

Variant-calling Workflow

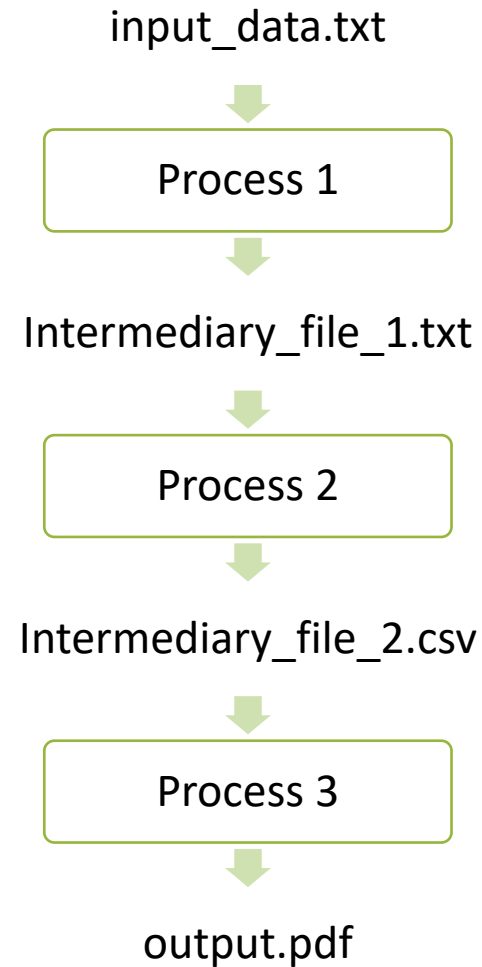
Overview

- Workflows
- Basic variant calling in one sample
- Basic variant calling in cohort
- Introduction to exercise

In separate talk Thursday at 9:

- GATK's Best practices

What is a workflow





Today:

- Basic variant calling workflow for one sample
- Extend to multiple samples

Tomorrow:

