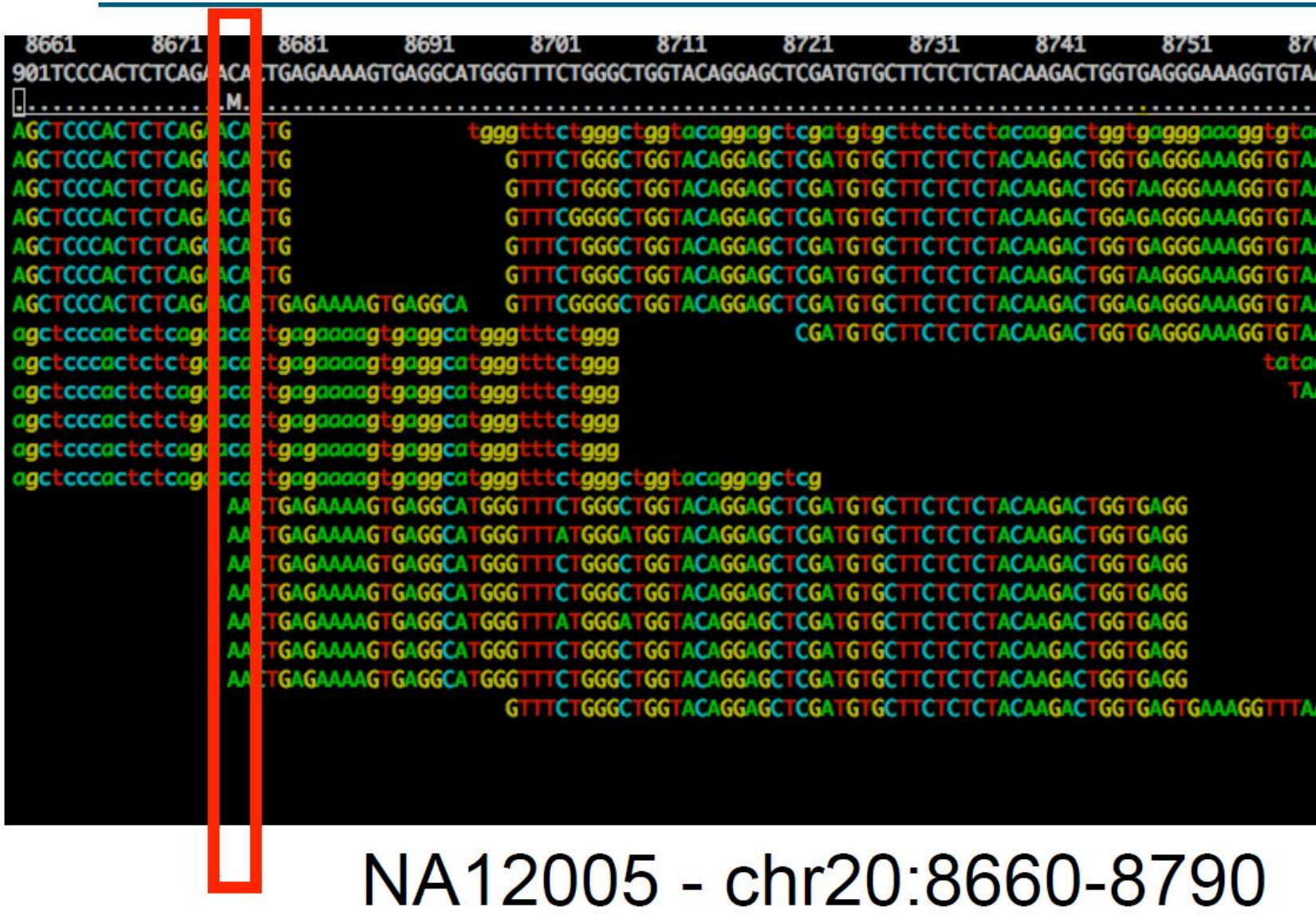
What we want: information from independent fragments
What we do not want: copies of the same information coming from one fragment



3- Bam refinement – Duplicate removal



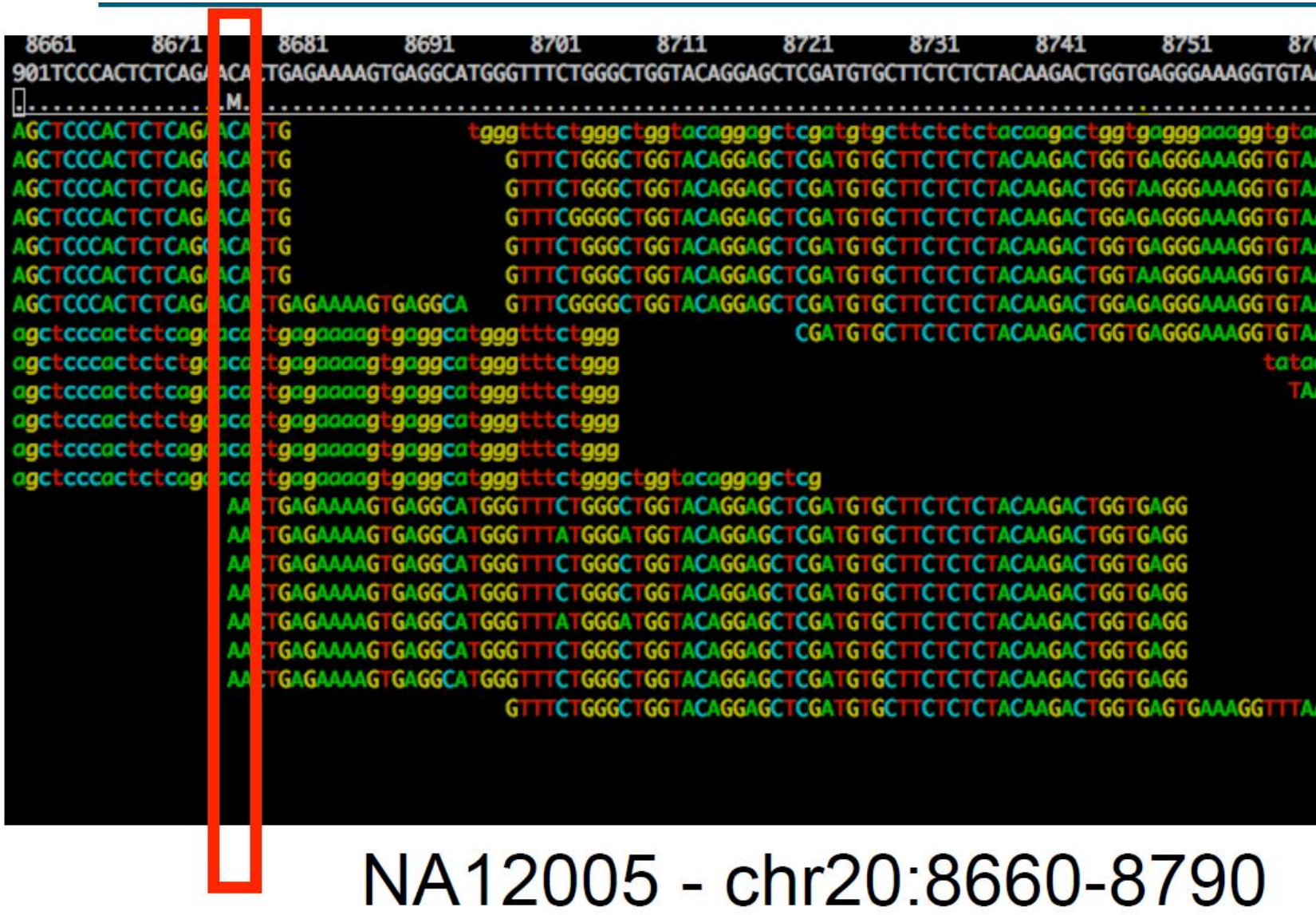
3- Bam refinement – Duplicate removal



Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal

C

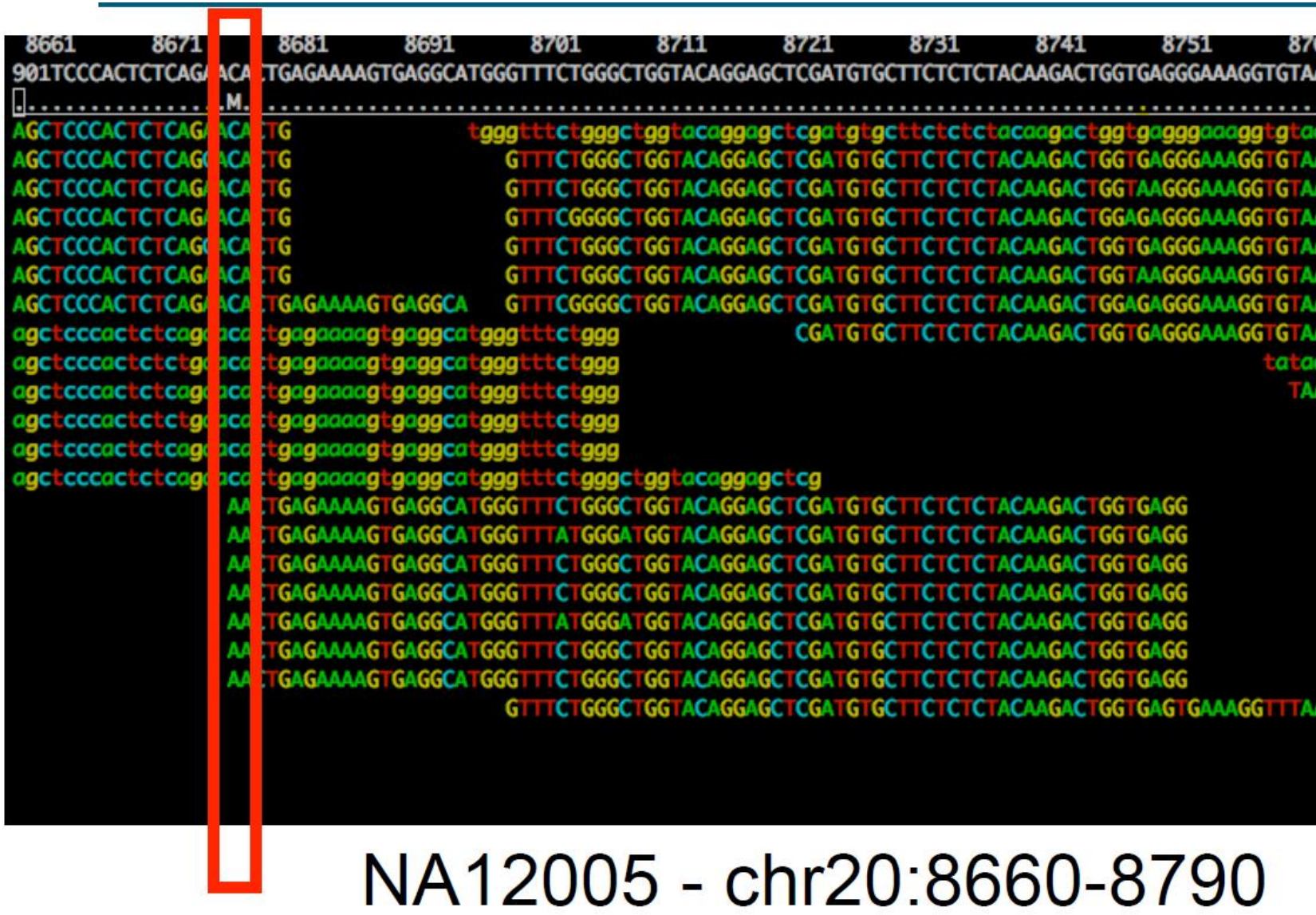


Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal

C

C



Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal

Possible heterozygote, SNP call

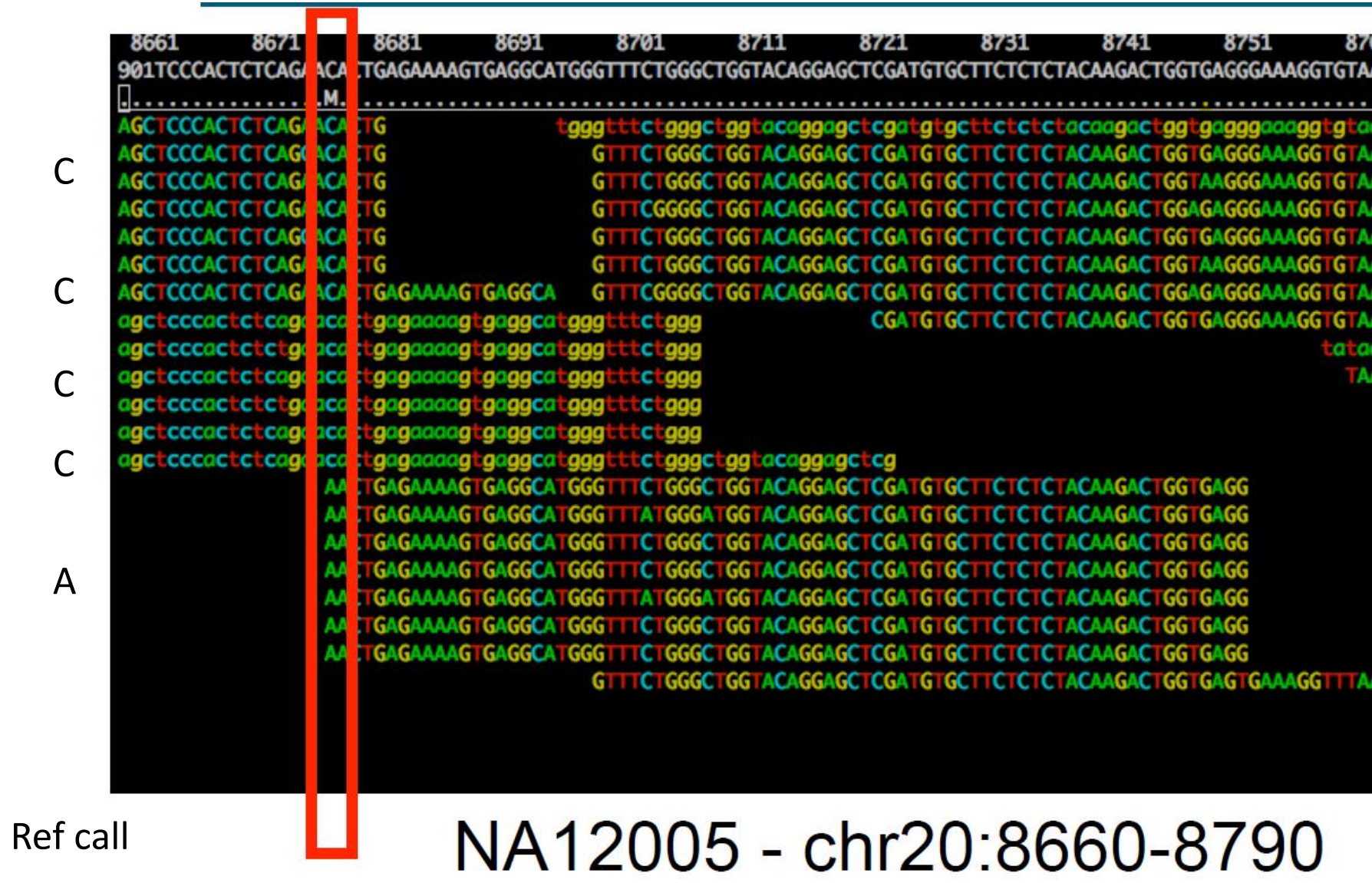
3- Bam refinement – Duplicate removal

Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal

Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal



Possible heterozygote, SNP call

3- Bam refinement – Duplicate removal

Why is it important to do it?

- It can result in false SNPs calls.

3- Bam refinement – Duplicate removal

Why is it important to do it?

- It can result in false SNPs calls.
- Duplicates may fake a high coverage thus giving high support to some variants.

3- Bam refinement – Duplicate removal

Why is it important to do it?

- It can result in false SNPs calls.
- Duplicates may fake a high coverage thus giving high support to some variants.

- PCR-free protocols exist but require a large amount of DNA.

3- Bam refinement – Duplicate removal

Number of duplicates varies according to the complexity of the library:

- whole genome experiments (<5%)
- custom enrichment ones (<30%)

3- Bam refinement – Duplicate removal

Number of duplicates varies according to the complexity of the library:

- whole genome experiments (<5%)
- custom enrichment ones (<30%)

It must be done after alignment and at the library level.

3- Bam refinement – Duplicate removal

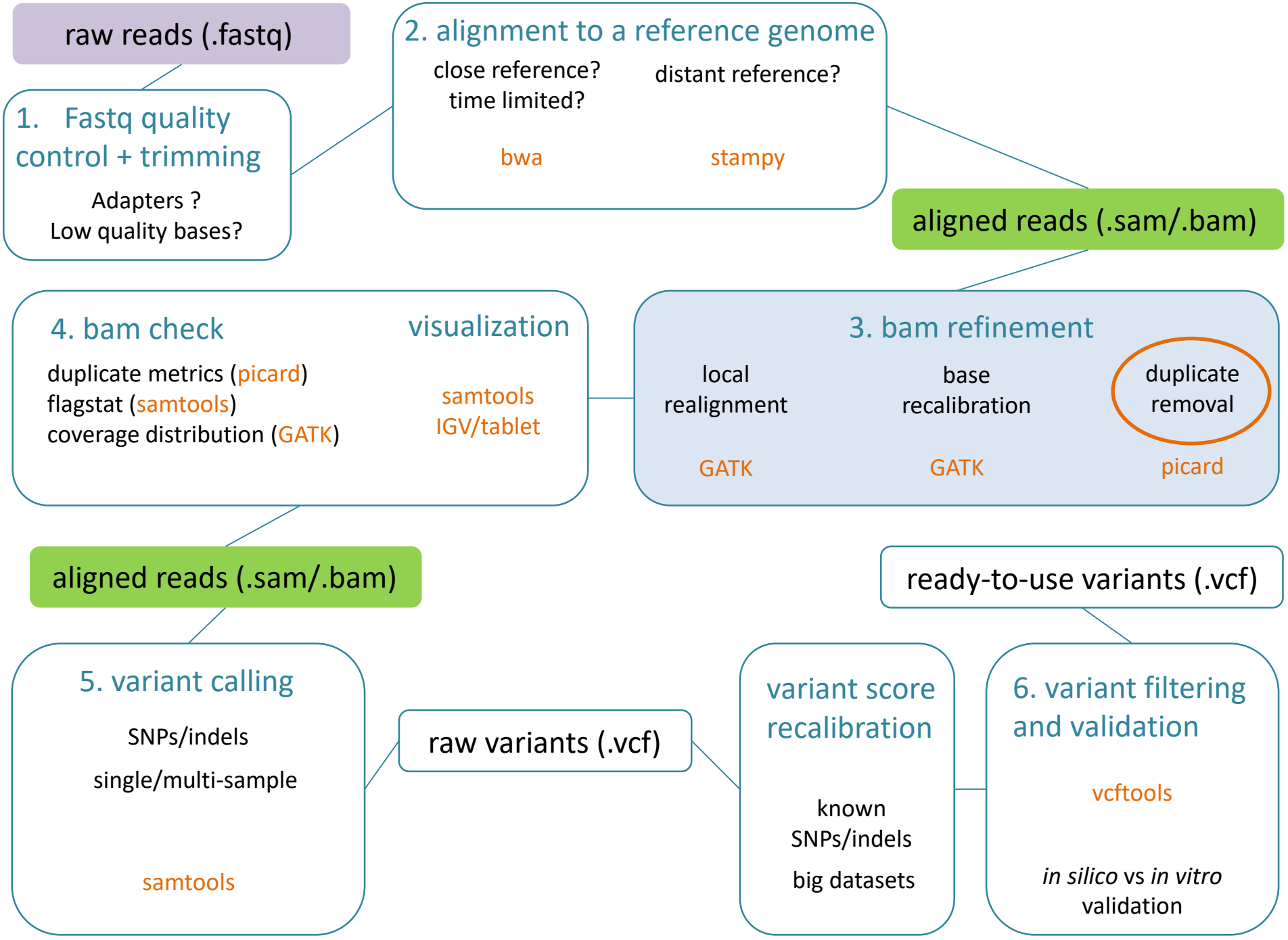
Number of duplicates varies according to the complexity of the library:

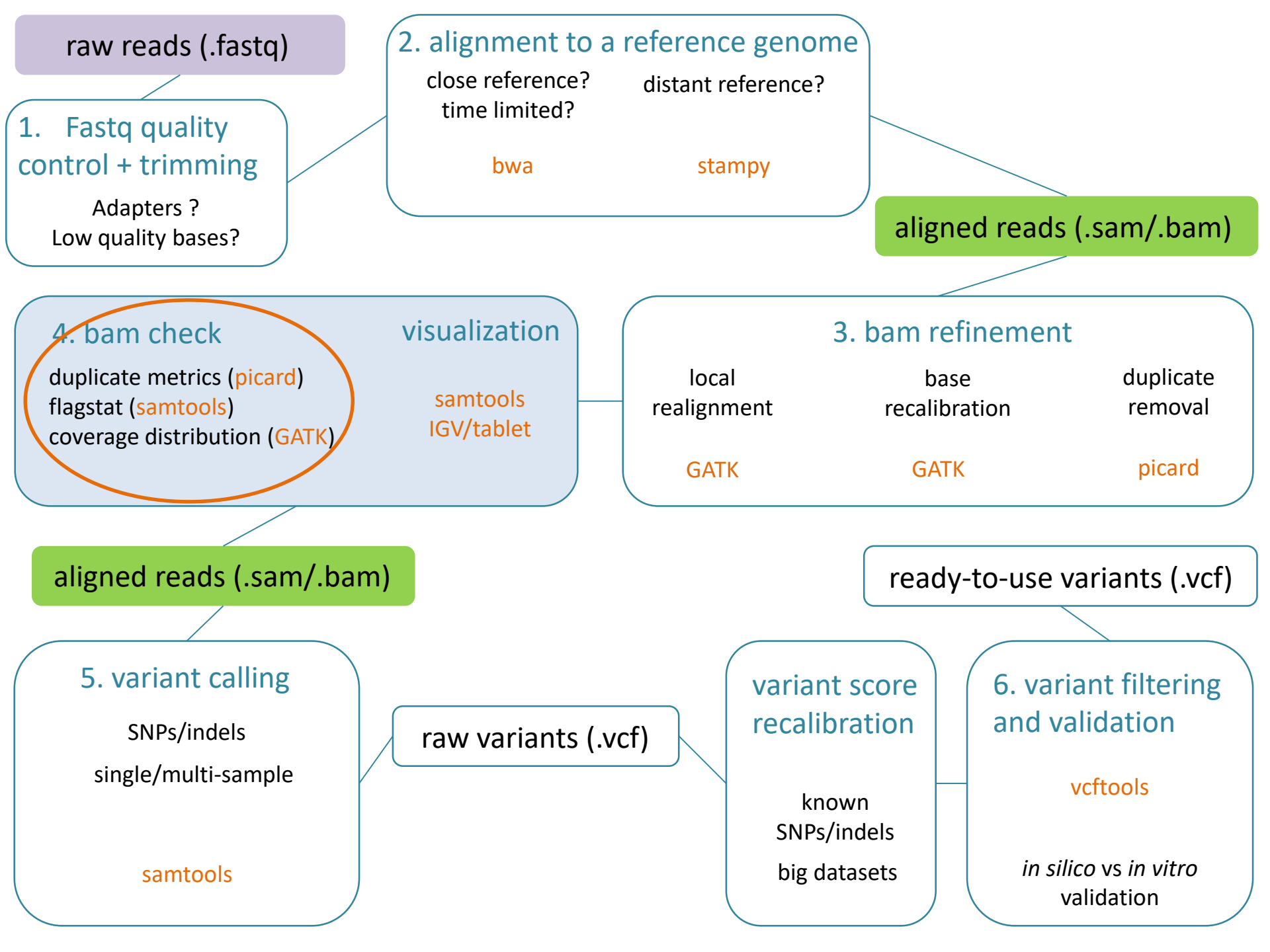
- whole genome experiments (<5%)
- custom enrichment ones (<30%)

It must be done after alignment and at the library level.

How does it work?

It identifies read-pairs where the outer ends map to the same position on the genome and removes all but one copy.


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

Alignment Stats

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

Alignment Stats

How much average coverage do I have?
Is the coverage evenly distributed along my region?

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

Alignment Stats

How much average coverage do I have?
Is the coverage evenly distributed along my region?