- GATK's Best practices

Example: Basic workflow, one sample



HG00097_1.fastq
HG00097_2.fastq

FASTQ files

Alignment

HG00097.bam

BAM files

VariantCalling

HG00097.vcf

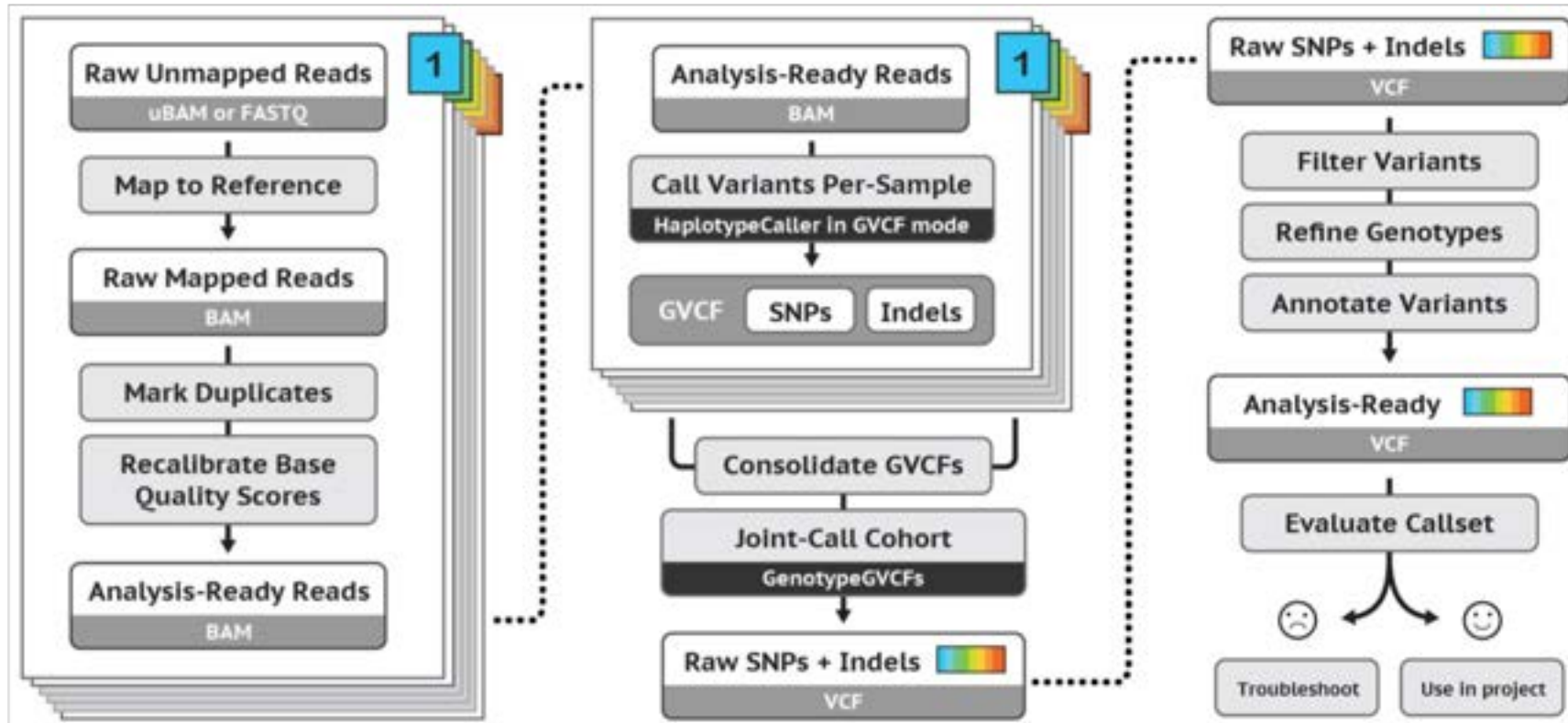
VCF files





1. Create a new output file in each process
2. Don't overwrite the input file
3. Use informative file names
4. Include information of the process + sample
5. Correct name extension e.g. .bam, .vcf, ...

GATK's best practices workflow for germline short variant discovery



Basic Variant Calling in one sample

Alignment



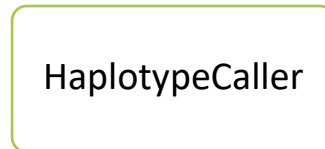
HG00097_1.fastq
HG00097_2.fastq

FASTQ files



HG00097.bam

BAM files



HG00097.vcf

VCF files



The reference genome

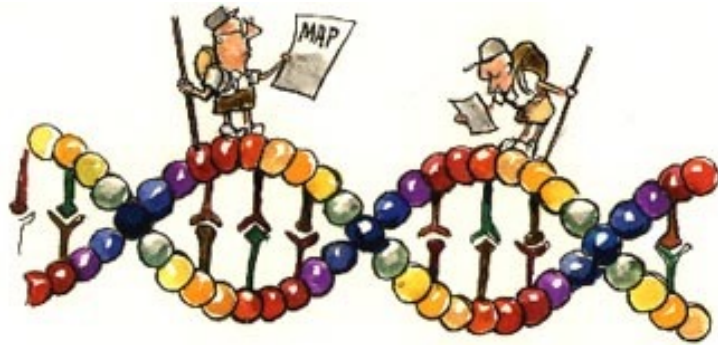


A reference genome is a haploid nucleic acid sequence which represents a species genome.

The first draft of the human genome contained 150,000 gaps.

GRCh37: 250 gaps

We will work with GRCh37 in the lab.



Keep track of the reference version!



The reference genome sequence is used as input in many bioinformatics applications for NGS data:

- mapping
- variant calling
- annotation

You must keep track of which version of the reference genome your data was mapped to.

The same version must be used in all downstream analyses.

Burrows-Wheeler Aligner



<http://bio-bwa.sourceforge.net>

[Home](#)

Burrows-Wheeler Aligner

BWA: [SF project page](#)

[Introduction](#) [FAQ](#)

BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The first algorithm is designed for Illumina sequence reads up to 100bp, while the rest two for longer reads ranging from 70bp to 1Mbp. BWA-MEM and BWA-SW share similar features: long-read support and split alignment, but BWA-MEM, which is the generally recommended for high-quality queries as it is faster and more accurate. BWA-MEM also has better performance than BWA-backtrack for 70-100bp reads.

How can I cite BWA?