Coverage

3- Bam refinement – BAM check: duplicate removal

net.sf.picard.metrics.StringHeader
# net.sf.picard.sam.MarkDuplicates
INPUT=[/media/pier/pierWD/backup/freecom/work/teaching/Unife_bioinfo_122016/material/white_shark_fastq_gn2/white_shark/DD10/final_files/cc_gn2_R12_rg_sorted.bam]
OUTPUT=/media/pier/pierWD/backup/freecom/work/teaching/Unife_bioinfo_122016/material/white_shark_fastq_gn2/white_shark/DD10/final_files/cc_gn2_R12_rg_sorted_rmdup.bam
METRICS_FILE=/media/pier/pierWD/backup/freecom/work/teaching/Unife_bioinfo_122016/material/white_shark_fastq_gn2/white_shark/DD10/final_files/dupl_metrics.txt PROGRAM_RECORD_ID=MarkDuplicates
PROGRAM_GROUP_NAME=MarkDuplicates REMOVE_DUPLICATES=false ASSUME_SORTED=false
MAX_SEQUENCES_FOR_DISK_READ_ENDS_MAP=50000 MAX_FILE_HANDLES_FOR_READ_ENDS_MAP=8000
SORTING_COLLECTION_SIZE_RATIO=0.25 READ_NAME_REGEX=[a-zA-Z0-9]+:[0-9]:([0-9]+):([0-9]+):([0-9]+).*
OPTICAL_DUPLICATE_PIXEL_DISTANCE=100 VERBOSITY=INFO QUIET=false VALIDATION_STRINGENCY=STRICT
COMPRESSION_LEVEL=5 MAX_RECORDS_IN_RAM=500000 CREATE_INDEX=false CREATE_MD5_FILE=false
net.sf.picard.metrics.StringHeader
Started on: Thu Nov 10 18:37:26 GMT 2016

METRICS CLASS net.sf.picard.sam.DuplicationMetrics

LIBRARY	UNPAIRED_READS_EXAMINED	READ_PAIRS_EXAMINED	UNMAPPED_READS	UNPAIRED_READ_DUPLICATES	READ_PAIR_DUPLICATES	READ_PAIR_OPTICAL_DUPLICATES	PERCENT_DUPLICATION	ESTIMATED_LIBRARY_SIZE
cc_gn2	12953	53727	379573	7026	8687	8687	**0.202646**	

We have 20.3% of PCR duplicates in our experiment

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

Alignment Stats

How much average coverage do I have?
Is the coverage evenly distributed along my region?

Coverage

3- Bam refinement – BAM check: alignment stats

BEFORE

509594 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
130021 + 0 mapped (25.51%:-nan%)
509594 + 0 paired in sequencing
256086 + 0 read1
253508 + 0 read2
100420 + 0 properly paired (19.71%:-nan%)
115180 + 0 with itself and mate mapped
14841 + 0 singletons (2.91%:-nan%)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

AFTER

509594 + 0 in total (QC-passed reads + QC-failed reads)
24400 + 0 duplicates
130021 + 0 mapped (25.51%:-nan%)
509594 + 0 paired in sequencing
256086 + 0 read1
253508 + 0 read2
100420 + 0 properly paired (19.71%:-nan%)
115180 + 0 with itself and mate mapped
14841 + 0 singletons (2.91%:-nan%)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5

3- Bam refinement – BAM check: alignment stats

24400 + 0 duplicates

3- Bam refinement – BAM check: alignment stats

24400 + 0 duplicates

LIBRARY	UNPAIRED_READ_DUPLICATES	READ_PAIR_DUPLICATES	PERCENT_DUPLICATION
cc_gn2	7026	8687	**0.202646**

3- Bam refinement – BAM check: alignment stats

24400 + 0 duplicates

LIBRARY	UNPAIRED_READ_DUPLICATES	READ_PAIR_DUPLICATES	PERCENT_DUPLICATION
cc_gn2	7026	8687	**0.202646**

(8687*2) + 7026 = 24400

3- Bam refinement – BAM check: alignment stats

24400 + 0 duplicates

LIBRARY	UNPAIRED_READ_DUPLICATES	READ_PAIR_DUPLICATES	PERCENT_DUPLICATION
cc_gn2	7026	8687	**0.202646**

(8687*2) + 7026 = 2400

3- Bam refinement – BAM check

BAM check gives us answers to several questions:

How many duplicates do I have?
Is that reasonable for my experiment?

Duplicate removal

How many of my reads mapped back to the reference?
How many of these are paired in mapping?
How many pairs are mapped to different chromosomes?

Alignment Stats

How much average coverage do I have?
Is the coverage evenly distributed along my region?

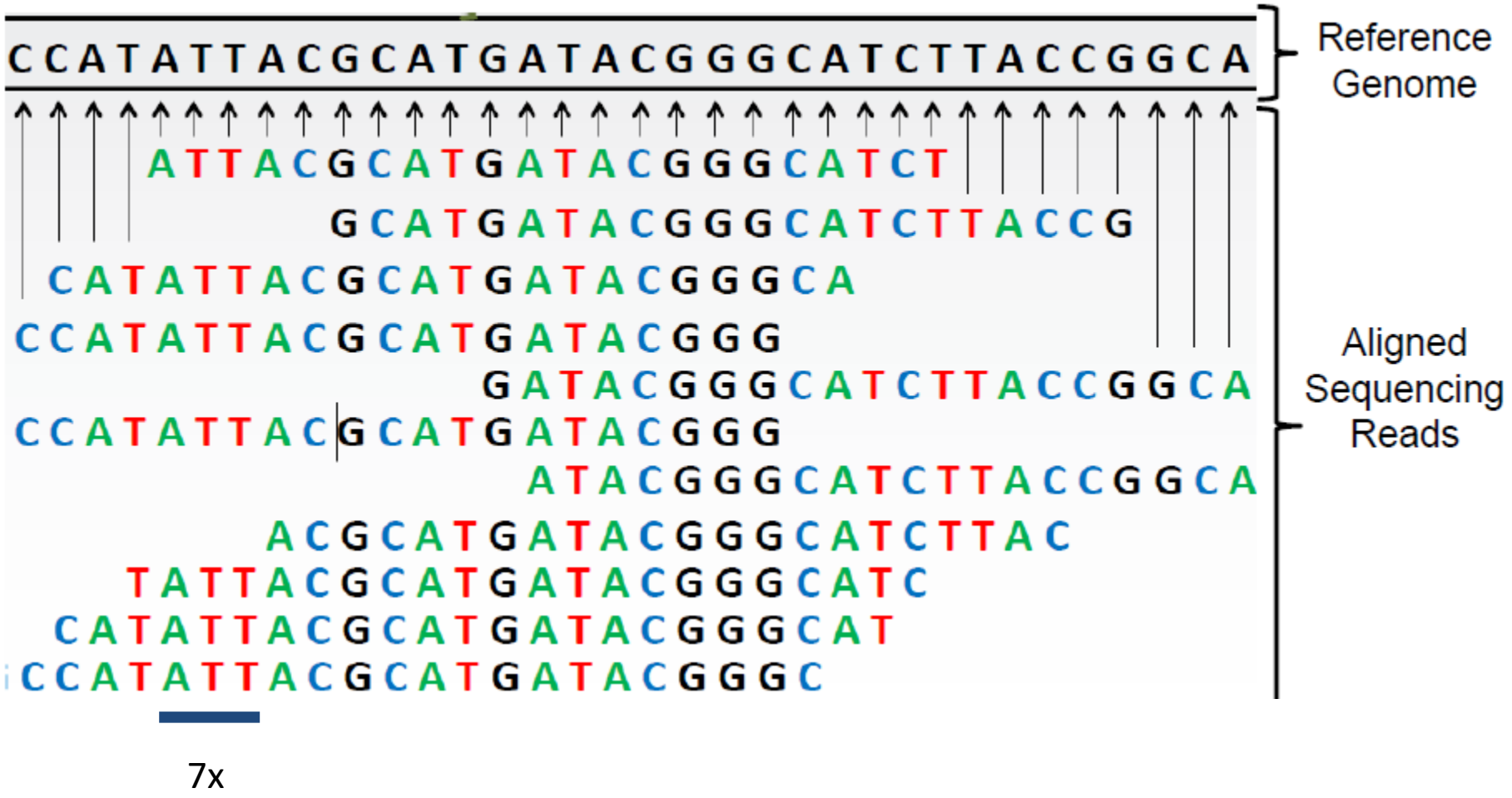
Coverage

3- Bam refinement – BAM check: coverage estimation

Depth of Coverage – The number of reads that spans a given DNA sequence of interest. This is commonly expressed in terms of “Yx” where “Y” is the number of reads and “x” is the unit reflecting the depth of coverage metric (i.e. 5x, 10x, 20x, 100x)

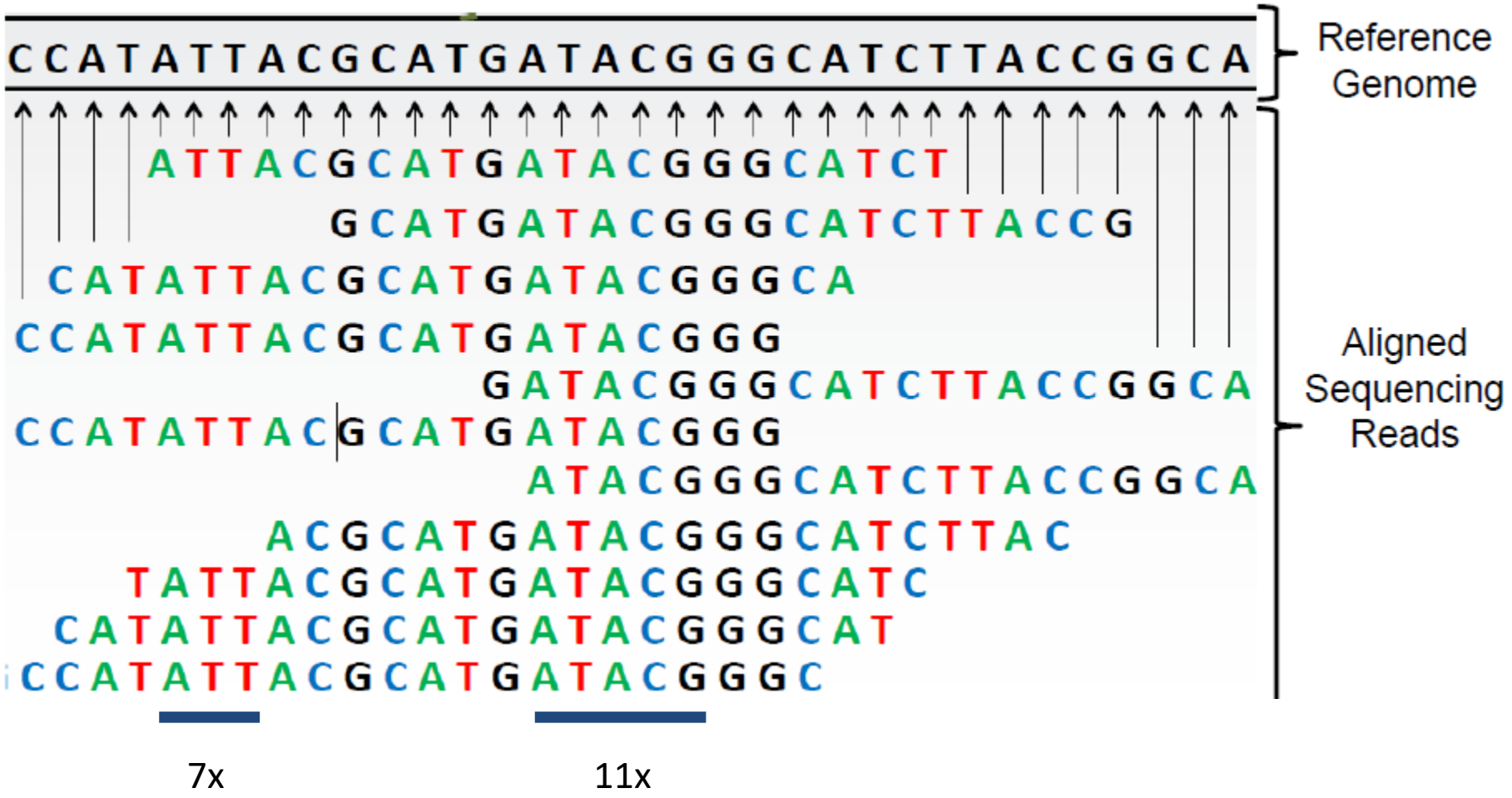
3- Bam refinement – BAM check: coverage estimation

Depth of Coverage – The number of reads that spans a given DNA sequence of interest. This is commonly expressed in terms of "Yx" where "Y" is the number of reads and "x" is the unit reflecting the depth of coverage metric (i.e. 5x, 10x, 20x, 100x)



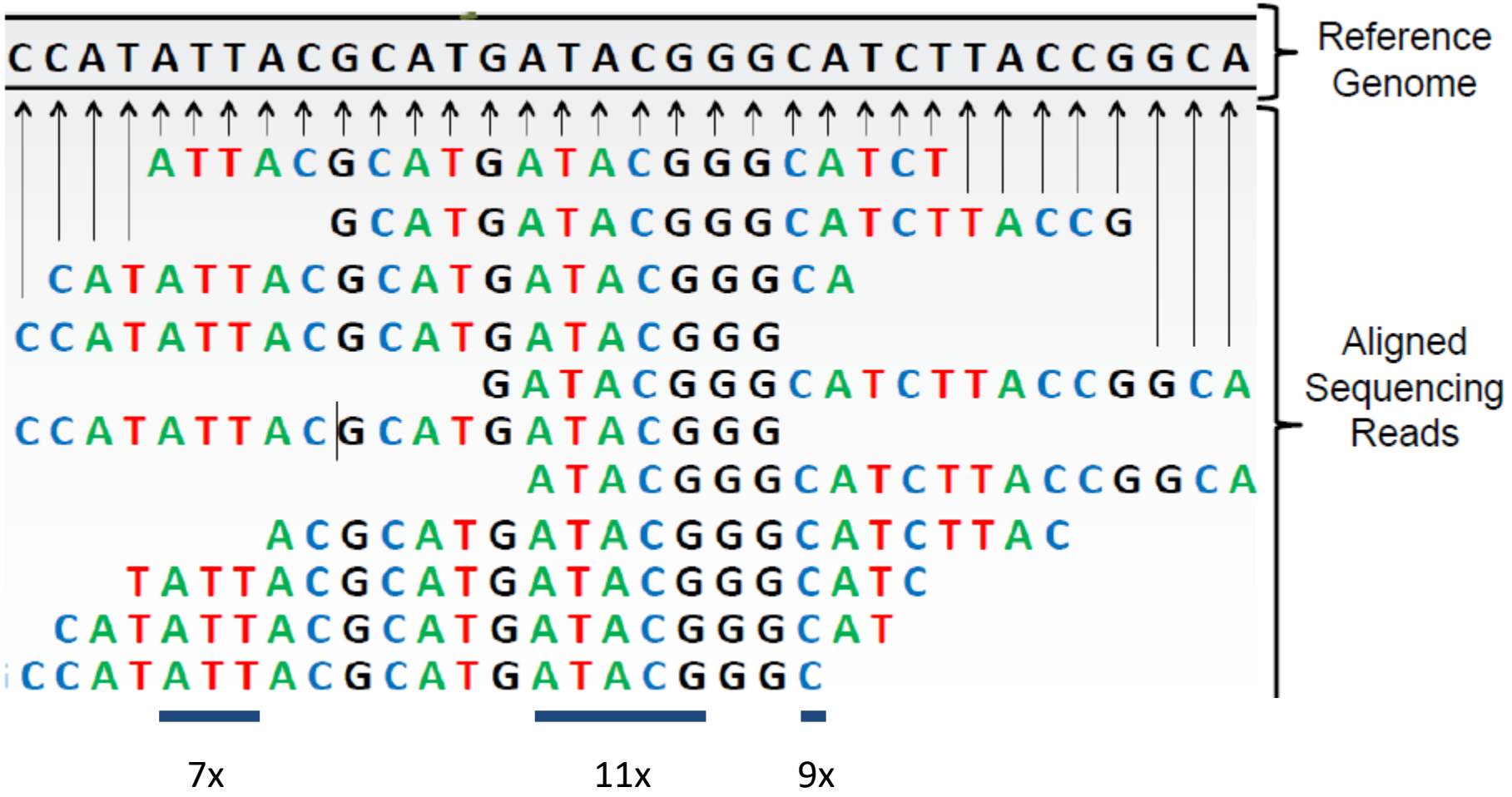
3- Bam refinement – BAM check: coverage estimation

Depth of Coverage – The number of reads that spans a given DNA sequence of interest. This is commonly expressed in terms of "Yx" where "Y" is the number of reads and "x" is the unit reflecting the depth of coverage metric (i.e. 5x, 10x, 20x, 100x)

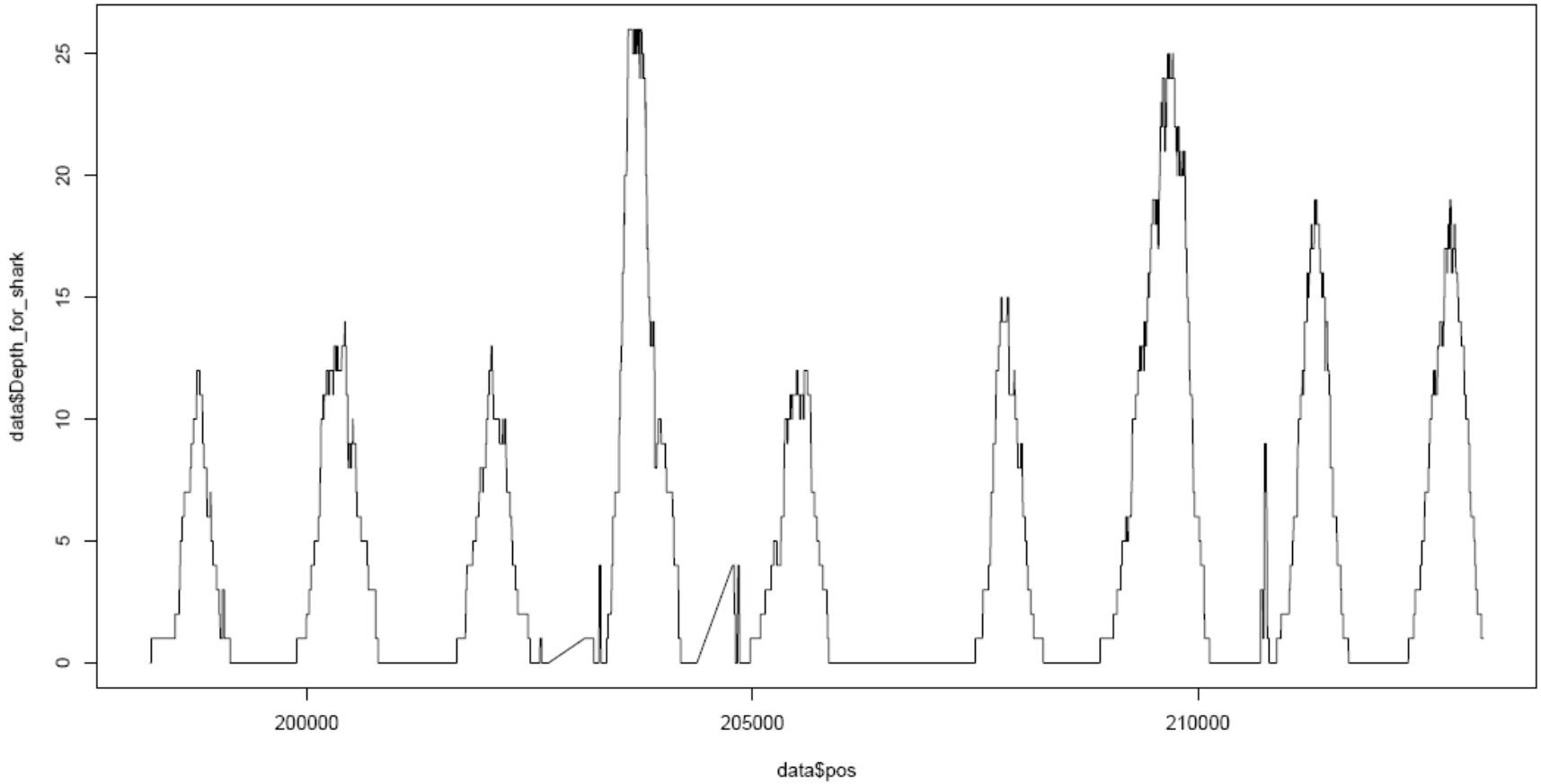


3- Bam refinement – BAM check: coverage estimation

Depth of Coverage – The number of reads that spans a given DNA sequence of interest. This is commonly expressed in terms of "Yx" where "Y" is the number of reads and "x" is the unit reflecting the depth of coverage metric (i.e. 5x, 10x, 20x, 100x)



3- Bam refinement – BAM check: coverage estimation



3- Bam refinement – BAM check: coverage estimation

Why is important to check the coverage?

- To check how your experiment performed (one of the ways to assess the quality of your experiment);

3- Bam refinement – BAM check: coverage estimation

Why is important to check the coverage?

- To check how your experiment performed (one of the ways to assess the quality of your experiment);
- To understand how confident can you be with your data;

3- Bam refinement – BAM check: coverage estimation

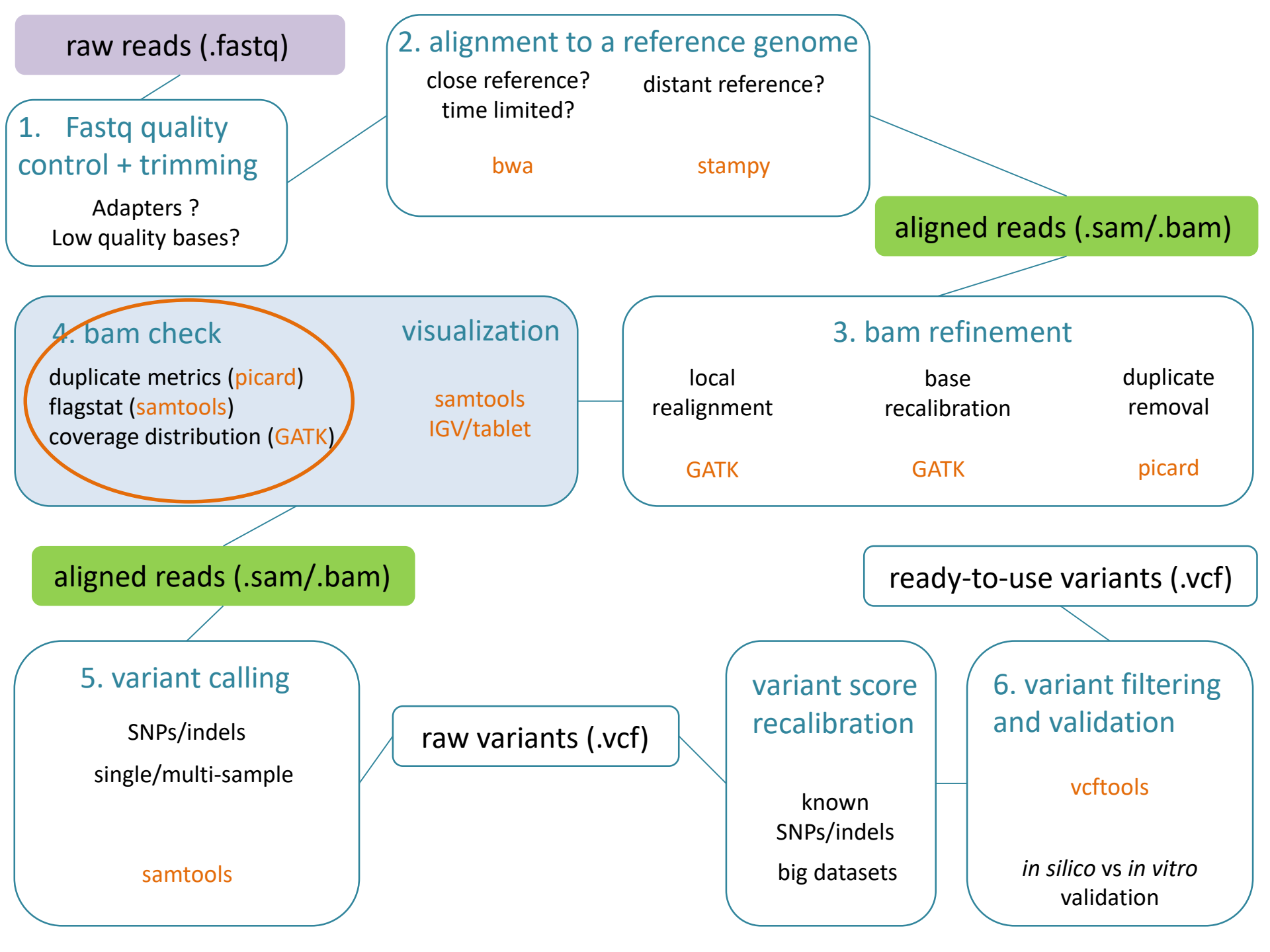
Why is important to check the coverage?

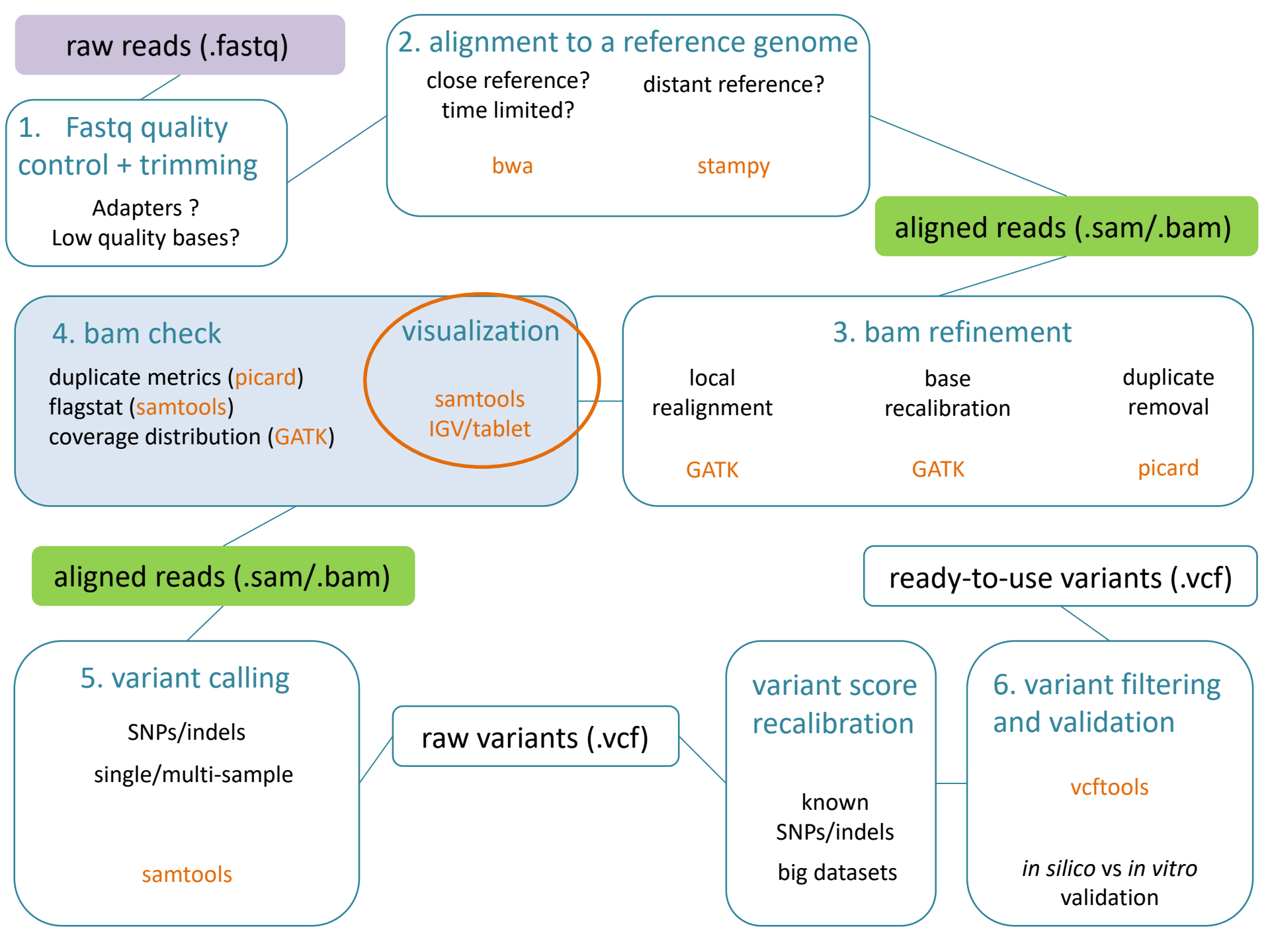
- To check how your experiment performed (one of the ways to assess the quality of your experiment);
- To understand how confident can you be with your data;
- To decide on filtering after variant calling;

3- Bam refinement – BAM check: coverage estimation

Why is important to check the coverage?

- To check how your experiment performed (one of the ways to assess the quality of your experiment);
- To understand how confident can you be with your data;
- To decide on filtering after variant calling;
- To look for structural variation.


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

4- BAM check: visualisation

Different tools to visualise aligned NGS data:

- IGV: https://www.broadinstitute.org/igv/;
- Tablet: https://ics.hutton.ac.uk/tablet/;
- …

4- BAM check: visualisation

Different tools to visualise aligned NGS data:

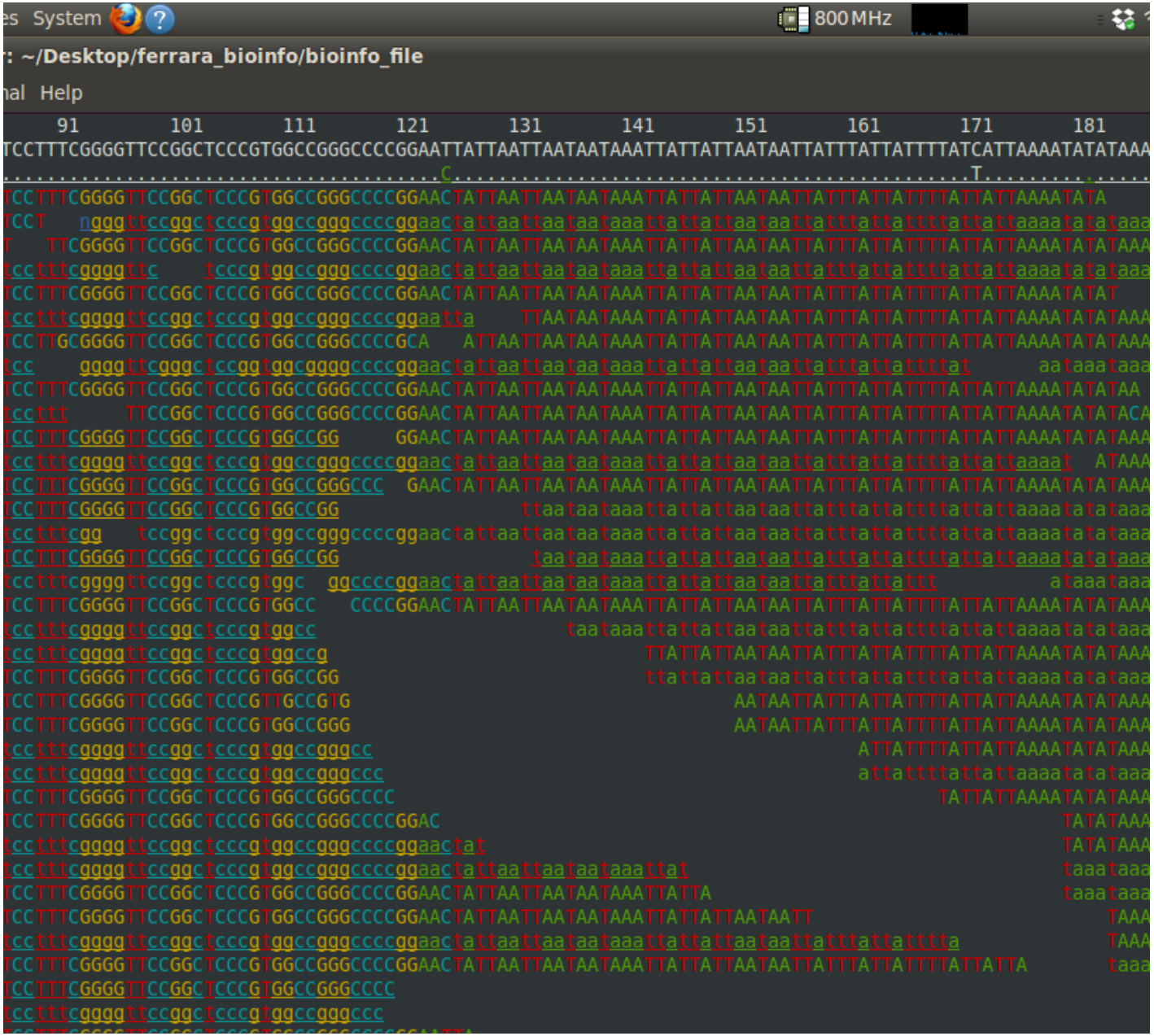
- IGV: https://www.broadinstitute.org/igv/;
- Tablet: https://ics.hutton.ac.uk/tablet/;
- …
- Samtools tview

4- BAM check: visualisation

Samtools tview:

- Basic visualization tool;
- Terminal based;
- No need of RAM/Java/extra packages;

4- BAM check: visualisation

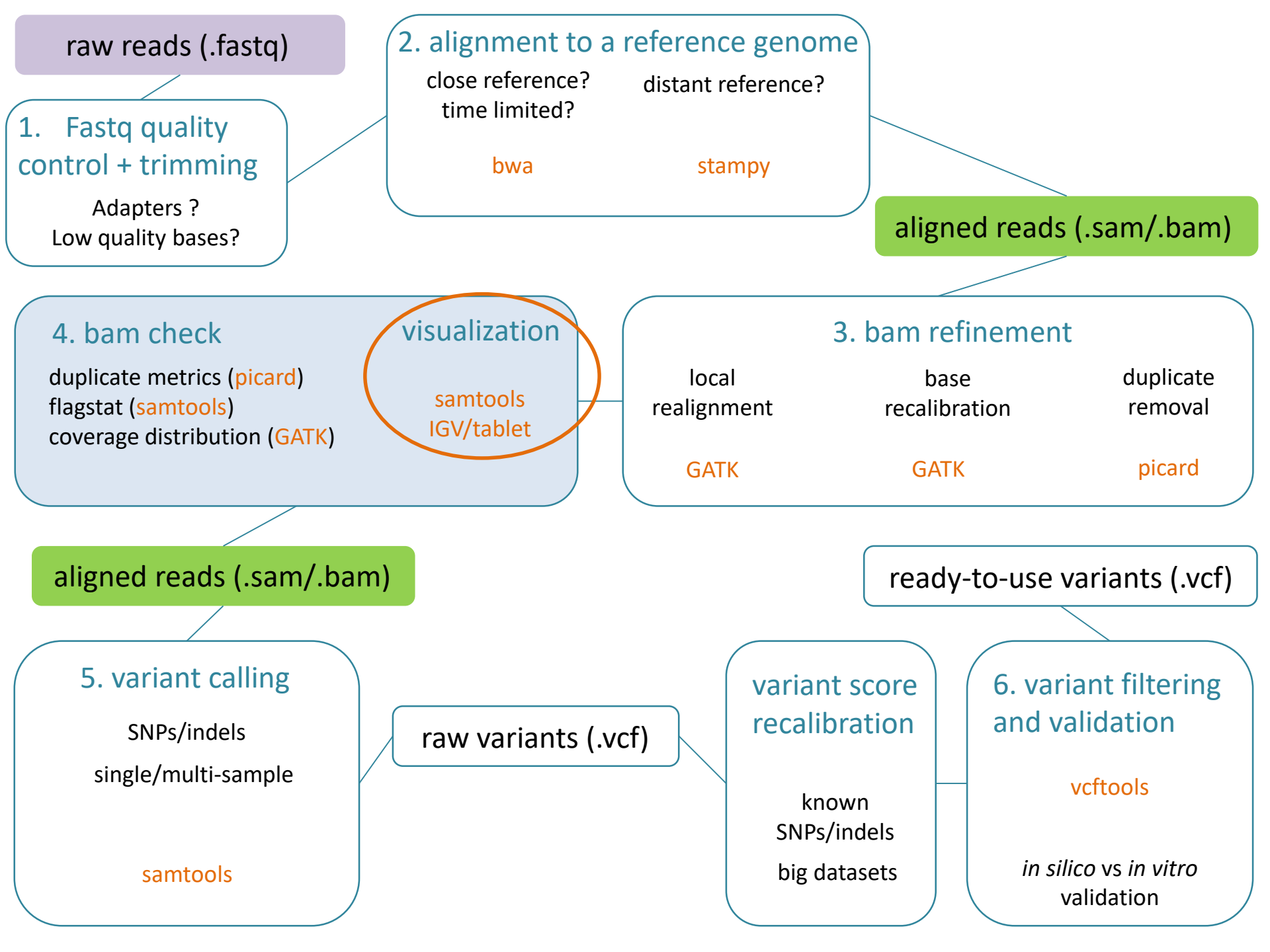


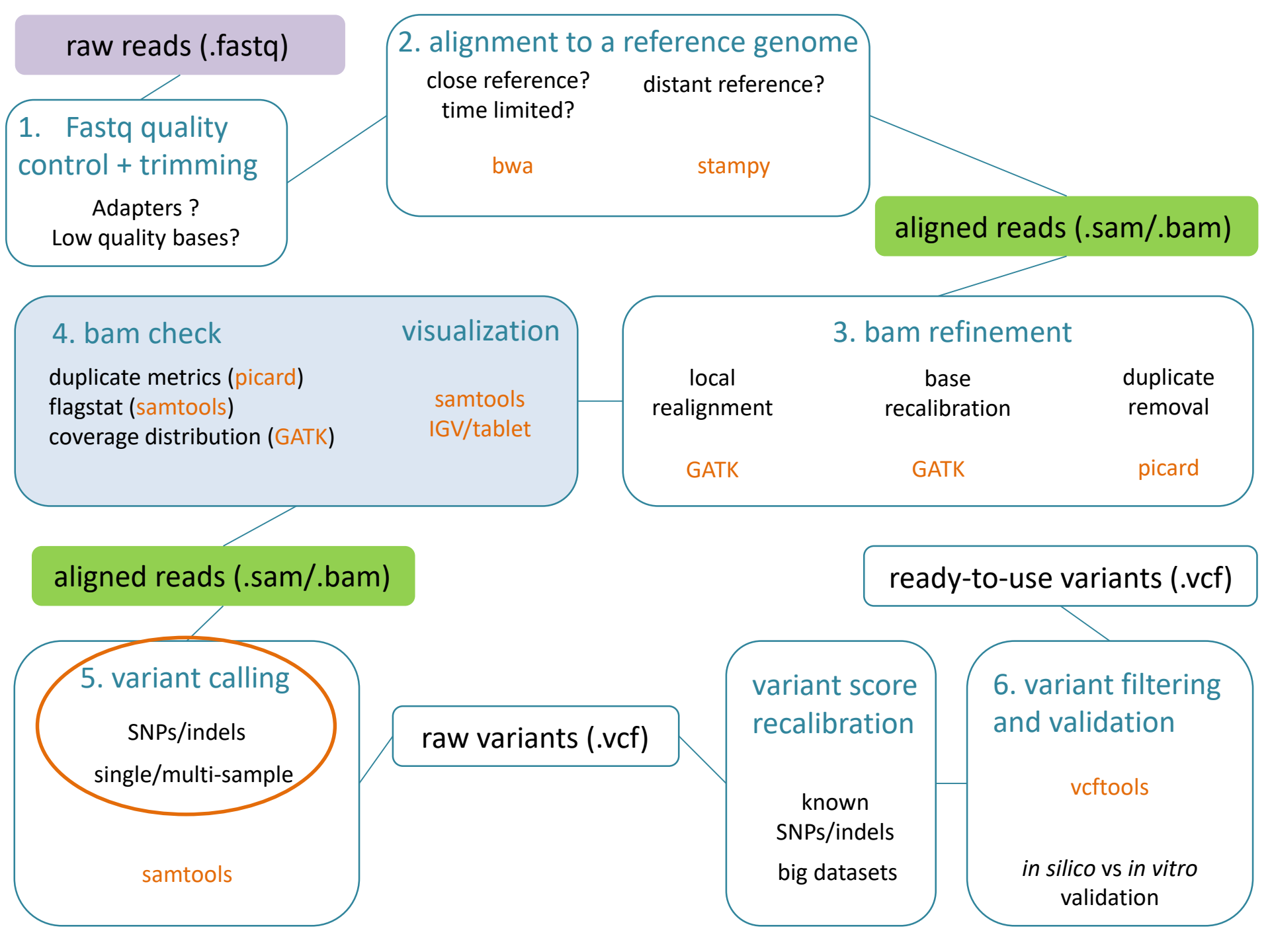
4- BAM check: visualisation

IGV

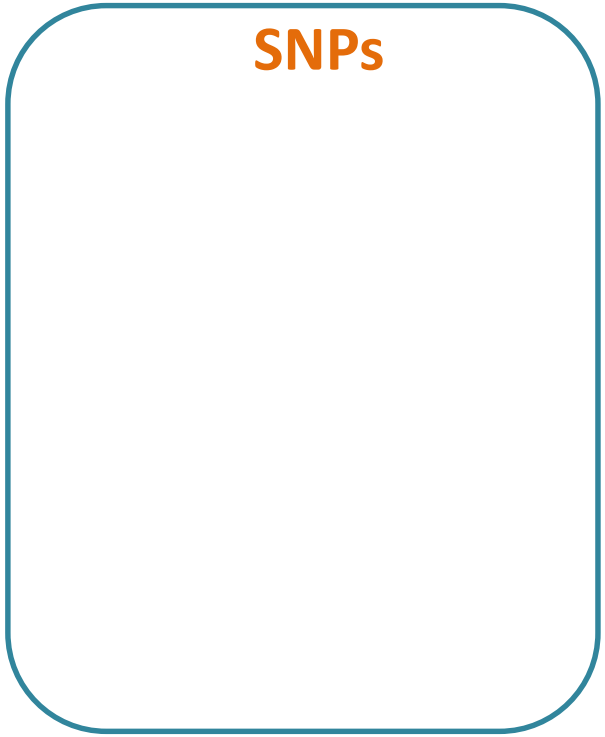
4- BAM check: visualisation

IGV

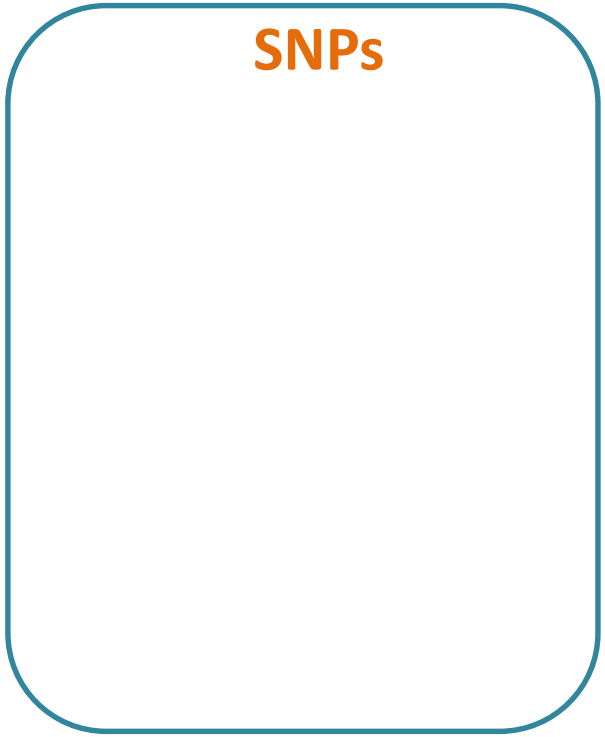

raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

5- Variant calling



5- Variant calling



SNPs: Single Nucleotide Polymorphisms

5- Variant calling

SNPs

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

SNPs: Single Nucleotide Polymorphisms

5- Variant calling

SNPs

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

indels

SNPs: Single Nucleotide Polymorphisms

5- Variant calling

SNPs

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

indels

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions

5- Variant calling

SNPs

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

indels

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

Dindel

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions

5- Variant calling

SNPs	indels	SV
samtools GATK: 1. Unified Genotyper 2. Haplotype caller	samtools GATK: 1. Unified Genotyper 2. Haplotype caller Dindel	

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions

5- Variant calling

SNPs	indels	SV
samtools GATK: 1. Unified Genotyper 2. Haplotype caller	samtools GATK: 1. Unified Genotyper 2. Haplotype caller Dindel	

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions
SV: Structural Variation

5- Variant calling

SNPs	indels	SV
samtools GATK: 1. Unified Genotyper 2. Haplotype caller	samtools GATK: 1. Unified Genotyper 2. Haplotype caller Dindel	SVMerge – pipeline combining many different tools

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions
SV: Structural Variation

5- Variant calling

SNPs

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

indels

samtools

GATK:
1. Unified Genotyper
2. Haplotype caller

Dindel

SV

SVMerge – pipeline combining many different tools

SNPs: Single Nucleotide Polymorphisms
Indels: insertions/deletions
SV: Structural Variation

Variant calling:

Examine the bases aligned to position and look for differences

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

- Base call qualities of each supporting base

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

- Base call qualities of each supporting base
- Proximity to indels and homopolymer run

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

- Base call qualities of each supporting base
- Proximity to indels and homopolymer run
- Mapping qualities of the reads supporting the SNP (increased read length or paired-end help MQ scores)

Variant calling:

Examine the bases aligned to position and look for differences

What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

- Base call qualities of each supporting base
- Proximity to indels and homopolymer run
- Mapping qualities of the reads supporting the SNP (increased read length or paired-end help MQ scores)
- Sequencing depth

Variant calling:

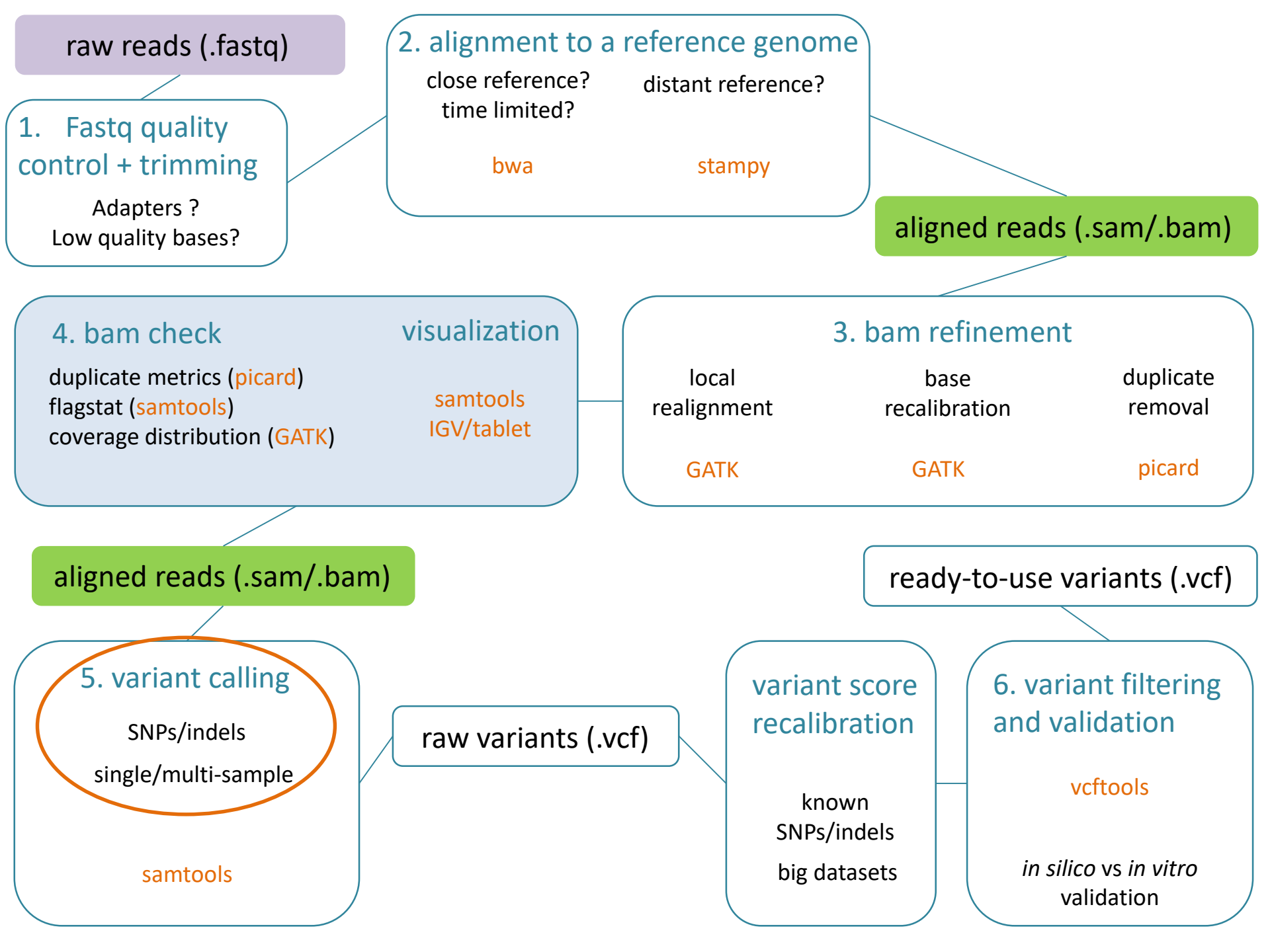
Examine the bases aligned to position and look for differences