The short read alignment component (bwa-short) has been published in:
Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25:1754-60. [PMID: 19451168]

If you use BWA-SW, please cite:
Li H. and Durbin R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler Transform. *Bioinformatics*, Epub. [PMID: 20477299]

(See also Errata below for a minor correction to the formulae in the above papers.)

There are three algorithms, which one should I choose?

For 70bp or longer Illumina, 454, Ion Torrent and Sanger reads, contigs and BAC sequences, BWA-MEM is usually the preferred algorithm. For short sequences, BWA-backtrack may be better. BWA-SW may be better for long reads.

Burrows-Wheeler transform of reference genome

0: googol\$		0: \$googo l
1: oogol\$g		1: 3 gol\$go o
2: ogol\$go		2: 0 googol \$
3: gol\$goo	String Sorting →	3: 5 l\$goog o
4: ol\$goog		4: 2 ogol\$g o
5: l\$googo		5: 4 ol\$goo g
6: \$googol		6: 1 oogol\$ g

Pos	i	$S(i)$	$B[i]$
	↓		↓
$X = \text{googol\$}$			lo\$oogg
		↓	
		(6,3,0,5,2,4,1)	

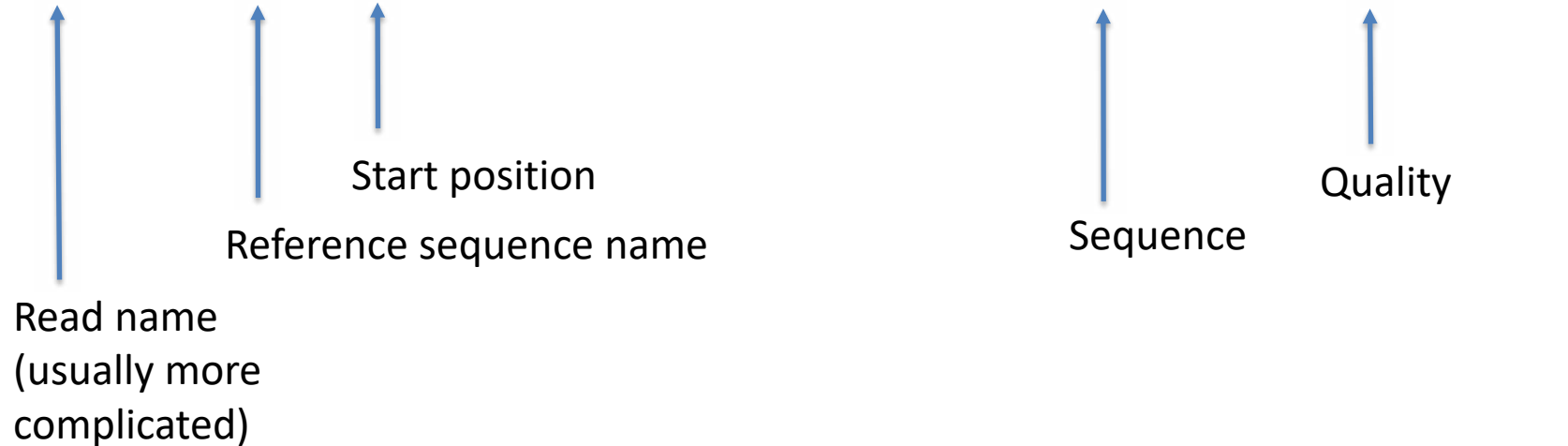


HEADER SECTION

```
@HD VN:1.6SO:coordinate
@SQ SN:2 LN:243199373
@PG ID:bwaPN:bwaVN:0.7.17-r1188 CL:bwa mem -t 1 human_g1k_v37_chr2.fasta HG00097_1.fq HG00097_2.fq
@PG ID:samtools PN:samtools PP:bwaVN:1.10 CL:samtools sort
@PG ID:samtools.1 PN:samtools PP:samtools VN:1.10 CL:samtools view -H HG00097.bam
```

ALIGNMENT SECTION

```
Read_001 99 2 3843448 0 101M = 3843625 278 TTTGGTTCCATATGAACTTT 0F<BFB<FFFBFBBBBFBFB
Read_001 147 2 3843625 0 101M = 3843448 -278 TTATTTTCATTGAGCAGTGGT FBBI7IIFIB<BBBB<BBFF
Read_002 163 2 4210055 0 101M = 4210377 423 TGGTACCAAAACAGAGATAT 0IIFBFFFIIIFIFIFBFBF
Read_003 99 2 4210066 0 101M = 4210317 352 CAGAGATATAGATCAATGGA 0IIFFFIFFFIFIFIIIIIF
```



Convert to Bam



Bam file is a binary representation of the Sam file



- Most large files we work with, such as the reference genome (**.fasta**) and the aligned reads (**.bam**) need an index
- The index is a small file
- Allows efficient access to the large file
- Different indices for different file types
- BWA index = Burrows-Wheeler transform of reference genome (**several files**)

Variant calling



HG00097_1.fastq

HG00097_2.fastq

FASTQ files



BWA mem



HG00097.bam

BAM files



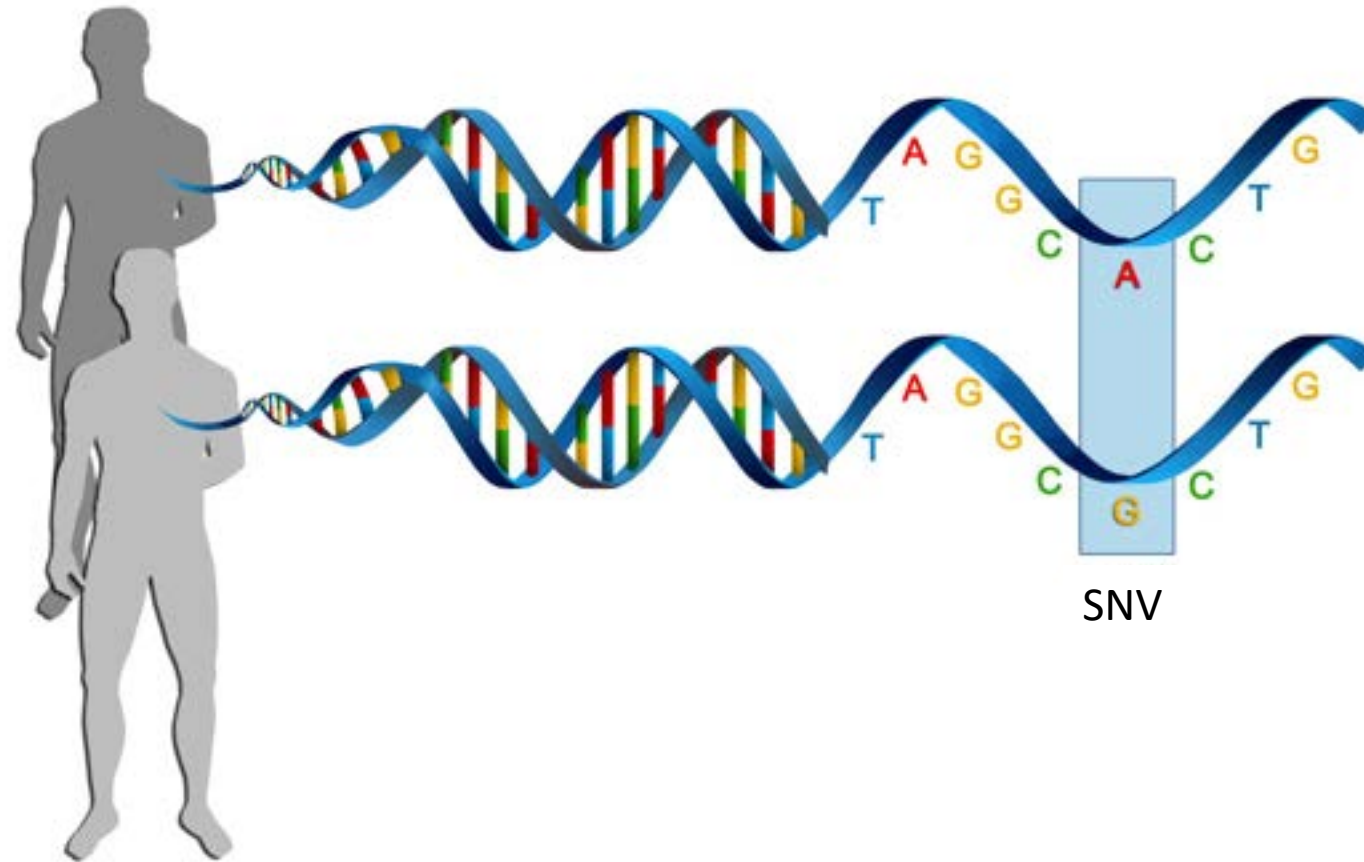