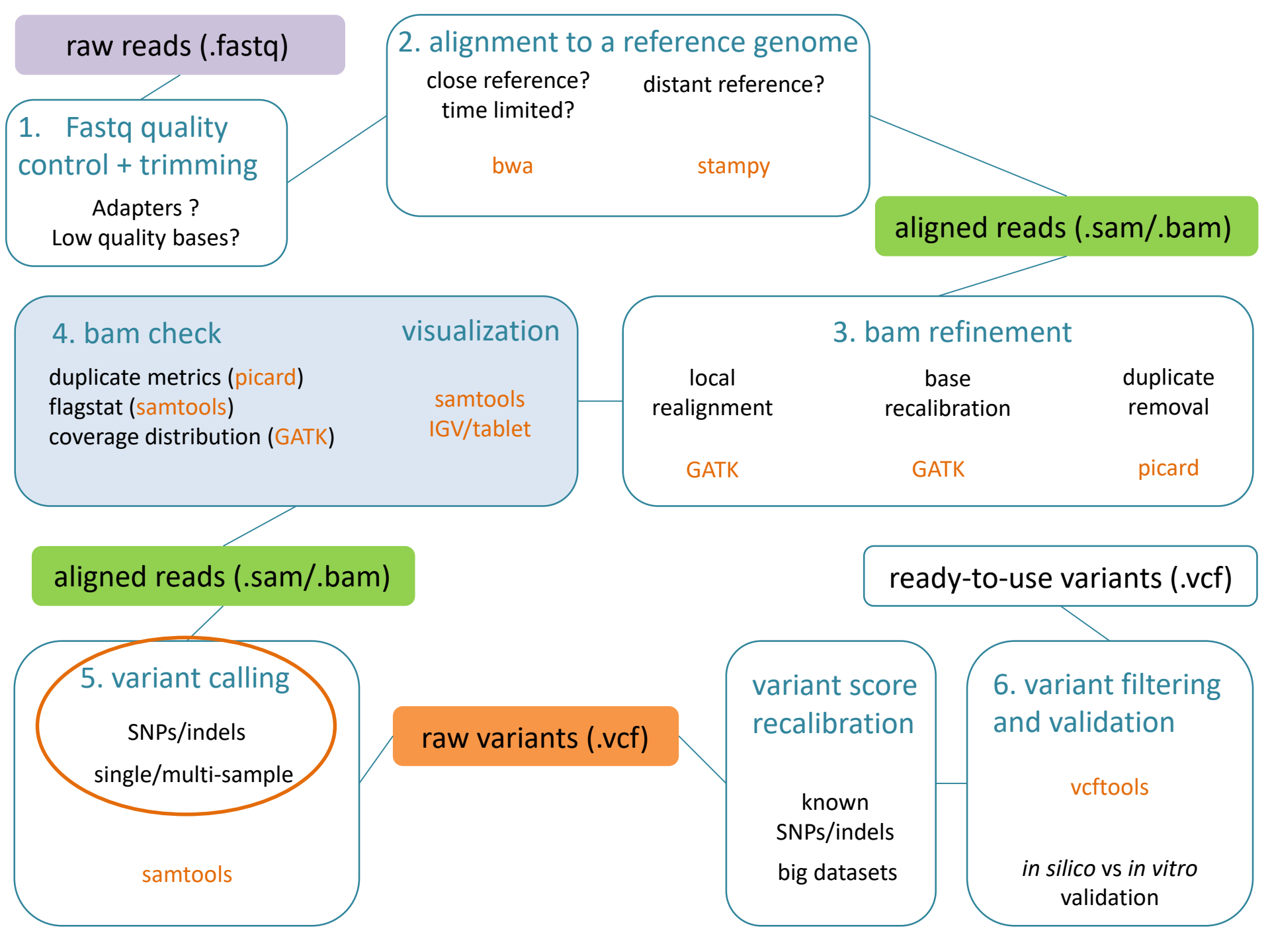
What we are looking for:

Polymorphic sites / monomorphic sites

Factors to consider:

- Base call qualities of each supporting base
- Proximity to indels and homopolymer run
- Mapping qualities of the reads supporting the SNP (increased read length or paired-end help MQ scores)
- Sequencing depth
- Individual vs multi-sample calling


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

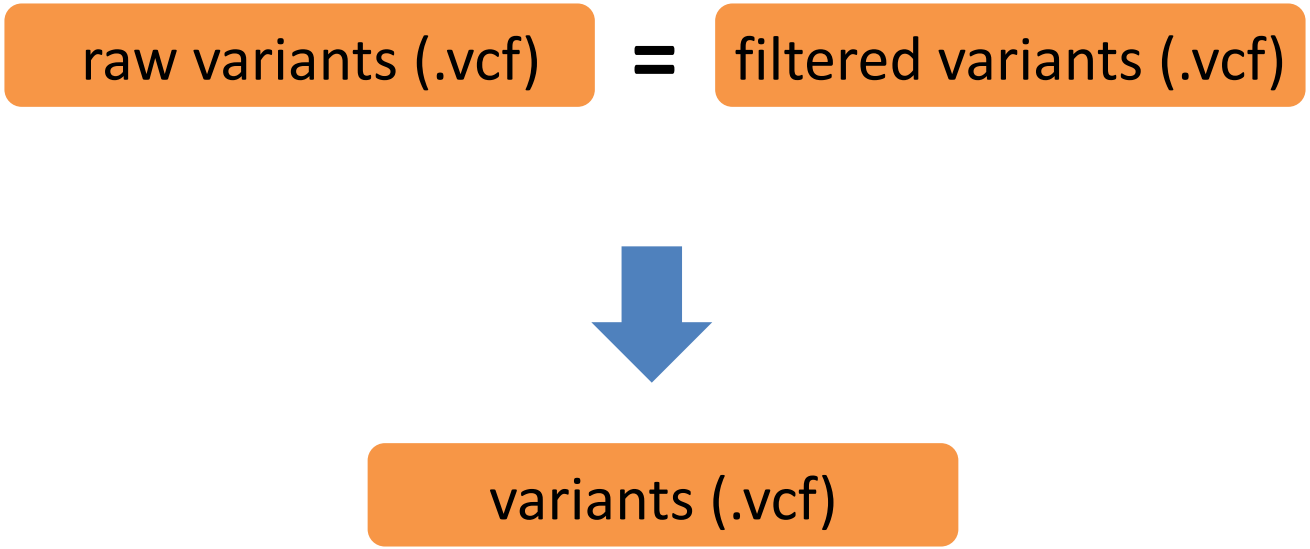
About the variant file:

raw variants (.vcf)

About the variant file:

raw variants (.vcf) = filtered variants (.vcf)

About the variant file:



variants (.vcf)

Standardised format for storing DNA polymorphism data

- SNPs, indels, SV
- Rich annotations

variants (.vcf)

Standardised format for storing DNA polymorphism data

- SNPs, indels, SV
- Rich annotations

Can be indexed for fast data retrieval of variants from a range of positions

variants (.vcf)

Standardised format for storing DNA polymorphism data

- SNPs, indels, SV
- Rich annotations

Can be indexed for fast data retrieval of variants from a range of positions

Can store variant information over many samples

variants (.vcf)

Standardised format for storing DNA polymorphism data

- SNPs, indels, SV
- Rich annotations

Can be indexed for fast data retrieval of variants from a range of positions

Can store variant information over many samples

Record meta-data about the site

- dbSNP accession, filter status

variants (.vcf)

Standardised format for storing DNA polymorphism data

- SNPs, indels, SV
- Rich annotations

Can be indexed for fast data retrieval of variants from a range of positions

Can store variant information over many samples

Record meta-data about the site

- dbSNP accession, filter status

Very flexible

- Tags can be introduced to describe new types of variants
- Different VCF files may contain different information/annotations

variants (.vcf)

9samples_scatter_MQ30_BQ20_ref2...

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##samtoolsVersion=1.1+htslib-1.1
##samtoolsCommand=samtools mpileup -AI -Q 20 -q 30 --output-tags DP,DV,DPR,INFO/DPR,SP -g -O -s -f /media/p
##reference=file:///media/pier/DATA/sharks_data/reference/refseq_patched_1077exons_introns_nogaps_200bpNs.f
##contig=<ID=ref_seq_exons_introns_200Ns_from_patched_ref,length=1724510>
##ALT=<ID=X,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts i
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is bett
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (big
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Number of high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="Number of high-quality non-reference bases">
##FORMAT=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##INFO=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-for
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.1+htslib-1.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3
ref_seq_exons_introns_200Ns_from_patched_ref    487     .       A       .       23.5133 .       DP=763;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    488     .       T       .       999     .       DP=774;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    489     .       G       .       999     .       DP=793;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    490     .       G       .       999     .       DP=807;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    491     .       G       .       14.8217 .       DP=818;VDB=
ref_seq_exons_introns_200Ns_from_patched_ref    492     .       C       .       23.5252 .       DP=824;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    493     .       A       .       999     .       DP=855;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    494     .       A       .       999     .       DP=869;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    495     .       T       .       999     .       DP=876;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    496     .       A       .       999     .       DP=904;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    497     .       A       .       999     .       DP=929;MQSB
```

variants (.vcf)

Header

9samples_scatter_MQ30_BQ20_ref2...

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##samtoolsVersion=1.1+htslib-1.1
##samtoolsCommand=samtools mpileup -AI -Q 20 -q 30 --output-tags DP,DV,DPR,INFO/DPR,SP -g -O -s -f /media/p
##reference=file:///media/pier/DATA/sharks_data/reference/refseq_patched_1077exons_introns_nogaps_200bpNs.f
##contig=<ID=ref_seq_exons_introns_200Ns_from_patched_ref,length=1724510>
##ALT=<ID=X,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts i
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is bett
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (big
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Number of high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="Number of high-quality non-reference bases">
##FORMAT=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##INFO=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-for
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.1+htslib-1.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3
ref_seq_exons_introns_200Ns_from_patched_ref    487     .       A       .       23.5133 .       DP=763;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    488     .       T       .       999     .       DP=774;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    489     .       G       .       999     .       DP=793;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    490     .       G       .       999     .       DP=807;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    491     .       G       .       14.8217 .       DP=818;VDB=
ref_seq_exons_introns_200Ns_from_patched_ref    492     .       C       .       23.5252 .       DP=824;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    493     .       A       .       999     .       DP=855;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    494     .       A       .       999     .       DP=869;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    495     .       T       .       999     .       DP=876;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    496     .       A       .       999     .       DP=904;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    497     .       A       .       999     .       DP=929;MQSB
```

variants (.vcf)

Header

Data

9samples_scatter_MQ30_BQ20_ref2...

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##samtoolsVersion=1.1+htslib-1.1
##samtoolsCommand=samtools mpileup -AI -Q 20 -q 30 --output-tags DP,DV,DPR,INFO/DPR,SP -g -O -s -f /media/p
##reference=file:///media/pier/DATA/sharks_data/reference/refseq_patched_1077exons_introns_nogaps_200bpNs.f
##contig=<ID=ref_seq_exons_introns_200Ns_from_patched_ref,length=1724510>
##ALT=<ID=X,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts i
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is bett
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (big
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Number of high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="Number of high-quality non-reference bases">
##FORMAT=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##INFO=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-for
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.1+htslib-1.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3
ref_seq_exons_introns_200Ns_from_patched_ref    487     .       A       .       23.5133 .       DP=763;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    488     .       T       .       999     .       DP=774;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    489     .       G       .       999     .       DP=793;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    490     .       G       .       999     .       DP=807;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    491     .       G       .       14.8217 .       DP=818;VDB=
ref_seq_exons_introns_200Ns_from_patched_ref    492     .       C       .       23.5252 .       DP=824;SGB=
ref_seq_exons_introns_200Ns_from_patched_ref    493     .       A       .       999     .       DP=855;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    494     .       A       .       999     .       DP=869;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    495     .       T       .       999     .       DP=876;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    496     .       A       .       999     .       DP=904;MQSB
ref_seq_exons_introns_200Ns_from_patched_ref    497     .       A       .       999     .       DP=929;MQSB
```

variants (.vcf)

Header

9samples_scatter_MQ30_BQ20_ref2...

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##samtoolsVersion=1.1+htslib-1.1
##samtoolsCommand=samtools mpileup -AI -Q 20 -q 30 --output-tags DP,DV,DPR,INFO/DPR,SP -g -O -s -f /media/pier/
##reference=file:///media/pier/DATA/sharks_data/reference/refseq_patched_1077exons_introns_nogaps_200bpNs.fa
##contig=<ID=ref_seq_exons_introns_200Ns_from_patched_ref,length=1724510>
##ALT=<ID=X,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts in RN
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better)">
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is better)"
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)">
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (bigger
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Number of high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="Number of high-quality non-reference bases">
##FORMAT=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##INFO=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same orde
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.1+htslib-1.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3599_S
```

variants (.vcf)

Header

Lines starting with ##: arbitrary number of meta-information lines

```
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and a
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
```

##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">

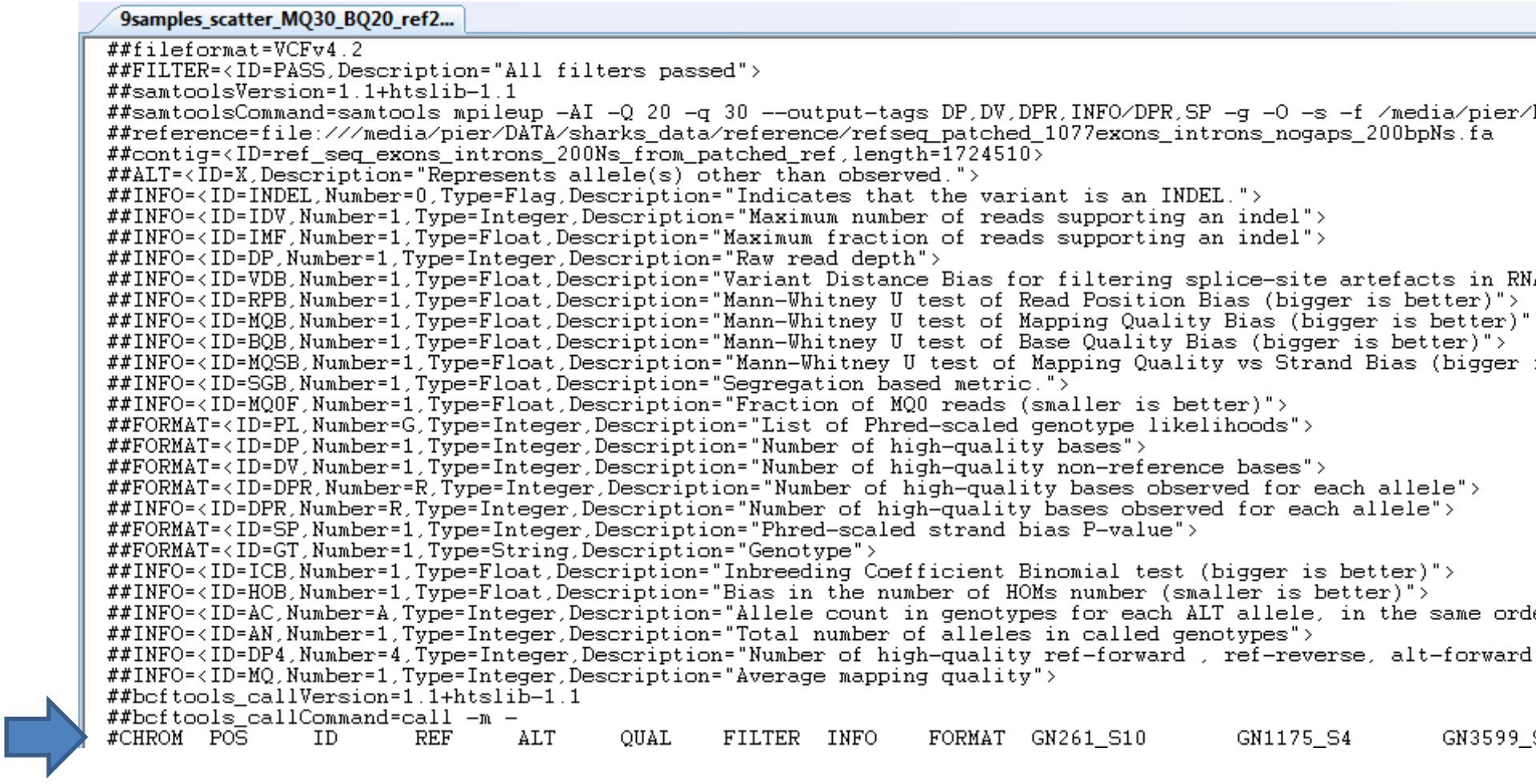
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases">

##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:



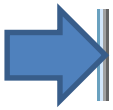
9samples_scatter_MQ30_BQ20_ref2...

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##samtoolsVersion=1.1+htslib-1.1
##samtoolsCommand=samtools mpileup -AI -Q 20 -q 30 --output-tags DP,DV,DPR,INFO/DPR,SP -g -O -s -f /media/pier/
##reference=file:///media/pier/DATA/sharks_data/reference/refseq_patched_1077exons_introns_nogaps_200bpNs.fa
##contig=<ID=ref_seq_exons_introns_200Ns_from_patched_ref,length=1724510>
##ALT=<ID=X,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts in RN
##INFO=<ID=RPB,Number=1,Type=Float,Description="Mann-Whitney U test of Read Position Bias (bigger is better)">
##INFO=<ID=MQB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality Bias (bigger is better)"
##INFO=<ID=BQB,Number=1,Type=Float,Description="Mann-Whitney U test of Base Quality Bias (bigger is better)">
##INFO=<ID=MQSB,Number=1,Type=Float,Description="Mann-Whitney U test of Mapping Quality vs Strand Bias (bigger
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Number of high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="Number of high-quality non-reference bases">
##FORMAT=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##INFO=<ID=DPR,Number=R,Type=Integer,Description="Number of high-quality bases observed for each allele">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=ICB,Number=1,Type=Float,Description="Inbreeding Coefficient Binomial test (bigger is better)">
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same orde
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.1+htslib-1.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3599_
```

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

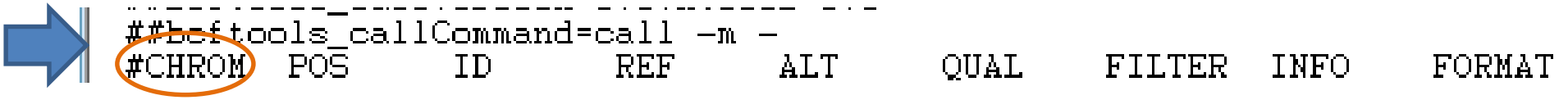


```
##fileformat=VCFv4.1
##bcftools_callCommand=call -m -
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT
```

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

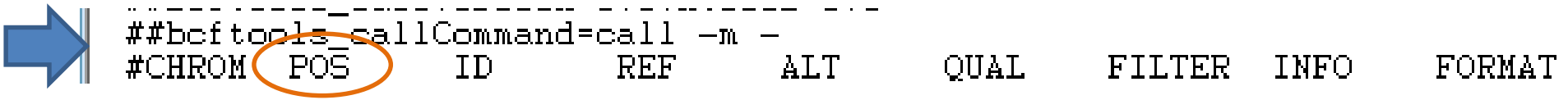


CHROM chromosome

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

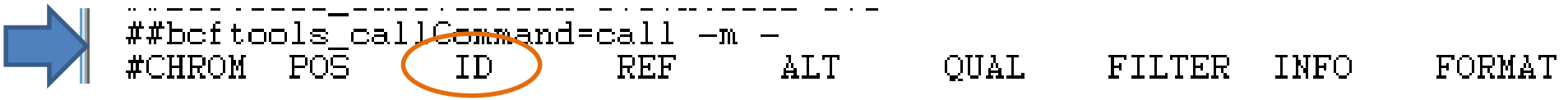


CHROM	chromosome
POS	position of the start of the variant

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

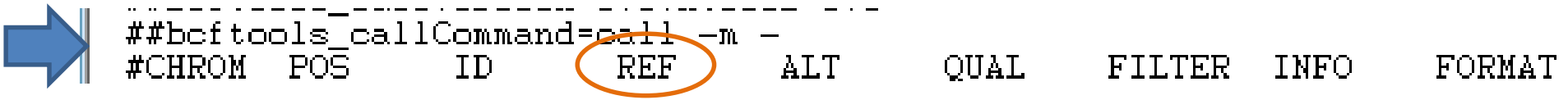


CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:



CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele

variants (.vcf)

Header

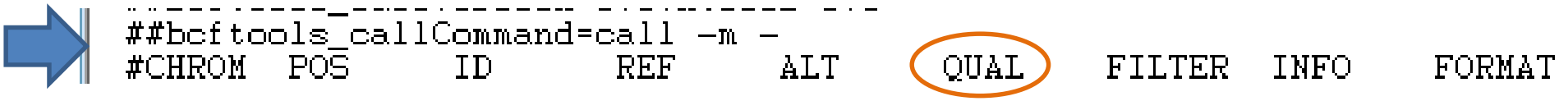
line starting with #: column definition – mandatory columns include:

CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

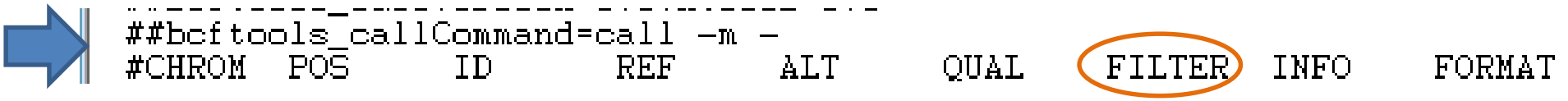


CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

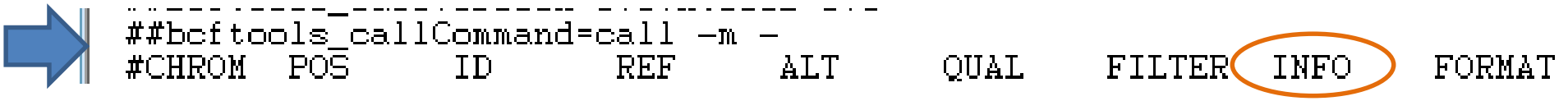


CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score
FILTER	site filtering information

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

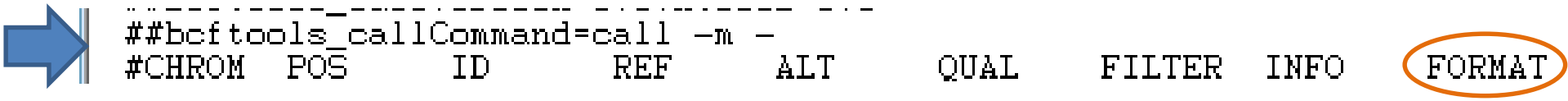


CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score
FILTER	site filtering information
INFO	user extensible annotation (e.g. samtools and GATK may differ in this)

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

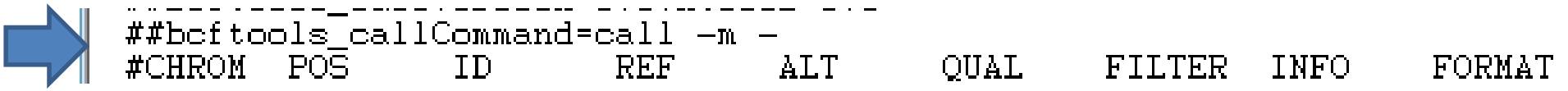


CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score
FILTER	site filtering information
INFO	user extensible annotation (e.g. samtools and GATK may differ in this)
FORMAT	how the information for each sample is presented (i.e. GT:DP:DV:SP:DPR)

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:



CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score
FILTER	site filtering information
INFO	user extensible annotation (e.g. samtools and GATK may differ in this)
FORMAT	how the information for each sample is presented (i.e. GT:DP:DV:SP:DPR)

samples follow

variants (.vcf)

Header

line starting with #: column definition – mandatory columns include:

```
QUAL    FILTER  INFO    FORMAT  GN261_S10       GN1175_S4       GN3599_S8
```

CHROM	chromosome
POS	position of the start of the variant
ID	unique identifier of the variant (e.g. rs number for SNPs)
REF	reference allele
ALT	comma separated list of alternate non-reference alleles
QUAL	phred-scaled quality score
FILTER	site filtering information
INFO	user extensible annotation (e.g. samtools and GATK may differ in this)
FORMAT	how the information for each sample is presented (i.e. GT:DP:DV:SP:DPR)

samples follow

variants (.vcf)

Data

one line per site (all columns described above per line);

useful information per site and per sample;

variants (.vcf)

Data

one line per site (all columns described above per line);

useful information per site and per sample;

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FOR
ref_seq_exons_introns_200Ns_from_patched_ref    487     .       A
ref_seq_exons_introns_200Ns_from_patched_ref    488     .       T
ref_seq_exons_introns_200Ns_from_patched_ref    489     .       G
```

5- Variant calling

#CHROM POS ID REF ALT QUAL FILTER **INFO** FORMAT shark
cc_ref 1610669 . G A 140 . DP=10;VDB=6.833620e 02;AF1=1;AC1=2;
DP4=0,0,6,1;MQ=47;FQ=-48 GT:PL:DP:SP:GQ 1/1:173,21,0:7:0:39

INFO: user extensible annotation (e.g. samtools and GATK may differ in this)

INFO: DP=10;VDB=6.833620e 02;AF1=1;AC1=2;DP4=0,0,6,1;MQ=47;FQ=-48

DP: "Raw read depth"
DPR: "Number of high-quality bases observed for each allele"
DP4: "Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases"
MQ: "Average mapping quality"

#CHROM POS ID REF ALT QUAL FILTER **INFO** FORMAT shark
cc_ref 1610669 . G A 140 . DP=10;VDB=6.833620e 02;AF1=1;AC1=2;
DP4=0,0,6,1;MQ=47;FQ=-48 GT:PL:DP:SP:GQ 1/1:173,21,0:7:0:39

INFO field regards the SITE and NOT the specific samples in multisample calling!!!

#CHROM POS ID REF ALT QUAL FILTER INFO **FORMAT** shark
cc_ref 1610669 . G A 140 . DP=10;VDB=6.833620e 02;AF1=1;AC1=2;
DP4=0,0,6,1;MQ=47;FQ=-48 GT:PL:DP:SP:GQ 1/1:173,21,0:7:0:39

FORMAT: how the information for each sample is presented
(i.e. GT:PL:DP:SP:GQ)

Each sample (multisample calling) has its own FORMAT field,
information are sample specific!!