HaplotypeCaller



HG00097.vcf

VCF files

Genetic variation



Genetic variation = differences in DNA among individuals of the same species

Alignment



Detecting variants in reads



Reference:

...GTGCGTAGACTGCTAGATCGAAGA...

Sample:

...GTGCGTAGACTG**A**TAGATCGAAGA...

...GTGCGTAGACTG**A**TAGATCGAAGA...

...GTGCGTAGACTGCTAGATCGAAGA...

...GTGCGTAGACTGCTAGATCGAAGA...

...GTGCGTAGACTG**A**TAGATCGAAGA...

...GTGCGTAGACTG**A**TAGATCGAAGA...

...GTGCGTAGACTGCTAGATCGAAGA...

...GTGCGTAGACTG**A**TAGATCGAAGA...

...GTGCGTAGACTGCTAGATCGAAGA...

...GTGCGTAGACTG**A**TAGATCGAAGA...



TGGGCTTTTCCAACAGGTATATCTTCCCCGCTAGCTCGCTAGCTACTTCAAATTCCT

Reference allele AGCTCGCTA

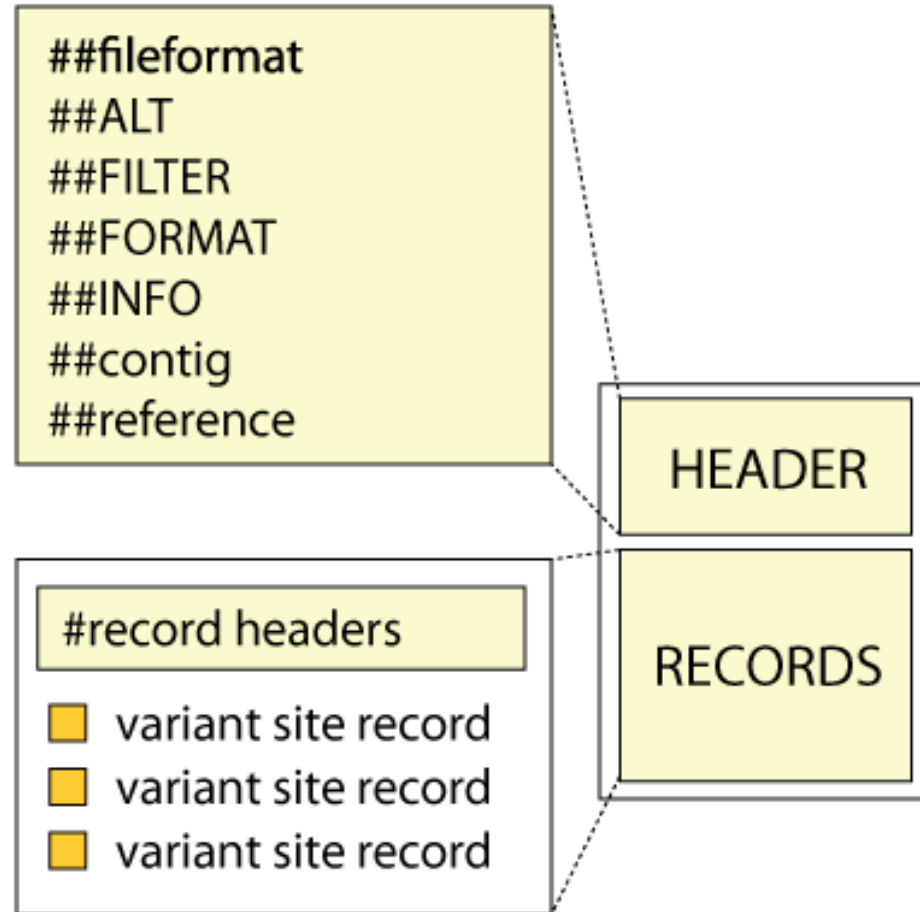
Alternative allele AGCTAGCTA

Reference allele = the allele in the reference genome

Alternative allele = the allele NOT in the reference genome



Variant Call Format (VCF)