GT:PL:DP:SP:GQ

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:GQ

GT: genotype

0/0: homozygote reference;
0/1: heterozygote;
1/1: homozygote alternative;

0: reference allele;
1:alternative allele;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:GQ

GT: genotype

0/0: homozygote reference;
0/1: heterozygote;
1/1: homozygote alternative;

1/1:173,21,0:7:0:39

0: reference allele;
1:alternative allele;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:GQ

GT: genotype

0/0: homozygote reference;
0/1: heterozygote;
1/1: homozygote alternative;

1/1:173,21,0:7:0:39

Homozygote alternative

0: reference allele;
1:alternative allele;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:**PL**:DP:SP:GQ

PL: List of Phred-scaled genotype likelihoods;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:**PL**:DP:SP:GQ

PL: List of Phred-scaled genotype likelihoods;

Approximate likelihood = 10^(-**PL**/10)

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:**PL**:DP:SP:GQ

PL: List of Phred-scaled genotype likelihoods;

Approximate likelihood = 10^(-**PL**/10)

1/1:173,21,0:7:0:39

P(0/0) = 10^(-173/10) = 10^(-17.3) = 0.00000000000000005011872
P(0/1) = 10^(-21/10) = 10^(-2.1) = 0.0007943
P(1/1) = 10^(-0/10) = 10^(0) = 1

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:**DP**:SP:GQ

DP: Number of high-quality bases;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:**DP**:SP:GQ

DP: Number of high-quality bases;

1/1:173,21,0:7:0:39

DP=7

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

0.05=10^(- **SP** /10) → -10*Log 0.05 = **SP** → **SP** =13.01

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

0.05=10^(- **SP** /10) → -10*Log 0.05 = **SP** → **SP** =13.01

Would you keep sites with SP ≥ 13 or sites with SP ≤ 13?

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

0.05=10^(- **SP** /10) → -10*Log 0.05 = **SP** → **SP** =13.01

Would you keep sites with SP ≥ 13 or sites with SP ≤ 13?

- ↑ SP → ↓ p value → there is a statistical significant difference in the number of alleles coming from the two strands;
- ↓ SP → ↑ p value → there is NOT a statistical significant difference in the number of alleles coming from the two strands.

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

0.05=10^(- **SP** /10) → -10*Log 0.05 = **SP** → **SP** =13.01

If we assume a p value threshold of 0.05,
we will keep all sites with **SP**≤13

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:**SP**:GQ

SP: Phred-scaled strand bias P-value;

Strand bias p value= 10^(- **SP** /10)

0.05=10^(- **SP** /10) → -10*Log 0.05 = **SP** → **SP** =13.01

If we assume a p value threshold of 0.05,
we will keep all sites with **SP**≤13

1/1:173,21,0:7:0:39

SP=0

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:**GQ**

GQ: conditional genotype quality, encoded as a phred quality;

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:**GQ**

GQ: conditional genotype quality, encoded as a phred quality;

GQ=−10log10 P(genotype call is wrong)

P(genotype call is wrong) = 10^(-**GQ**/10)

FORMAT: how the information for each sample is presented (i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:**GQ**

GQ: conditional genotype quality, encoded as a phred quality;

GQ=−10log10 P(genotype call is wrong)

P(genotype call is wrong) = 10^(-**GQ**/10)

1/1:173,21,0:7:0:39

P(genotype call is wrong) = 10^(-**39**/10)=10^-3.9=0.00012589

#CHROM POS ID REF ALT QUAL FILTER INFO **FORMAT** shark
cc_ref 1610669 . G A 140 . DP=10;VDB=6.833620e 02;AF1=1;AC1=2;
DP4=0,0,6,1;MQ=47;FQ=-48 GT:PL:DP:SP:GQ 1/1:173,21,0:7:0:39

FORMAT: how the information for each sample is presented
(i.e. GT:PL:DP:SP:GQ)

GT:PL:DP:SP:GQ

GT: Genotype;
PL: List of Phred-scaled genotype likelihoods;
DP: Number of high-quality bases;
SP: Phred-scaled strand bias P-value;
DPR: Genotype quality.

Useful references and resources:

https://samtools.github.io/hts-specs/VCFv4.2.pdf

https://gist.github.com/inutano/f0a2f5c219ab4920c5b5

https://faculty.washington.edu/browning/beagle/intro-to-vcf.html

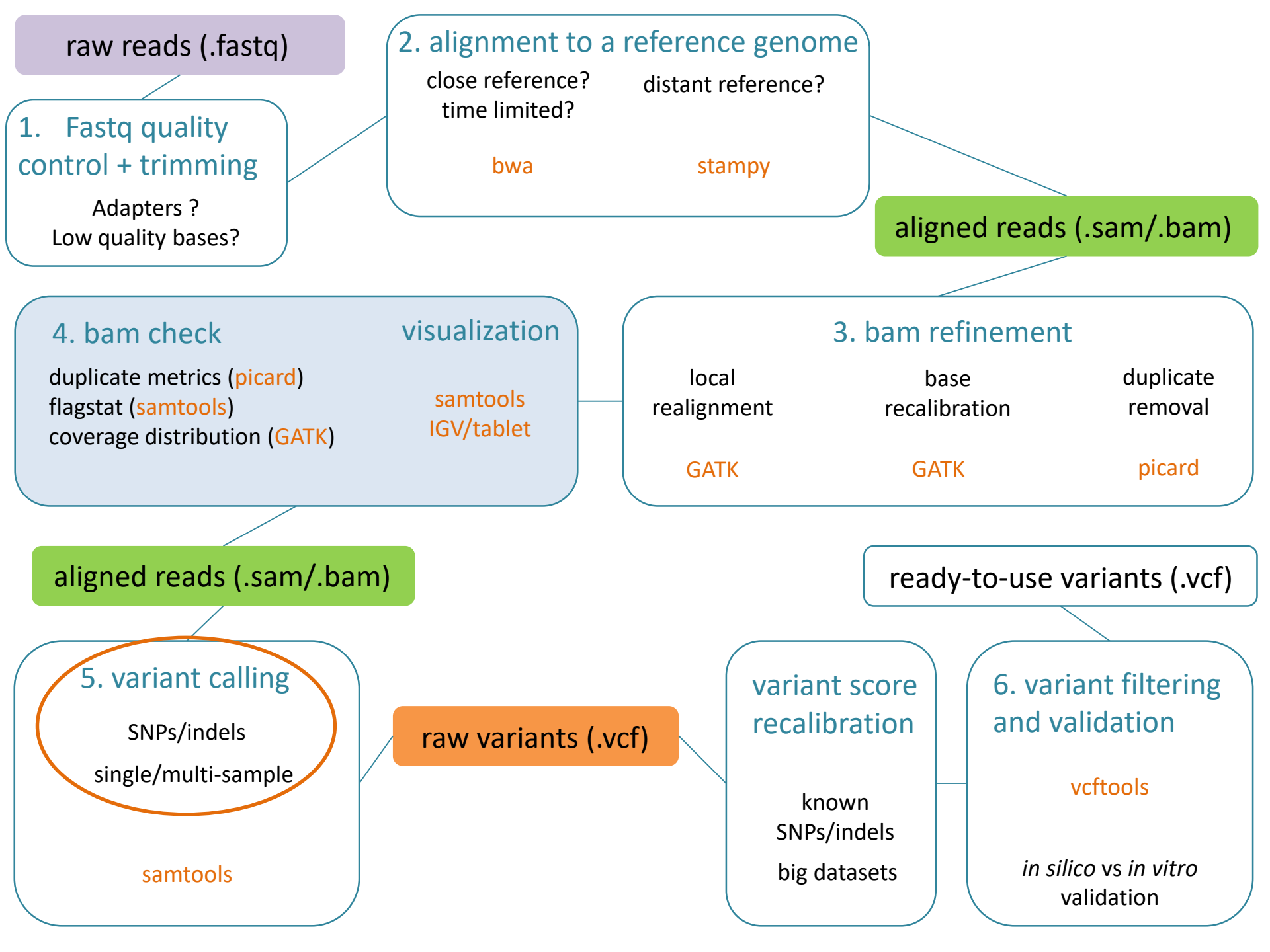
http://gatkforums.broadinstitute.org/gatk/discussion/1268/what-is-a-vcf-and-how-should-i-interpret-it

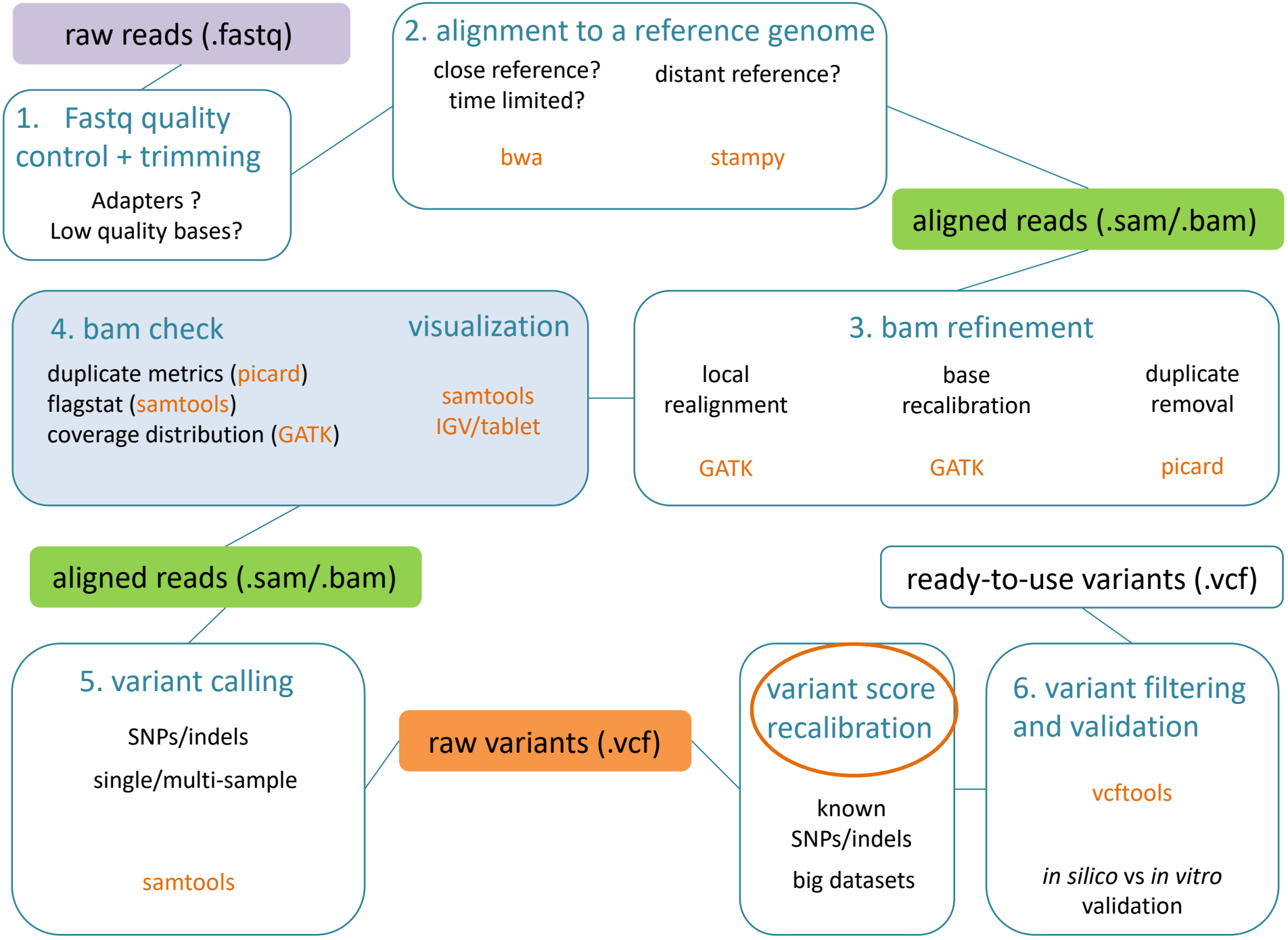
5- Variant calling

We have our raw variants but now we need to refine our dataset…how?

- Variant score recalibration;
- Variant filtering;

- Validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

5- Variant calling: Variant quality score recalibration

- Available in GATK.
- It aims at producing well-calibrated probabilities for the variants called.

5- Variant calling: Variant quality score recalibration

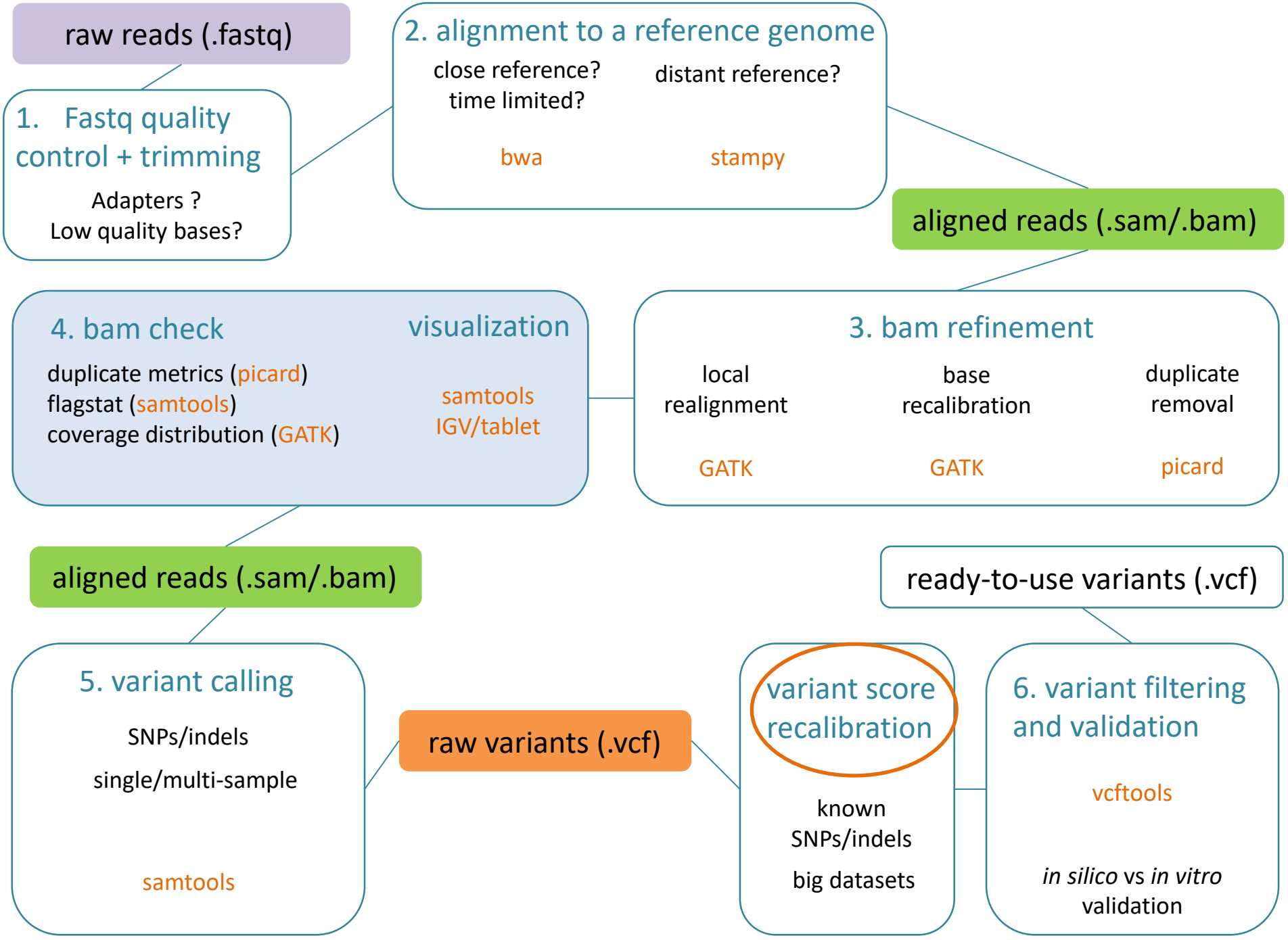
- Available in GATK.
- It aims at producing well-calibrated probabilities for the variants called.
- It develops a continuous, covarying estimate of the relationship between SNP call annotations (e.g. MQ, QD…) and the probability that a SNP is a true genetic variant versus a sequencing or data processing artifact.

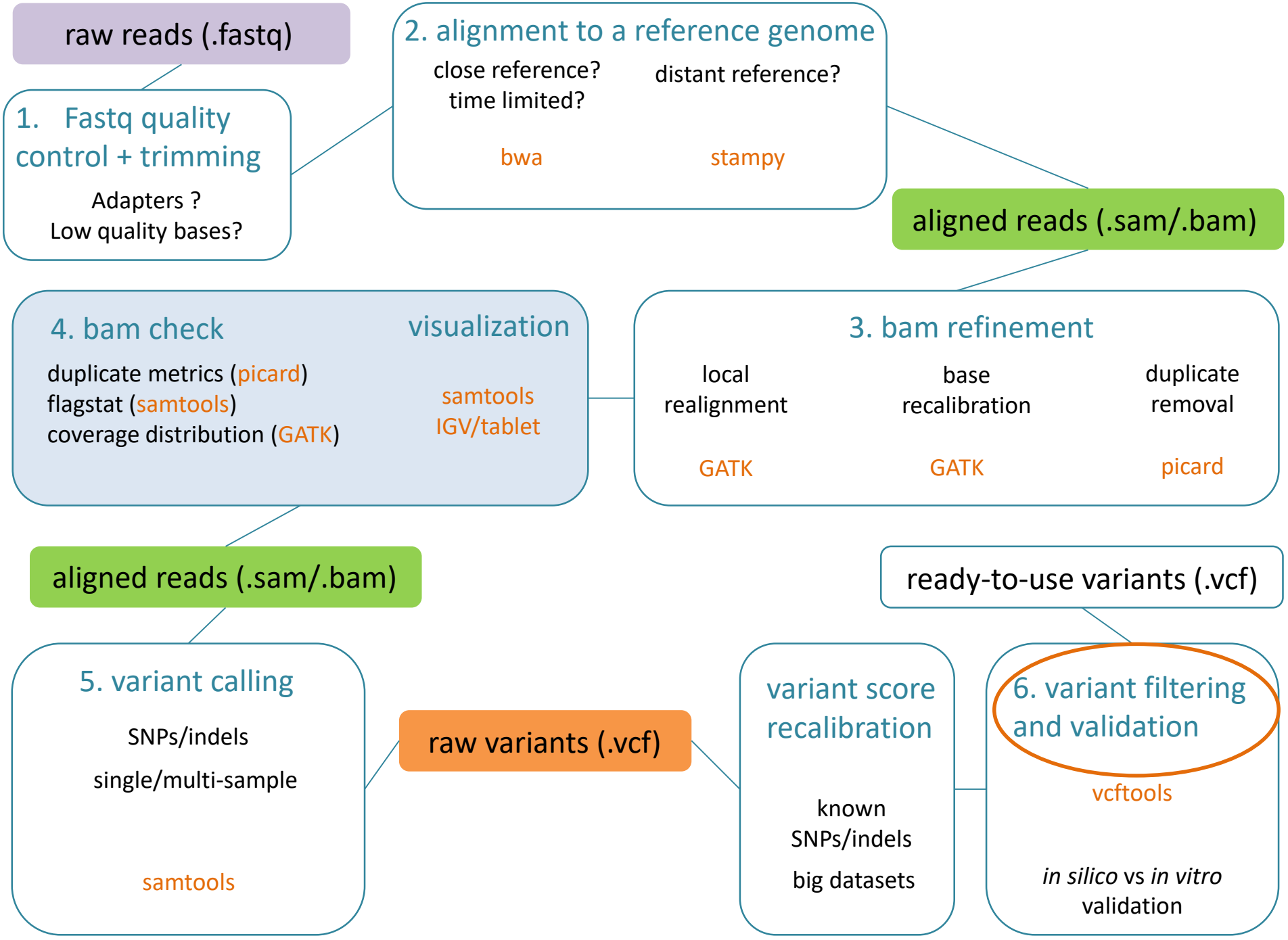
5- Variant calling: Variant quality score recalibration

- Available in GATK.
- It aims at producing well-calibrated probabilities for the variants called.
- It develops a continuous, covarying estimate of the relationship between SNP call annotations (e.g. MQ, QD…) and the probability that a SNP is a true genetic variant versus a sequencing or data processing artifact.
- It needs “true sites” to be trained (i.e. HapMap Phase 3 data, OMNI 2.5 M, etc…).

5- Variant calling: Variant quality score recalibration

- Available in GATK.
- It aims at producing well-calibrated probabilities for the variants called.
- It develops a continuous, covarying estimate of the relationship between SNP call annotations (e.g. MQ, QD…) and the probability that a SNP is a true genetic variant versus a sequencing or data processing artifact.
- It needs “true sites” to be trained (i.e. HapMap Phase 3 data, OMNI 2.5 M, etc…).
- We are not going to use it, because it needs big datasets (either many samples, or whole genome data) to work properly.
- You can find more information at http://gatkforums.broadinstitute.org/discussion/39/variant-quality-score-recalibration-vqsr


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;
- Presence of sequencing error;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;
- Presence of sequencing error;
- To create a high quality dataset;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;
- Presence of sequencing error;
- To create a high quality dataset;

WHY validation???

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;
- Presence of sequencing error;
- To create a high quality dataset;

WHY validation???

- An error-free dataset is unrealistic;

WHY filtering???

- Each caller tends to call as many sites it can but it provides useful information on several parameters;
- Many calls are “guesses”;
- Artifacts;
- Presence of sequencing error;
- To create a high quality dataset;

WHY validation???

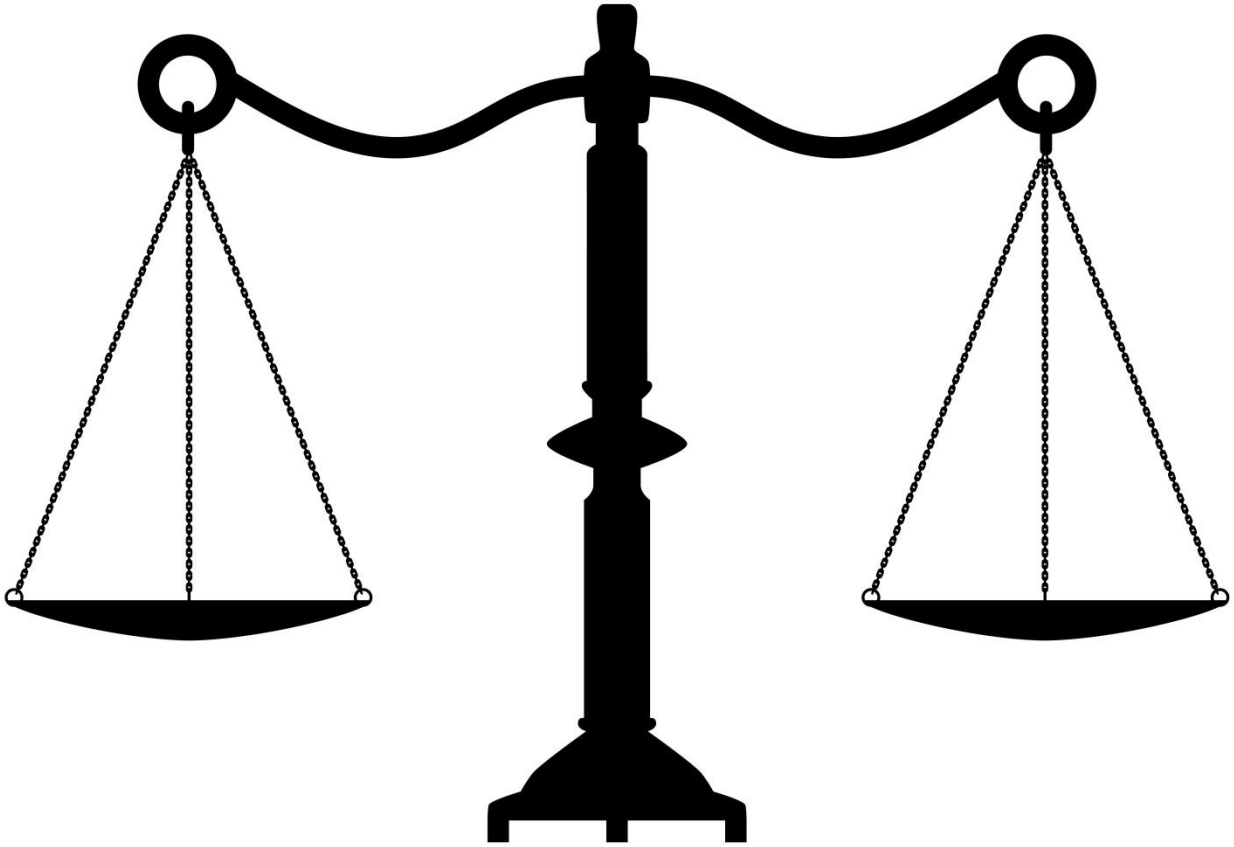
- An error-free dataset is unrealistic;
- To estimate the % of error within our high quality dataset;

5- Variant filtering and validation: **FILTERING**

Filtering is all about finding the right balance:

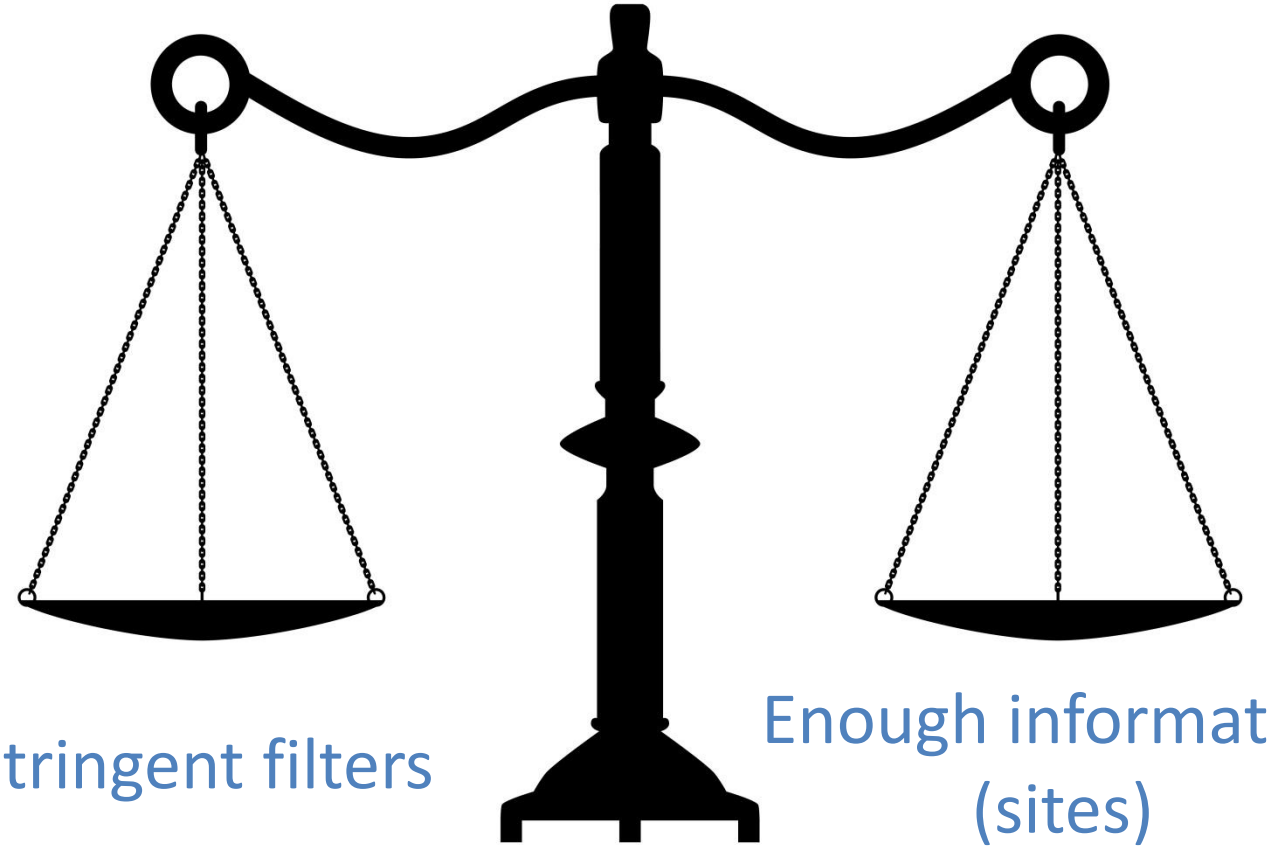
5- Variant filtering and validation: **FILTERING**

Filtering is all about finding the right balance:



5- Variant filtering and validation: **FILTERING**

Filtering is all about finding the right balance:



Stringent filters

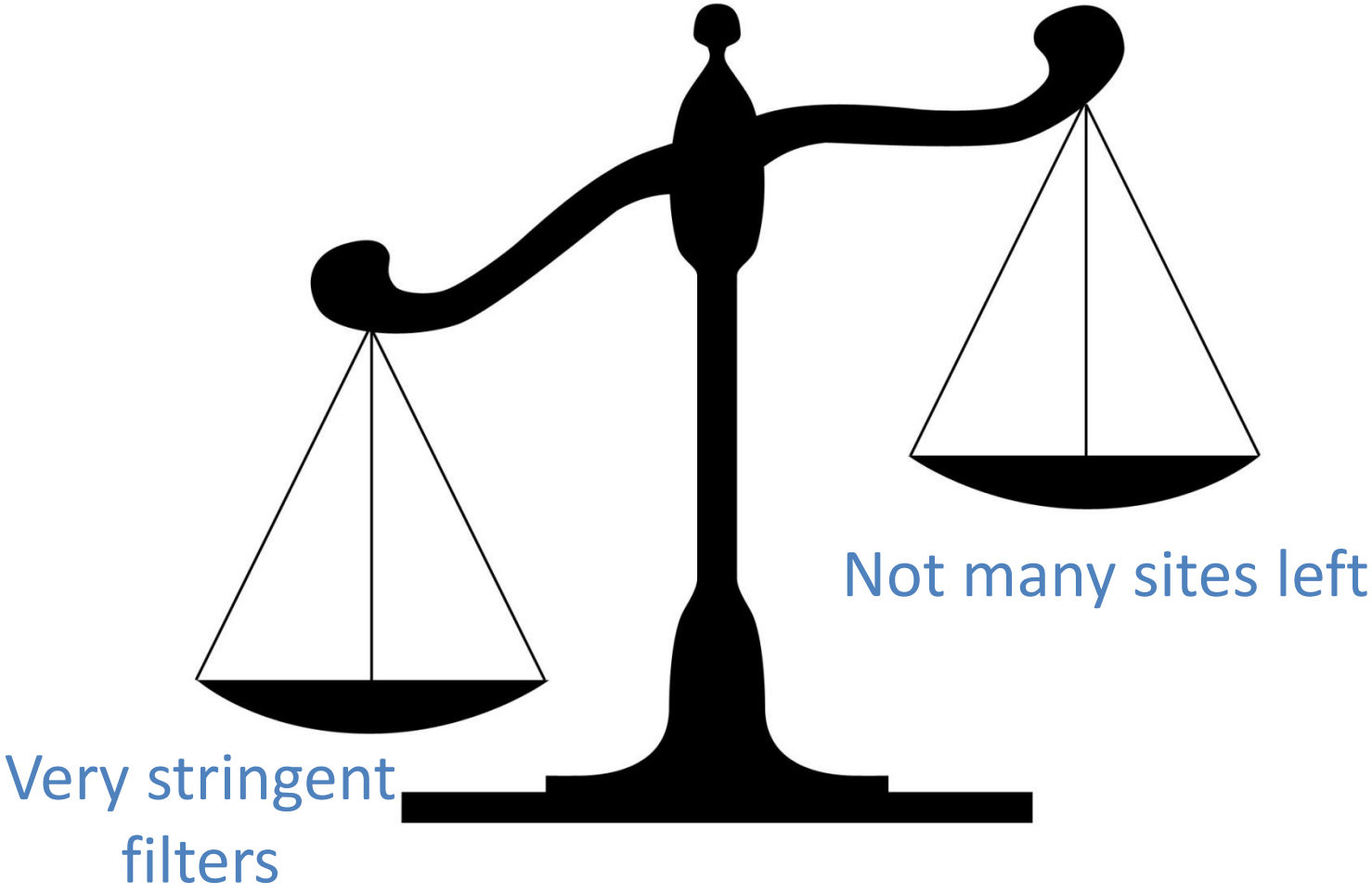
Enough information (sites)

5- Variant filtering and validation: **FILTERING**

Less stringent filters

Lots of sites but with many possible errors

5- Variant filtering and validation: **FILTERING**



5- Variant filtering and validation: **FILTERING**

General approach:

5- Variant filtering and validation: **FILTERING**

General approach:

- Try different thresholds for each parameter and draw a distribution;

5- Variant filtering and validation: **FILTERING**

General approach:

- Try different thresholds for each parameter and draw a distribution;
- If possible, look for sudden changes in the distribution and set a threshold;

5- Variant filtering and validation: **FILTERING**

General approach:

- Try different thresholds for each parameter and draw a distribution;
- If possible, look for sudden changes in the distribution and set a threshold;
- No fixed rule on which thresholds you should use;

5- Variant filtering and validation: **FILTERING**

General approach:

- Try different thresholds for each parameter and draw a distribution;
- If possible, look for sudden changes in the distribution and set a threshold;
- No fixed rule on which thresholds you should use;
- It is data and project specific;

5- Variant filtering and validation: **FILTERING**

General approach:

- Try different thresholds for each parameter and draw a distribution;
- If possible, look for sudden changes in the distribution and set a threshold;
- No fixed rule on which thresholds you should use;
- It is data and project specific;
- It is a crucial step to create a high quality dataset.

5- Variant filtering and validation: **FILTERING**

Common cautions:

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality BQ20

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality BQ20
- Depth (min and max) very dependent on your average

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)
- Strand-bias	p-value>0.05

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)
- Strand-bias	p-value>0.05
- SNP density	dependent on the genome [e.g. no more than 1 SNP/4bp]

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)
- Strand-bias	p-value>0.05
- SNP density	dependent on the genome [e.g. no more than 1 SNP/4bp]
- Indel proximity	not closer than 10bp to an indel

5- Variant filtering and validation: **FILTERING**

Common cautions:

- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)
- Strand-bias	p-value>0.05
- SNP density	dependent on the genome [e.g. no more than 1 SNP/4bp]
- Indel proximity	not closer than 10bp to an indel

- Missing data?

5- Variant filtering and validation: **FILTERING**

Common cautions:

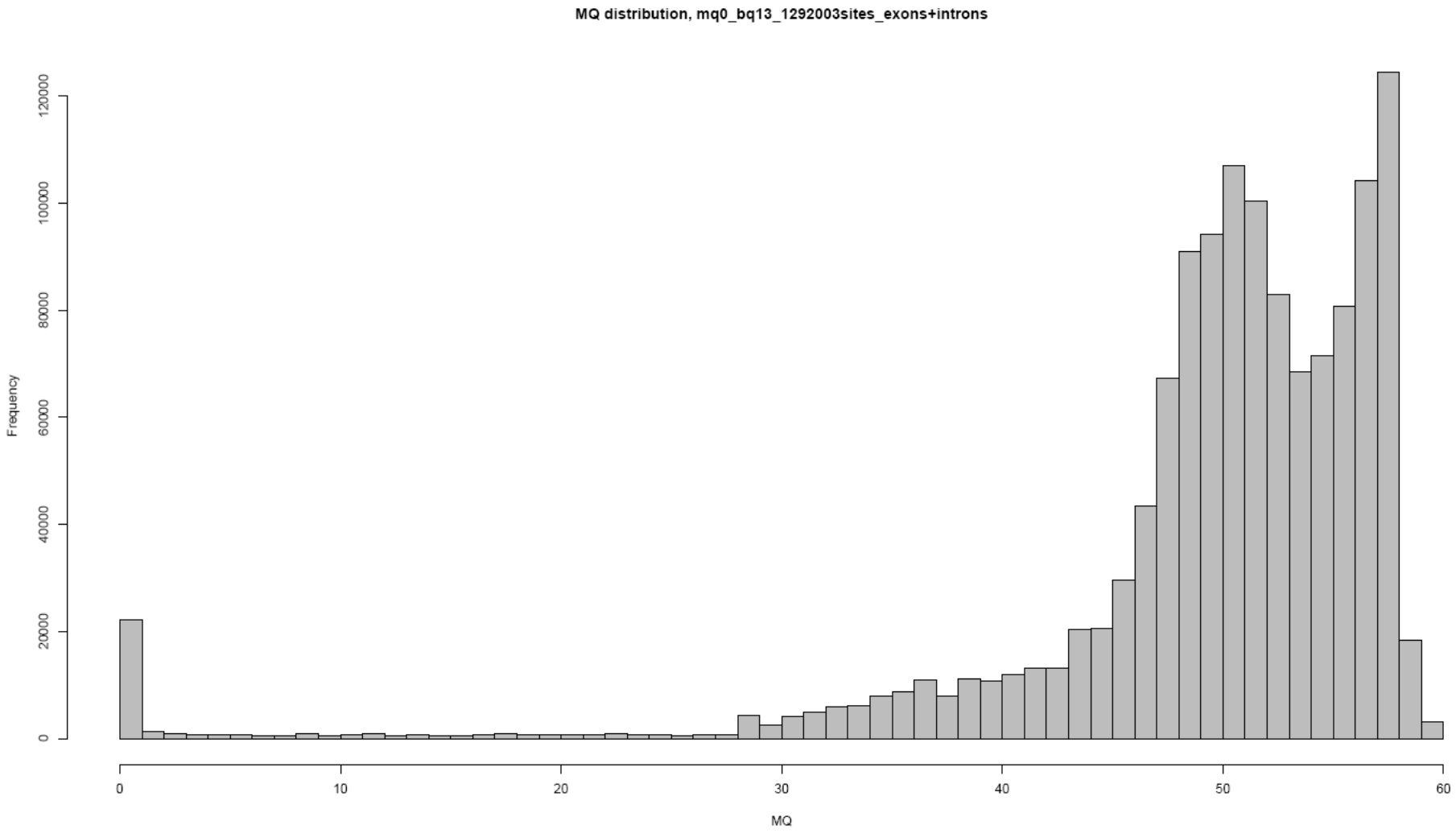
- Base quality	BQ20
- Depth (min and max)	very dependent on your average
- Mapping quality	MQ40/50 (minimum MQ30)
- Strand-bias	p-value>0.05
- SNP density	dependent on the genome [e.g. no more than 1 SNP/4bp]
- Indel proximity	not closer than 10bp to an indel

- Missing data?

- Some filters may be applied during the variant calling while others are applied afterwards.

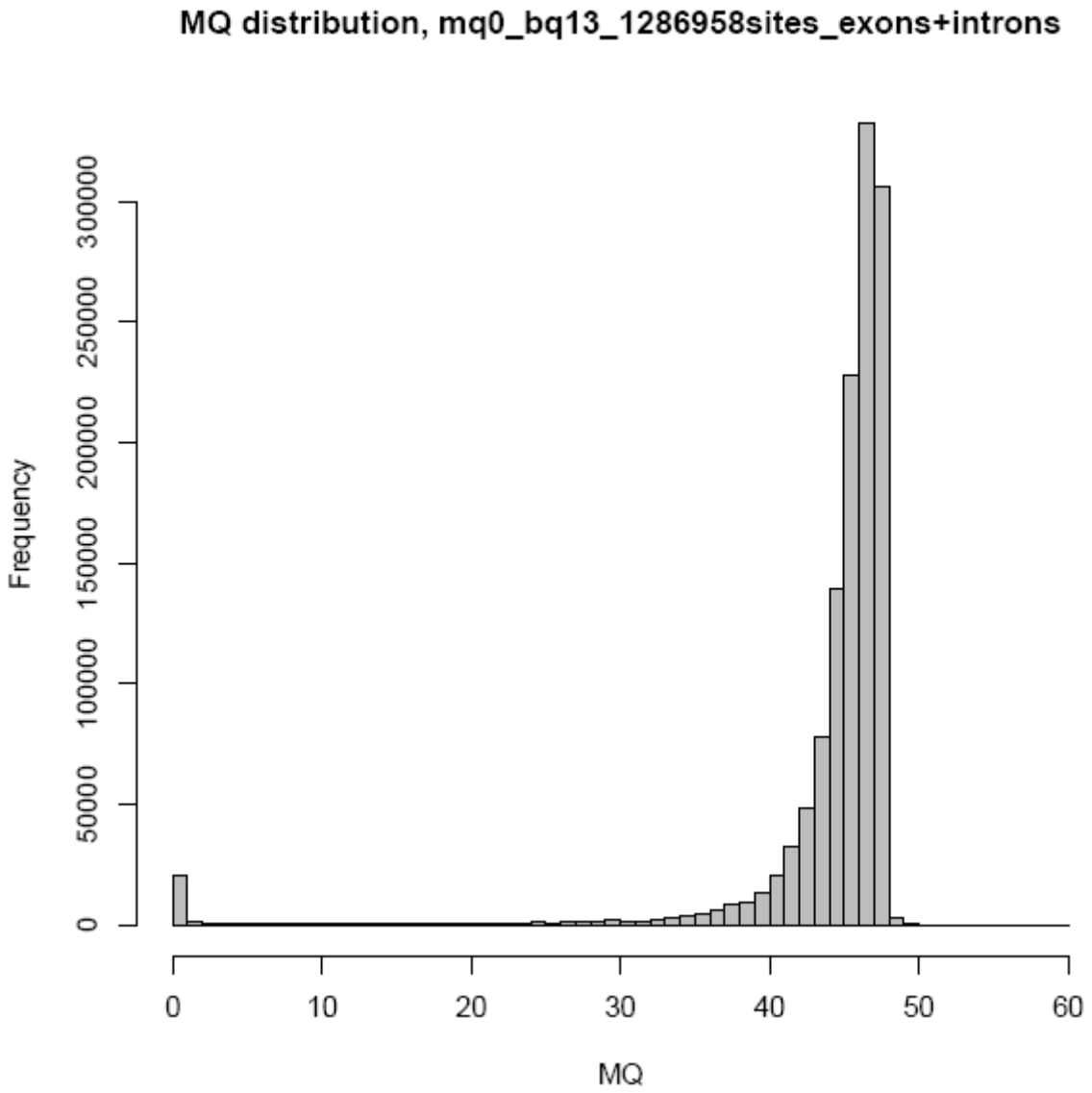
5- Variant filtering and validation: **FILTERING**

MQ distribution



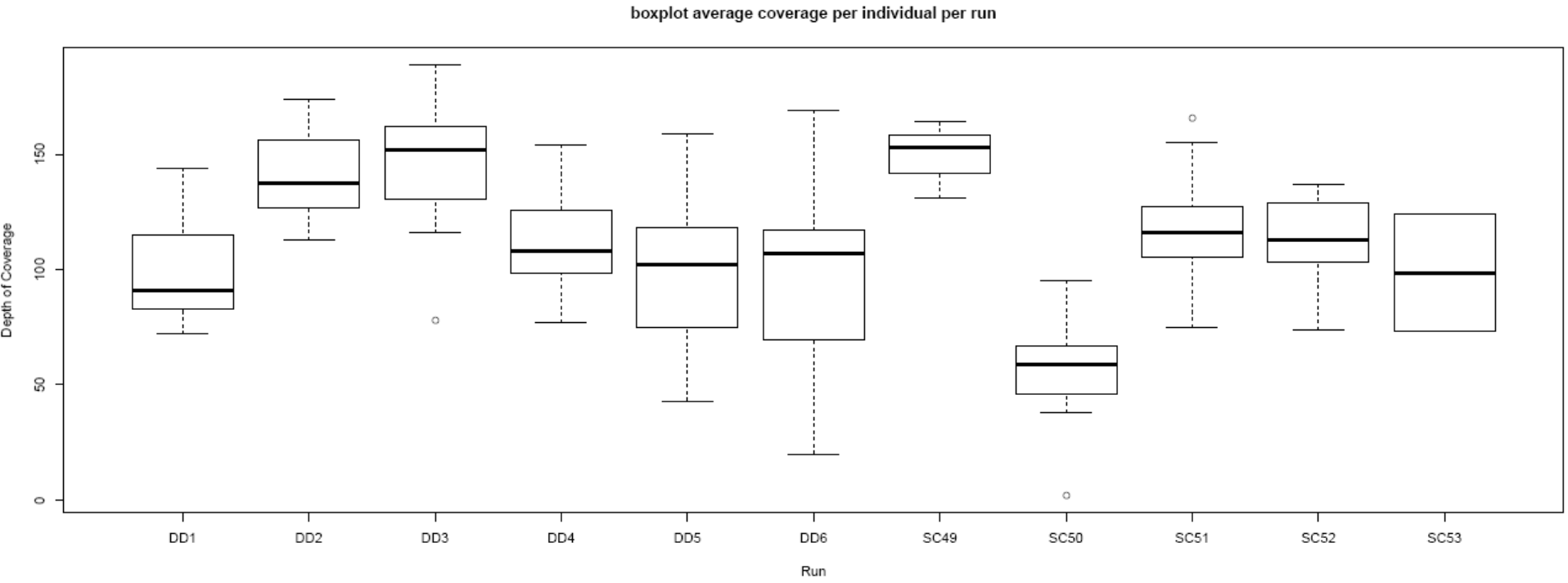
5- Variant filtering and validation: **FILTERING**

MQ distribution



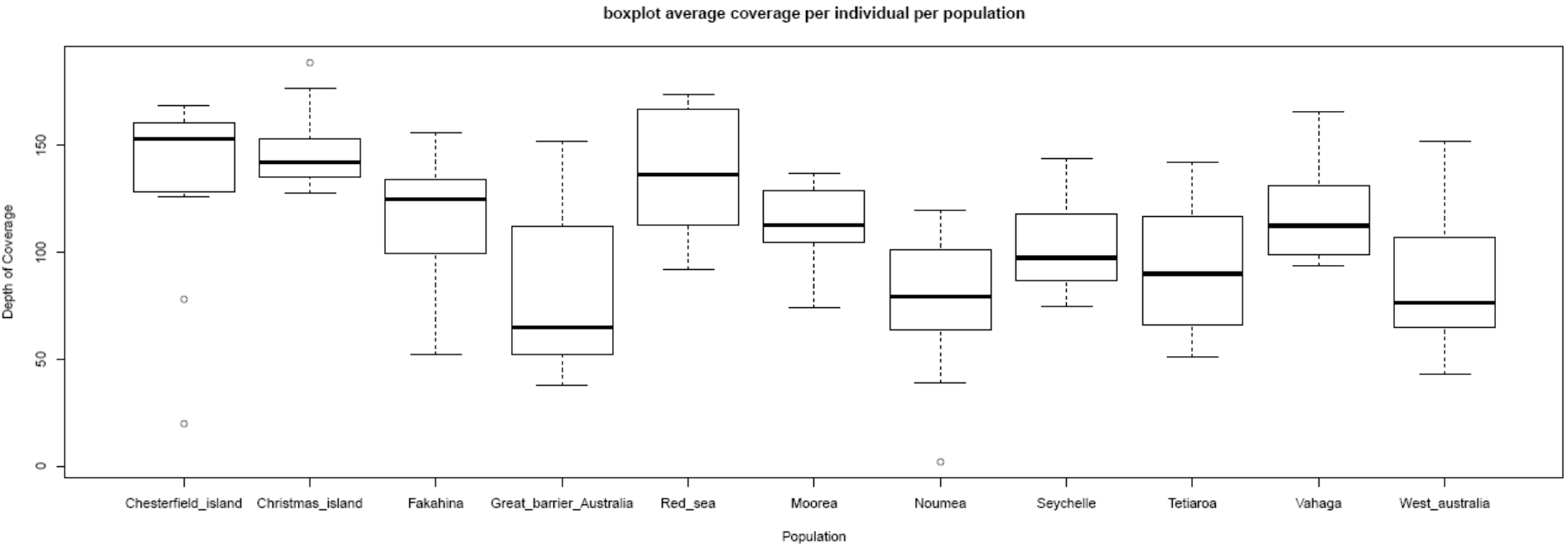
5- Variant filtering and validation: **FILTERING**