Variant Call Format (VCF)



```
##fileformat=VCFv4.2
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##contig=<ID=2,length=243199373>
##source=HaplotypeCaller
#CHROM  POS          ID          REF          ALT          QUAL        FILTER        INFO          FORMAT        HG00097
2       136220992    .           G            GT           30.64       .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,2:5
2       136226814    .           GAC          G            44.60       .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:4,2:6
2       136234279    .           C            T            102.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,4:7
2       136234284    .           C            T            102.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,4:7
2       136263277    .           T            A            148.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:8,5:13
...
...
```

Basic workflow, one sample



HG00097_1.fastq
HG00097_2.fastq

FASTQ files

Alignment

HG00097.bam

BAM files

VariantCalling

HG00097.vcf

VCF files



Variant Call Format (VCF)

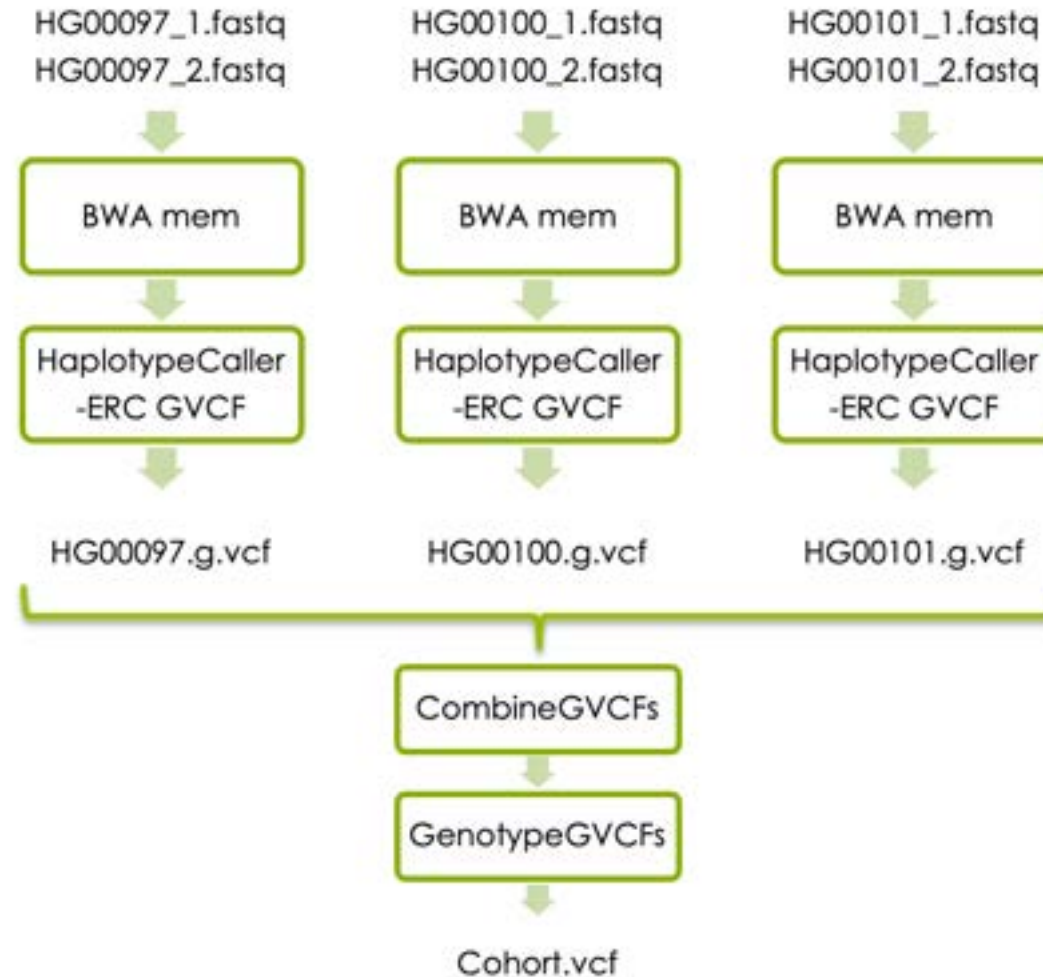


```
##fileformat=VCFv4.2
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##contig=<ID=2,length=243199373>
##source=HaplotypeCaller
#CHROM  POS          ID          REF          ALT          QUAL        FILTER        INFO          FORMAT        HG00097
2       136220992    .           G            GT           30.64       .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,2:5
2       136226814    .           GAC          G            44.60       .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:4,2:6
2       136234279    .           C            T            102.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,4:7
2       136234284    .           C            T            102.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:3,4:7
2       136263277    .           T            A            148.60      .            AC=1;AF=0.500;AN=2  GT:AD:DP     0/1:8,5:13
...
...
```