Coverage: distribution per **run**



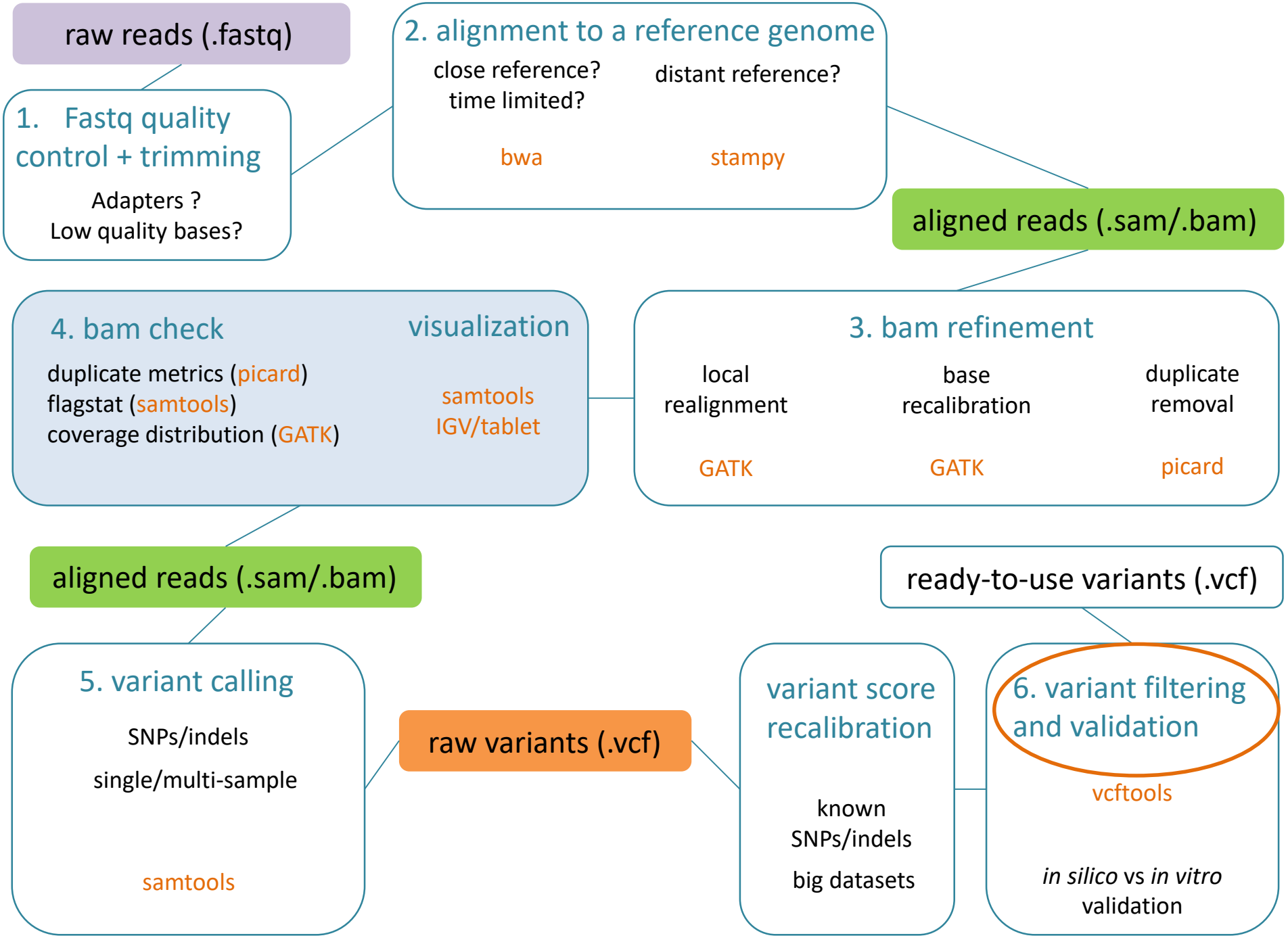
5- Variant filtering and validation: **FILTERING**

Coverage: distribution per **population**



5- Variant filtering and validation: **FILTERING**

VCFtools (http://vcftools.sourceforge.net/)


raw reads (.fastq)
1. Fastq quality control + trimming
Adapters ?
Low quality bases?
2. alignment to a reference genome
close reference?
time limited?
bwa
distant reference?
stampy
aligned reads (.sam/.bam)
3. bam refinement
local realignment
GATK
base recalibration
GATK
duplicate removal
picard
4. bam check
duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)
visualization
samtools
IGV/tablet
aligned reads (.sam/.bam)
5. variant calling
SNPs/indels
single/multi-sample
samtools
raw variants (.vcf)
variant score recalibration
known SNPs/indels
big datasets
ready-to-use variants (.vcf)
6. variant filtering and validation
vcftools
in silico vs in vitro validation

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

- How to do it:

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

- How to do it:

    - Sanger Sequencing vs NGS (our) dataset;

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

- How to do it:

  - Sanger Sequencing vs NGS (our) dataset;
  - Public dataset vs NGS (our) dataset;

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

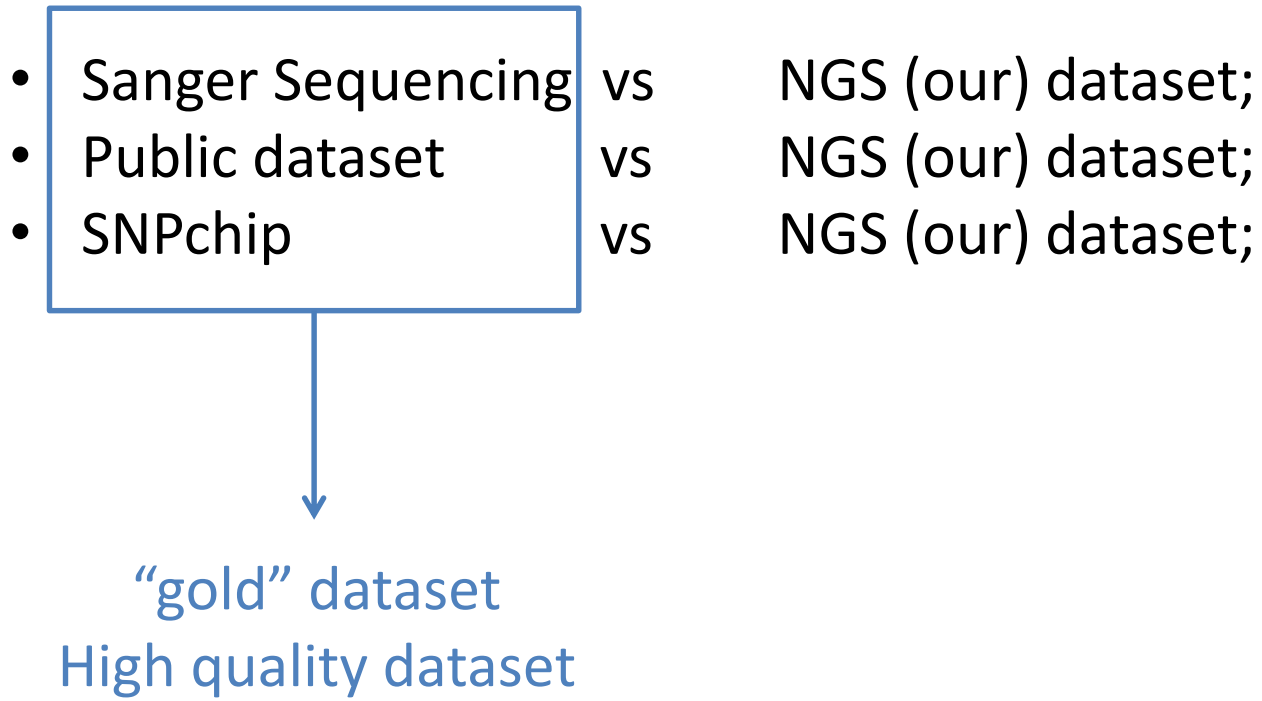
- How to do it:

  - Sanger Sequencing vs NGS (our) dataset;
  - Public dataset vs NGS (our) dataset;
  - SNPchip vs NGS (our) dataset;

5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

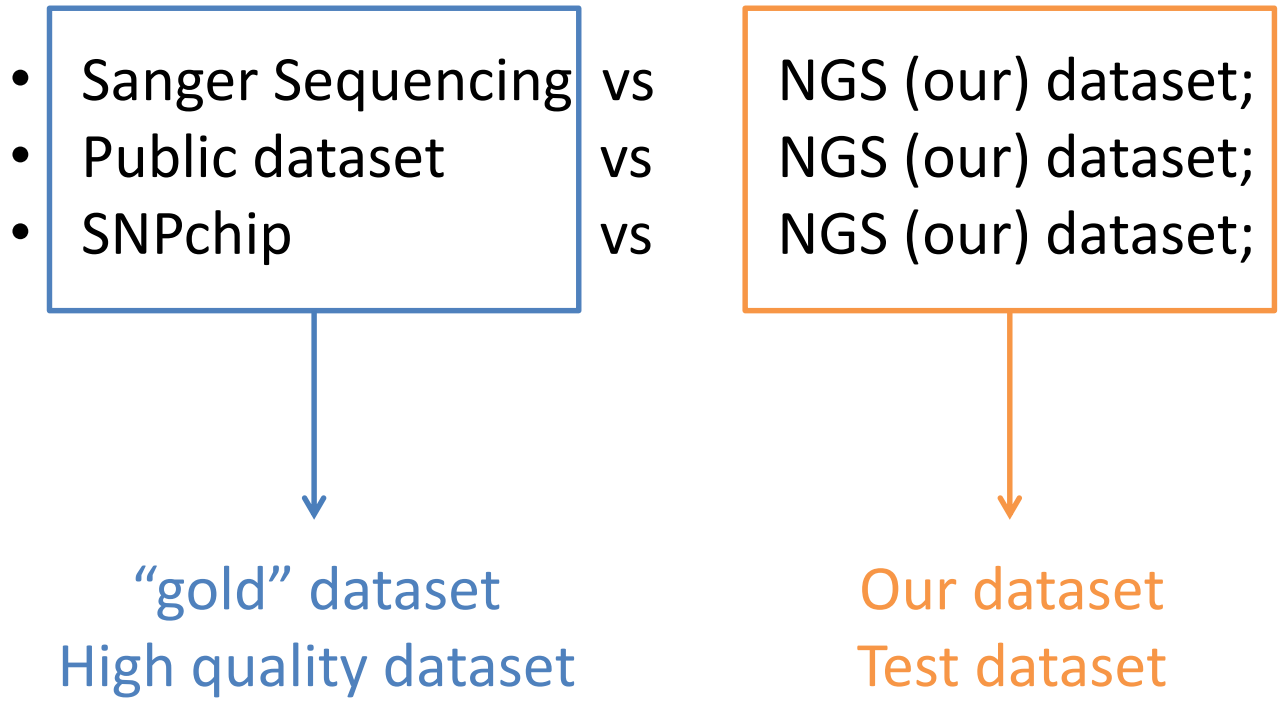
- How to do it:



5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

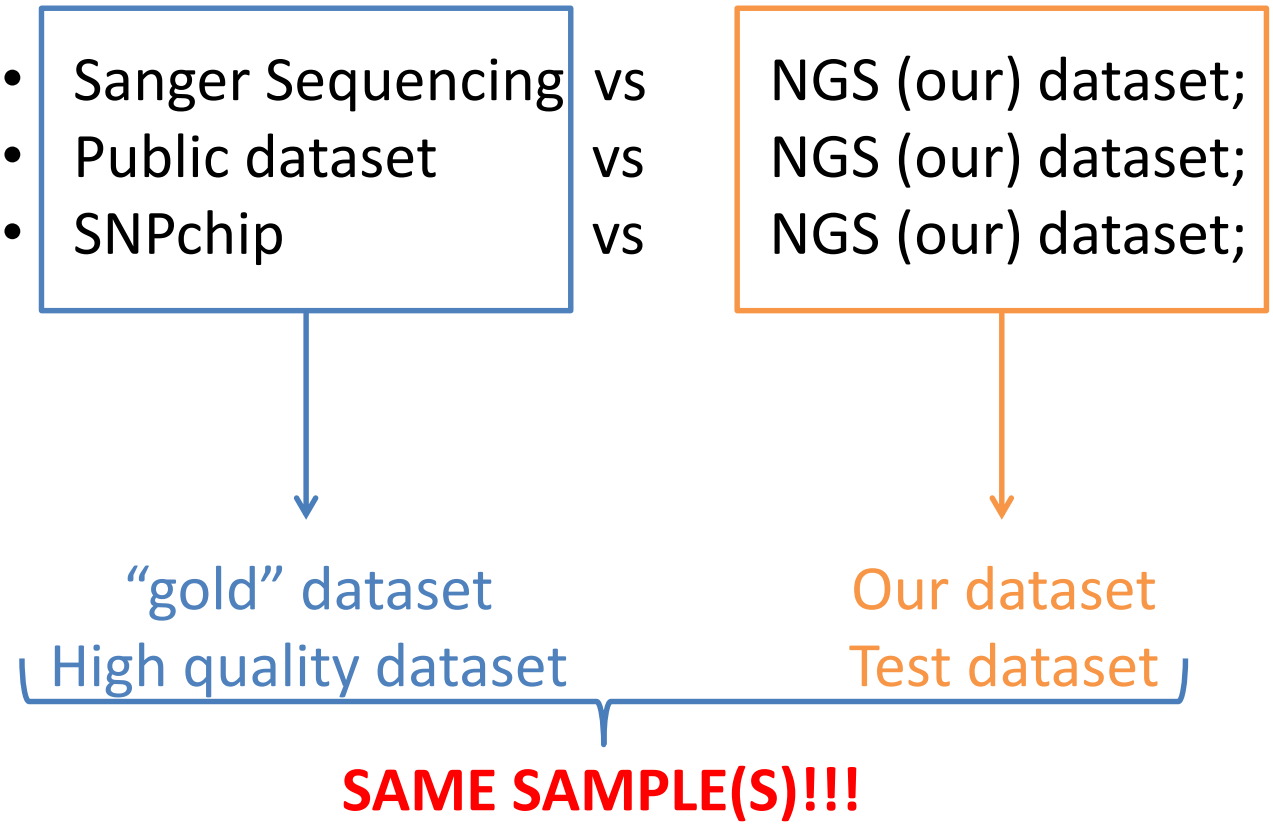
- How to do it:



5- Variant filtering and validation: **VALIDATION**

- Why?
  - An error-free dataset is unrealistic;
  - To estimate the % of error within our high quality dataset;

- How to do it:



5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

- All sites, site by site;

5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

- All sites, site by site;
- TRUE: concordant site;

5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

- All sites, site by site;
- TRUE: concordant site;
- FALSE: discordant site;

5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

- All sites, site by site;
- TRUE: concordant site;
- FALSE: discordant site;
- POSITIVE: polymorphic site (compared to reference sequence);

5- Variant filtering and validation: **VALIDATION**

Compare our sample to a “gold” dataset (same sample!!!):

- All sites, site by site;
- TRUE: concordant site;
- FALSE: discordant site;
- POSITIVE: polymorphic site (compared to reference sequence);
- NEGATIVE: reference site (compared to the reference sequence carrying a variant at that position);

5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!

our sequenced sample

5- Variant filtering and validation: VALIDATION

SAME SAMPLE(S)!!!

our sequenced sample

“Gold” dataset – same sample

5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!

our sequenced sample

“Gold” dataset – same sample

5- Variant filtering and validation: **VALIDATION**

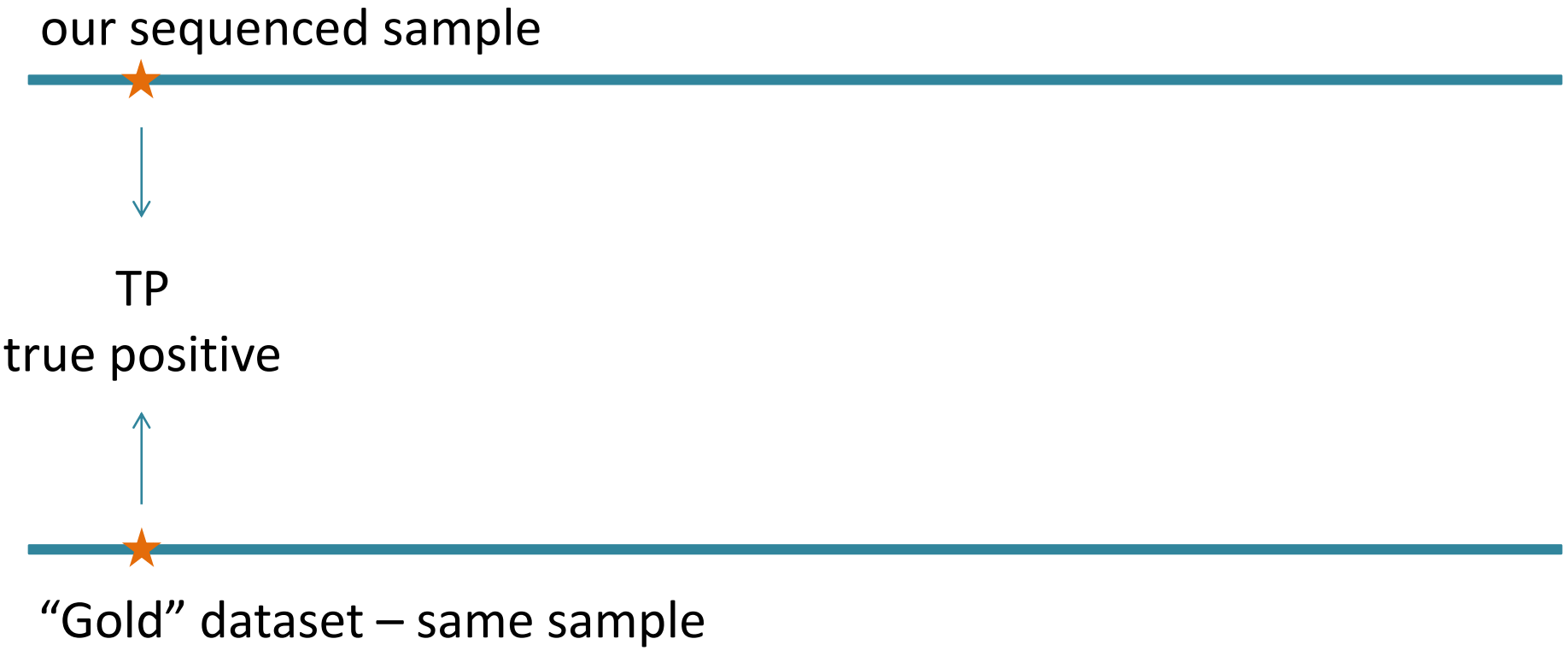
SAME SAMPLE(S)!!!

our sequenced sample

“Gold” dataset – same sample

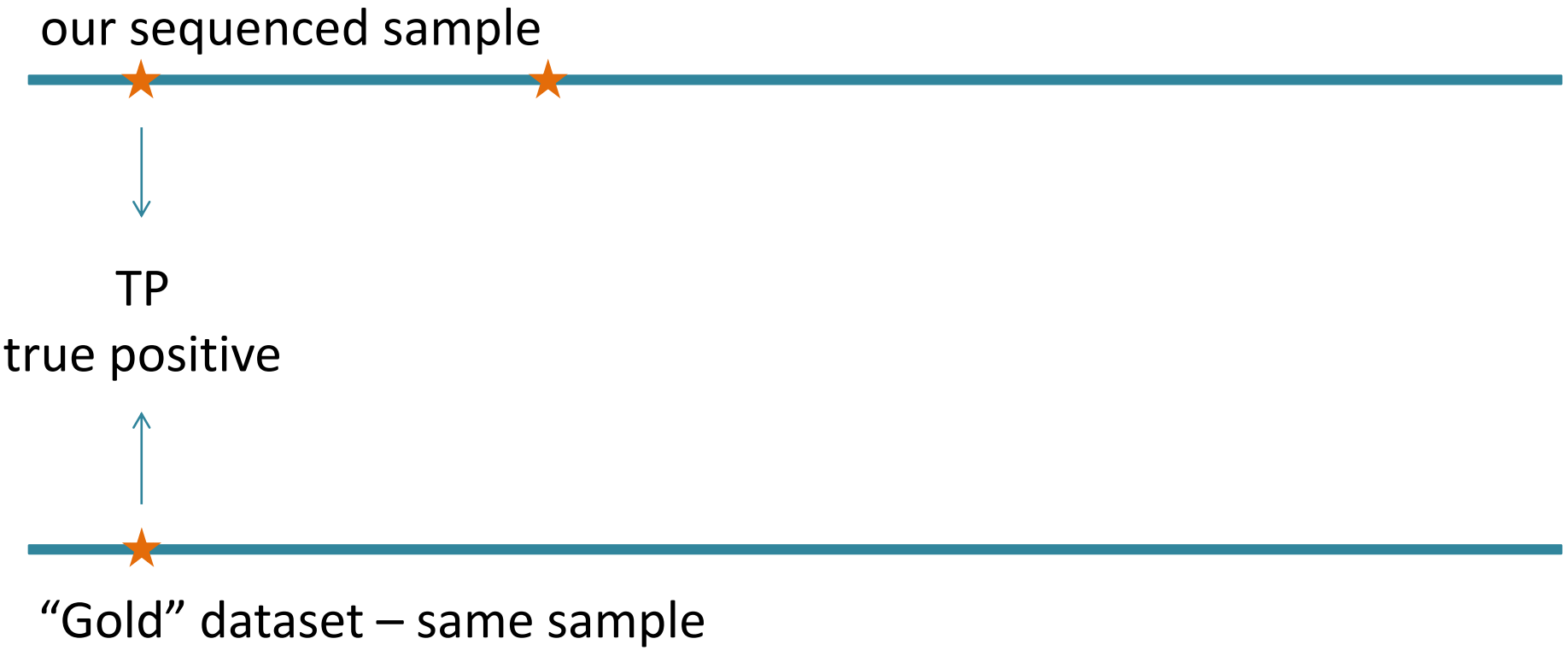
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



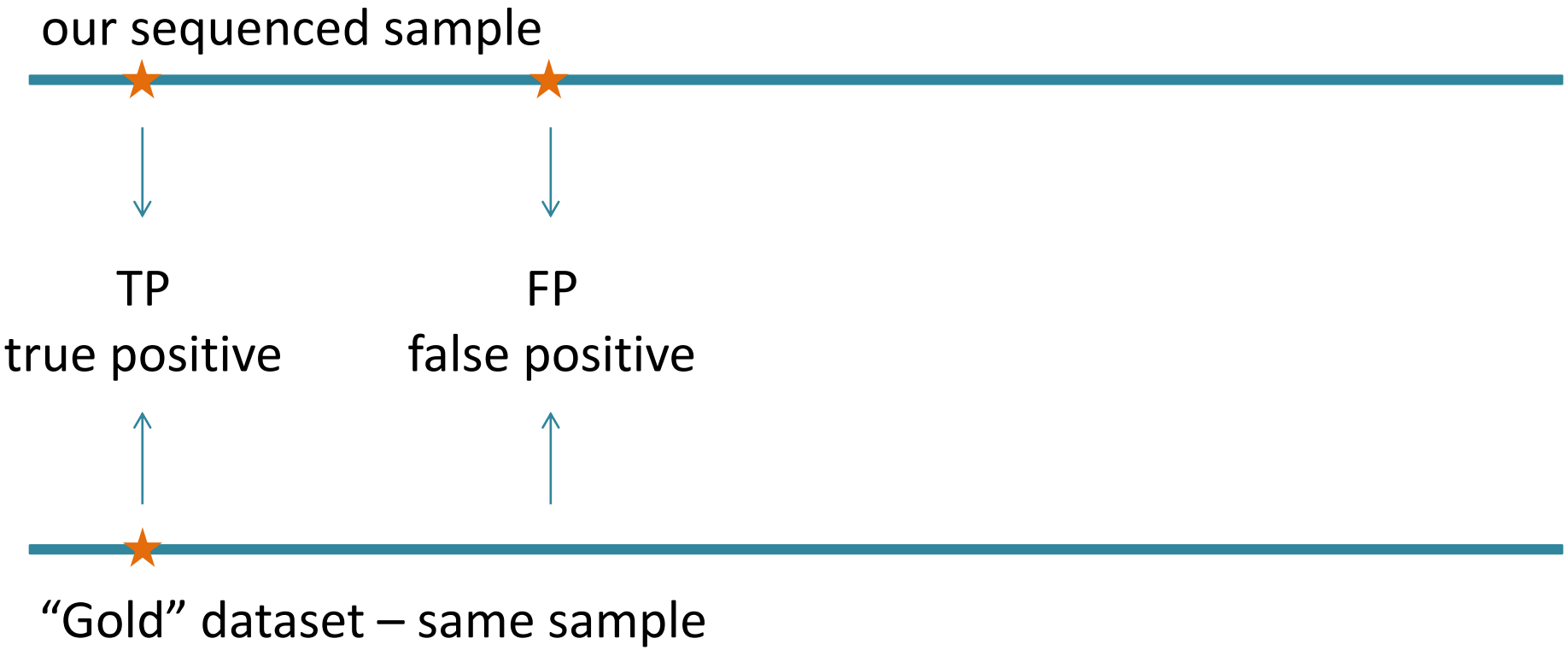
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



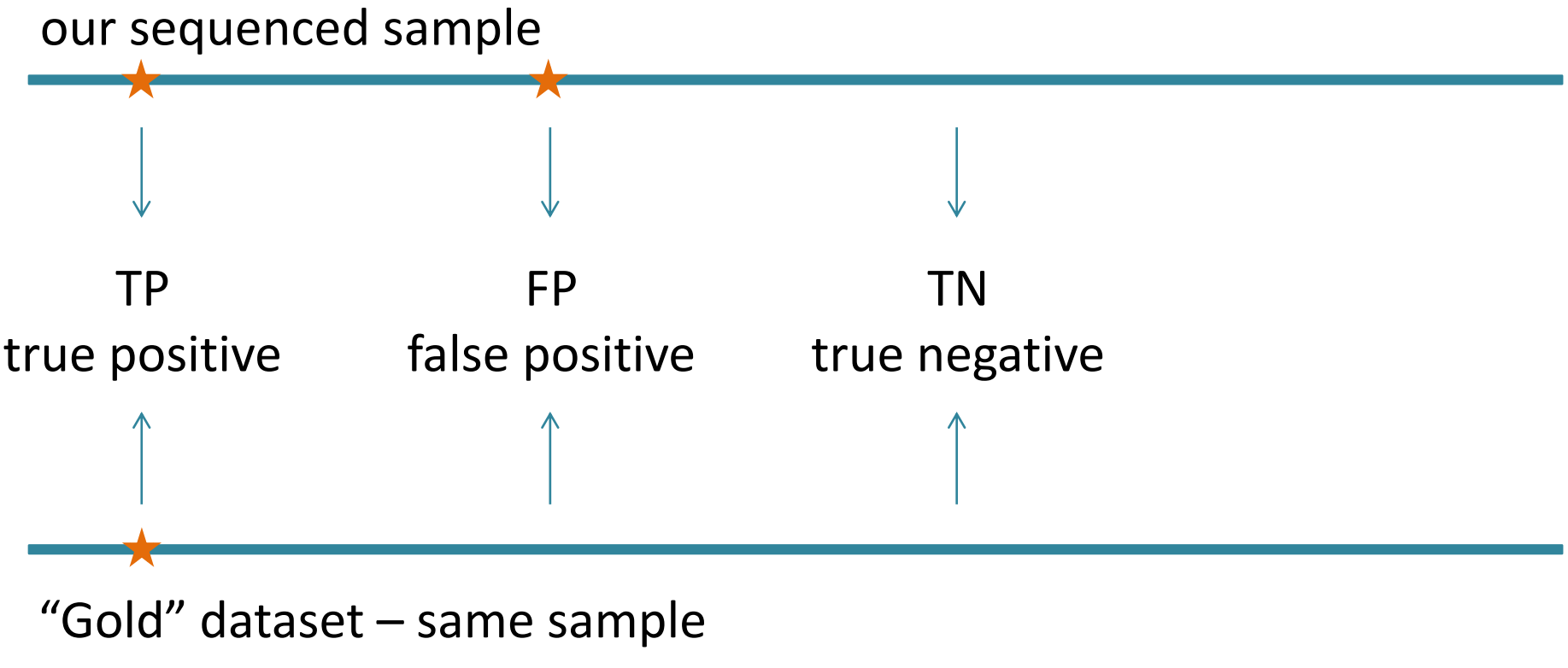
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



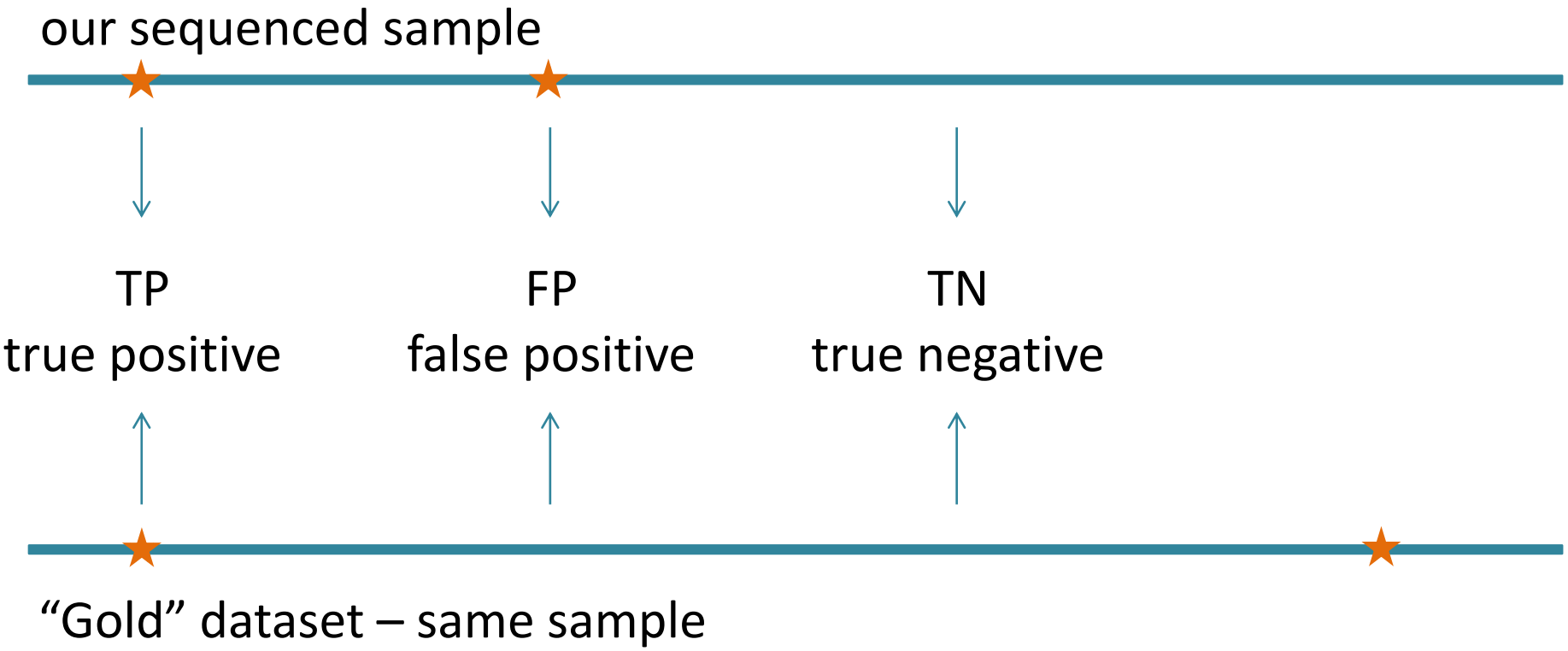
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



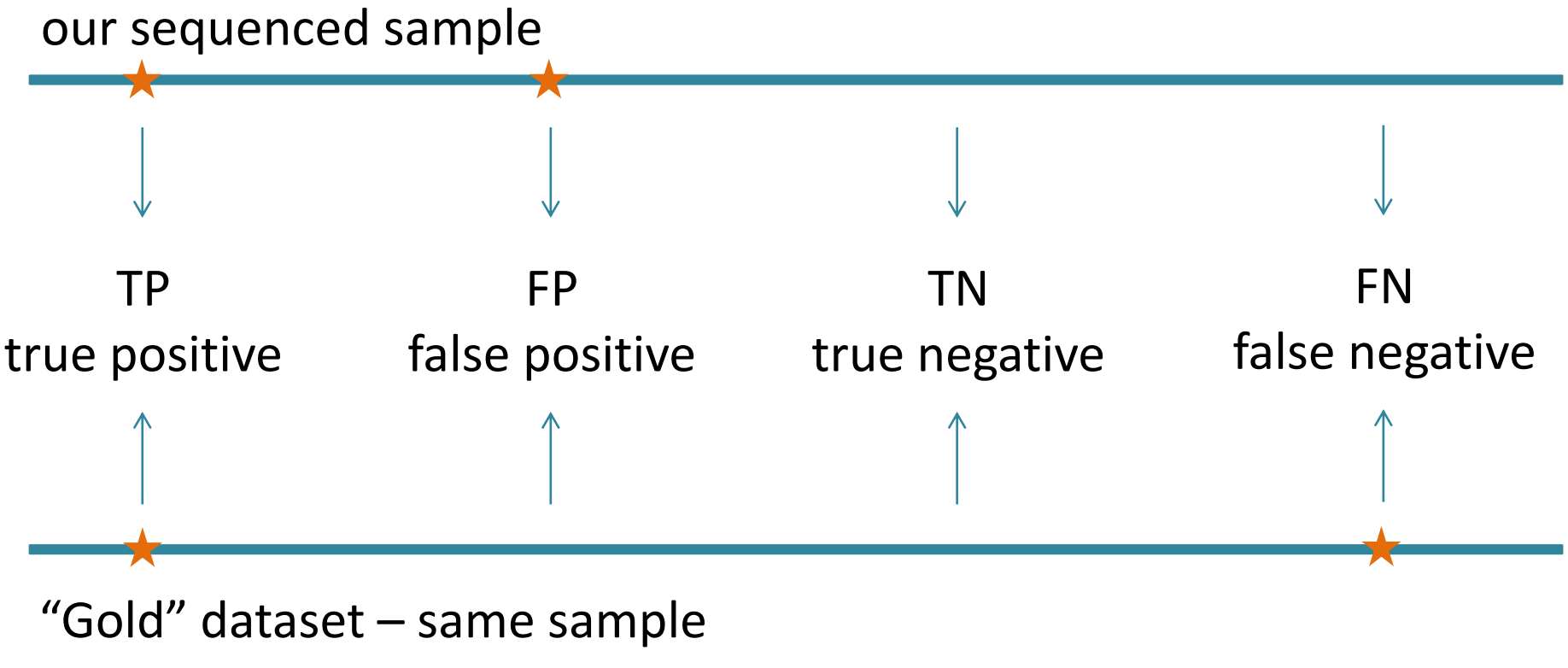
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



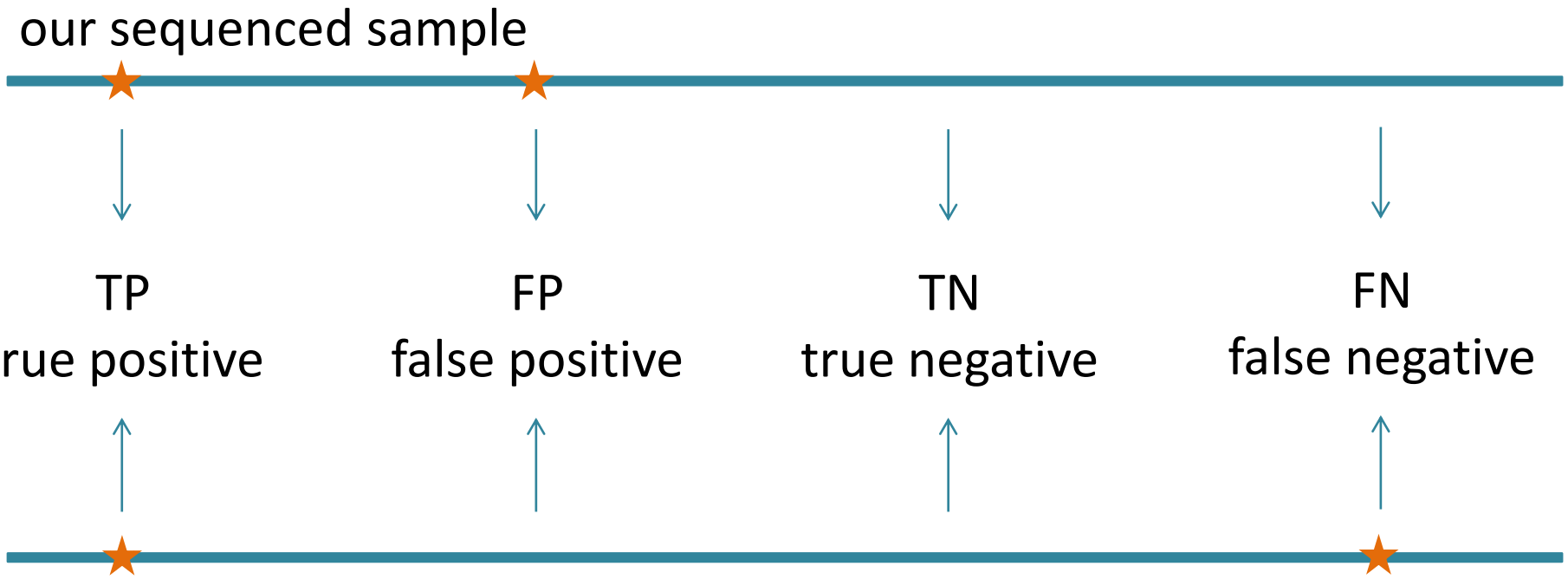
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!

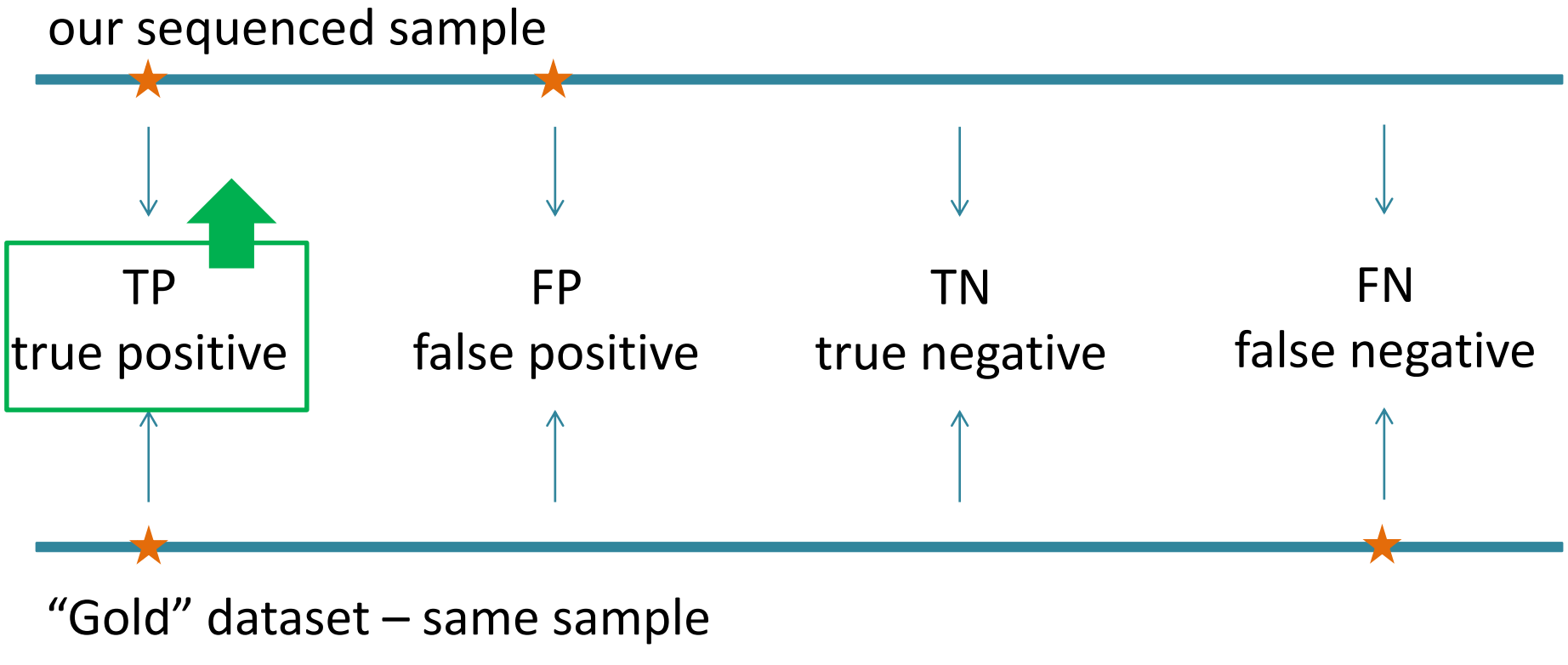


True positive
False positive
True negative
False negative

% : 0-100

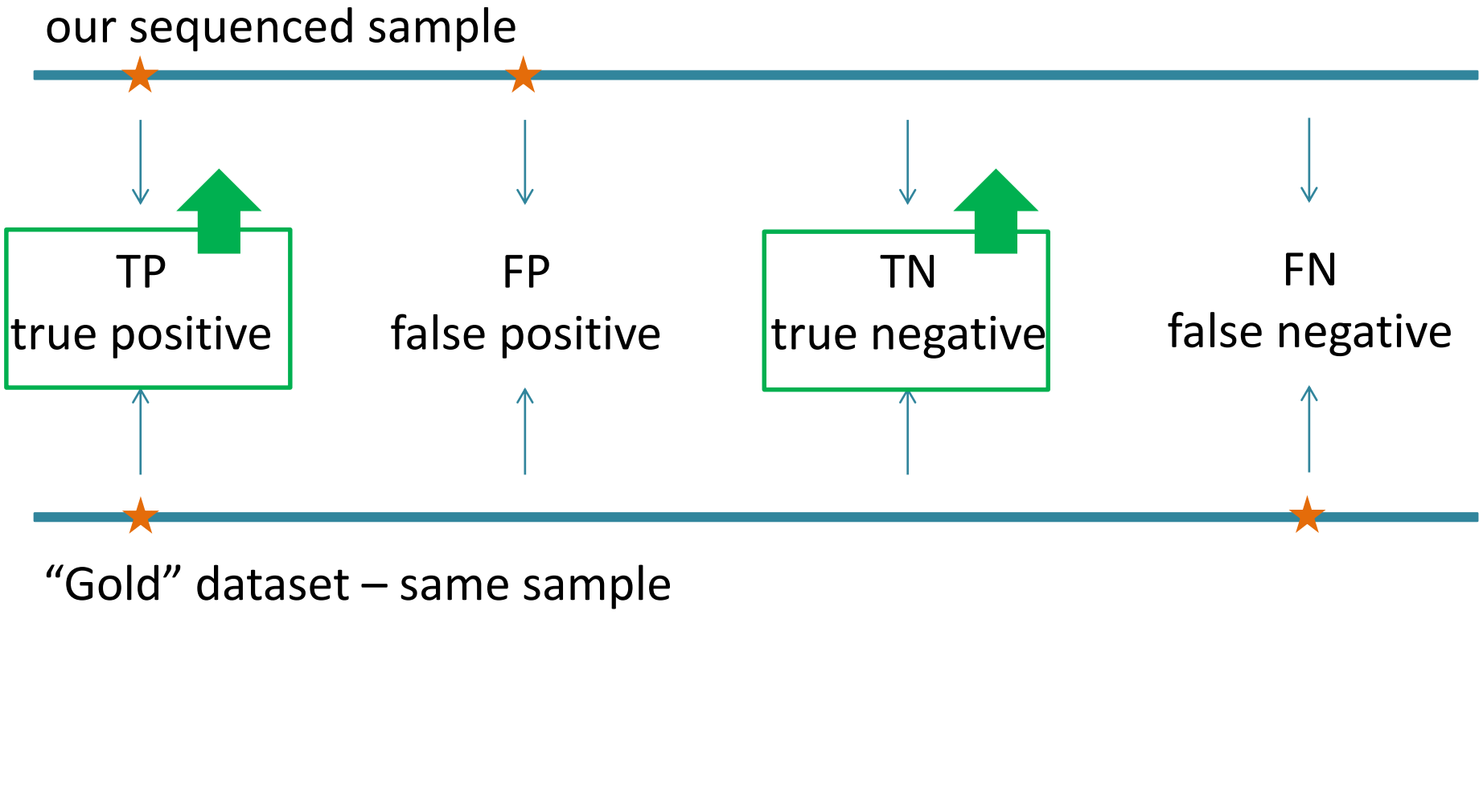
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



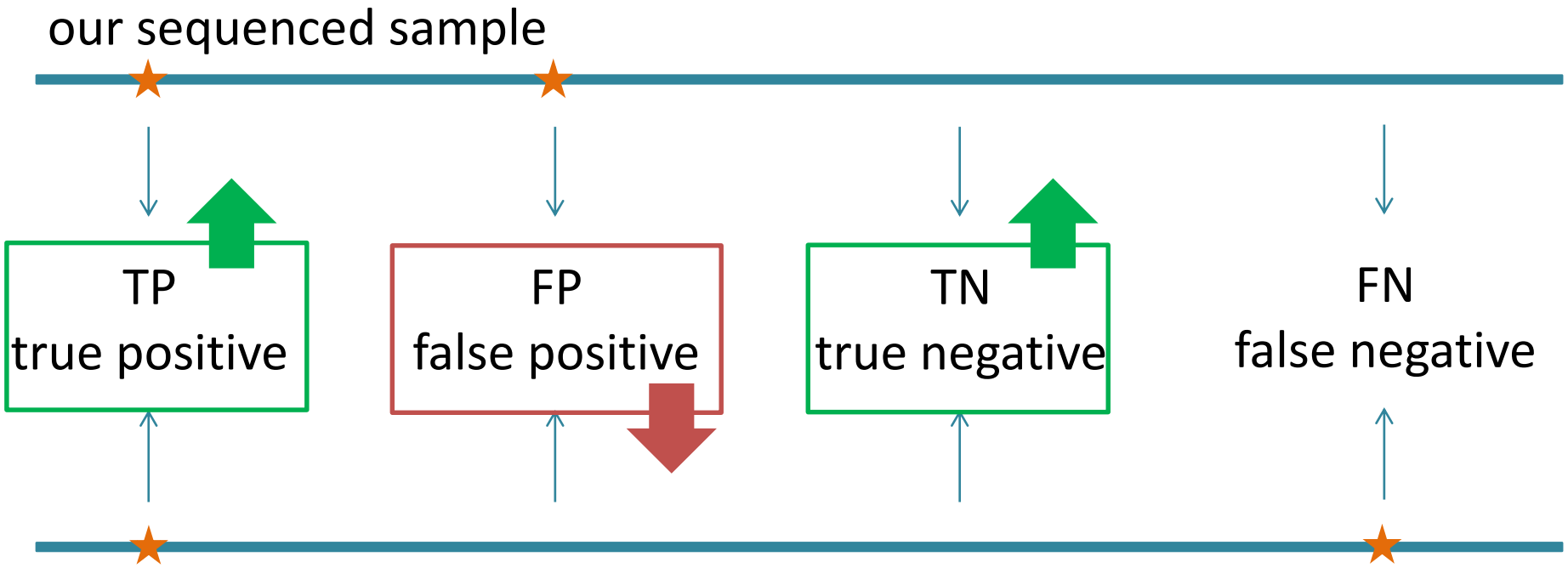
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



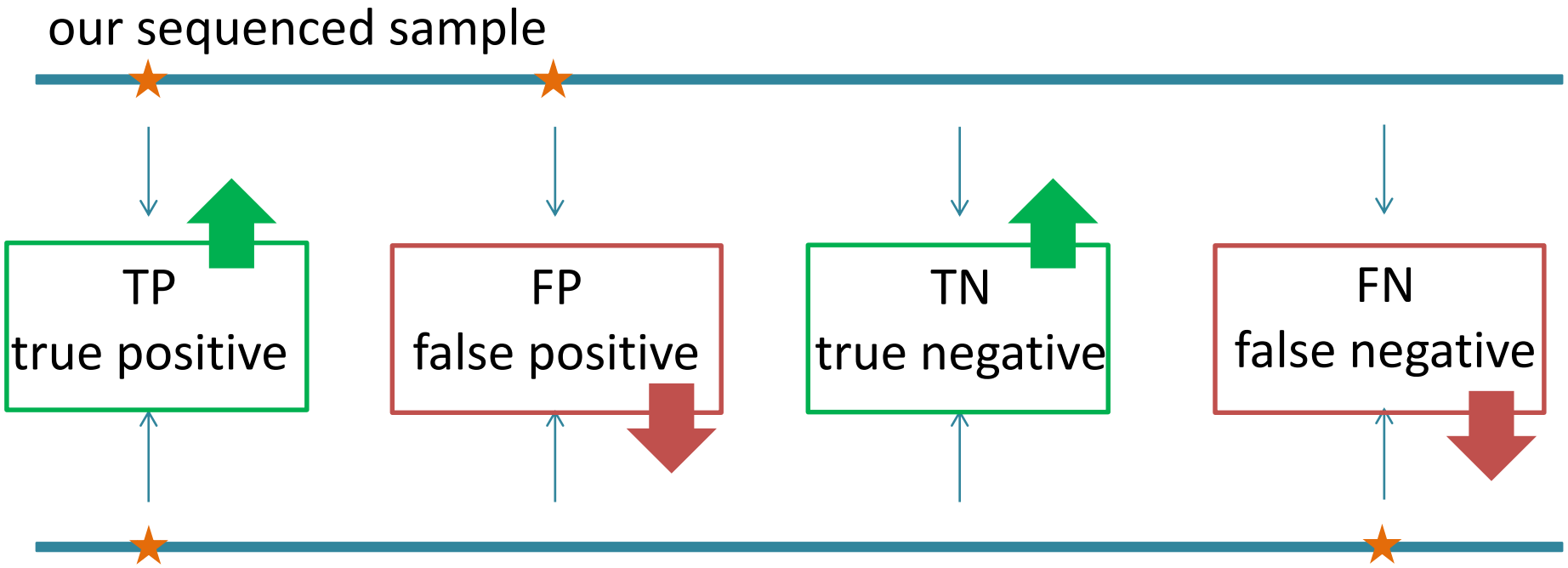
5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



5- Variant filtering and validation: **VALIDATION**

SAME SAMPLE(S)!!!



5- Variant filtering and validation: **VALIDATION**

True positive
False positive
True negative
False negative

5- Variant filtering and validation: **VALIDATION**

True positive
False positive
True negative
False negative

True positive
True negative

False positive
False negative

5- Variant filtering and validation: **VALIDATION**

True positive
False positive
True negative
False negative

True positive
True negative

False positive
False negative

Correct calls

5- Variant filtering and validation: **VALIDATION**

True positive
False positive
True negative
False negative

True positive
True negative

Correct calls

False positive
False negative

False calls / Sequencing error

5- Variant filtering and validation: **VALIDATION**

Human X chr	**True calls (%)**	**False calls (%)**
Ion Torrent vs Complete Genomics (427,000 genotypes)	99.9995	0.0005
Ion Torrent vs Illumina (3.65*10^6 genotypes)	99.99997	0.00003

5- Variant filtering and validation: **VALIDATION**

Human X chr	**True calls (%)**	**False calls (%)**
Ion Torrent vs Complete Genomics (427,000 genotypes)	99.9995	0.0005
Ion Torrent vs Illumina (3.65*10^6 genotypes)	99.99997	0.00003

Shark autosomal data	**Concordant calls (%)**	**Sequencing error (%)**
2 samples (independent runs) ~741,000 genotypes	99.986	0.014

5- Variant filtering and validation: **FILTERING** and **VALIDATION**

After filtering and validation we have a high quality vcf file ready for downstream analyses

(population genetics, medical genetics, association studies, SNP discovery, etc…)