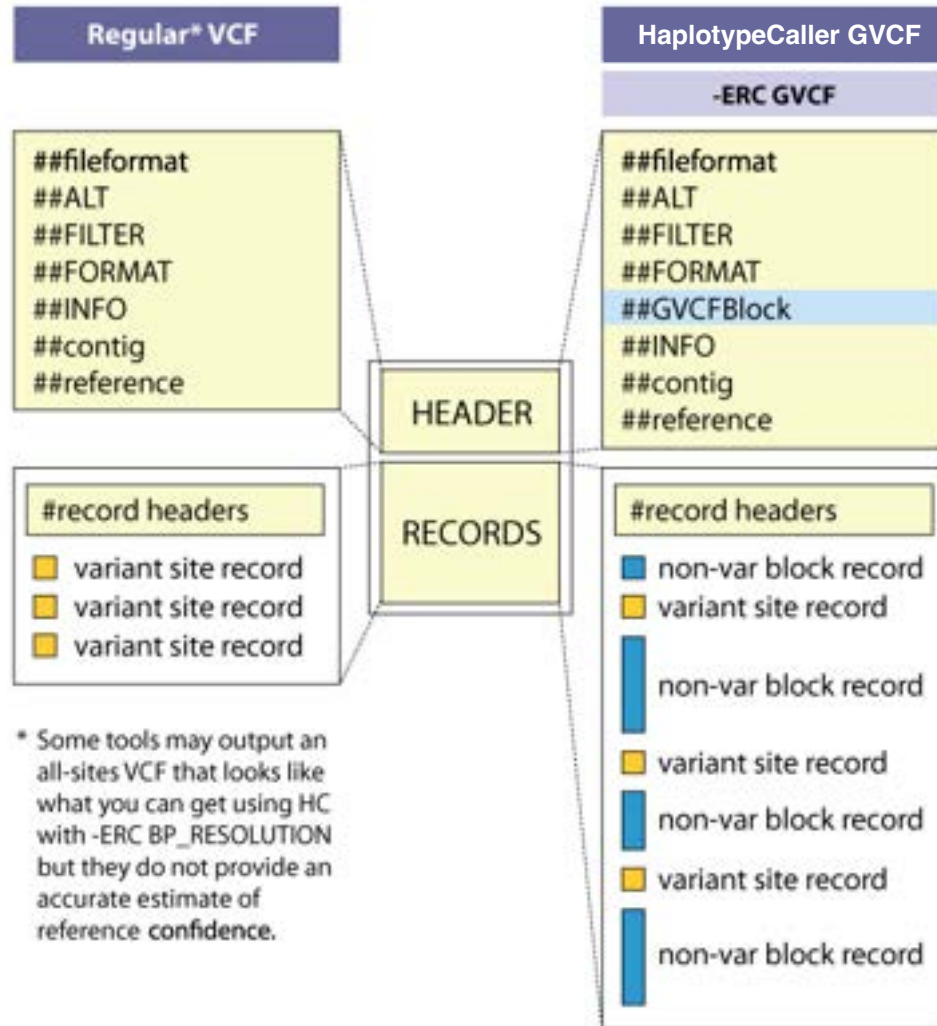
Basic variant calling in cohort



Basic variant calling in cohort

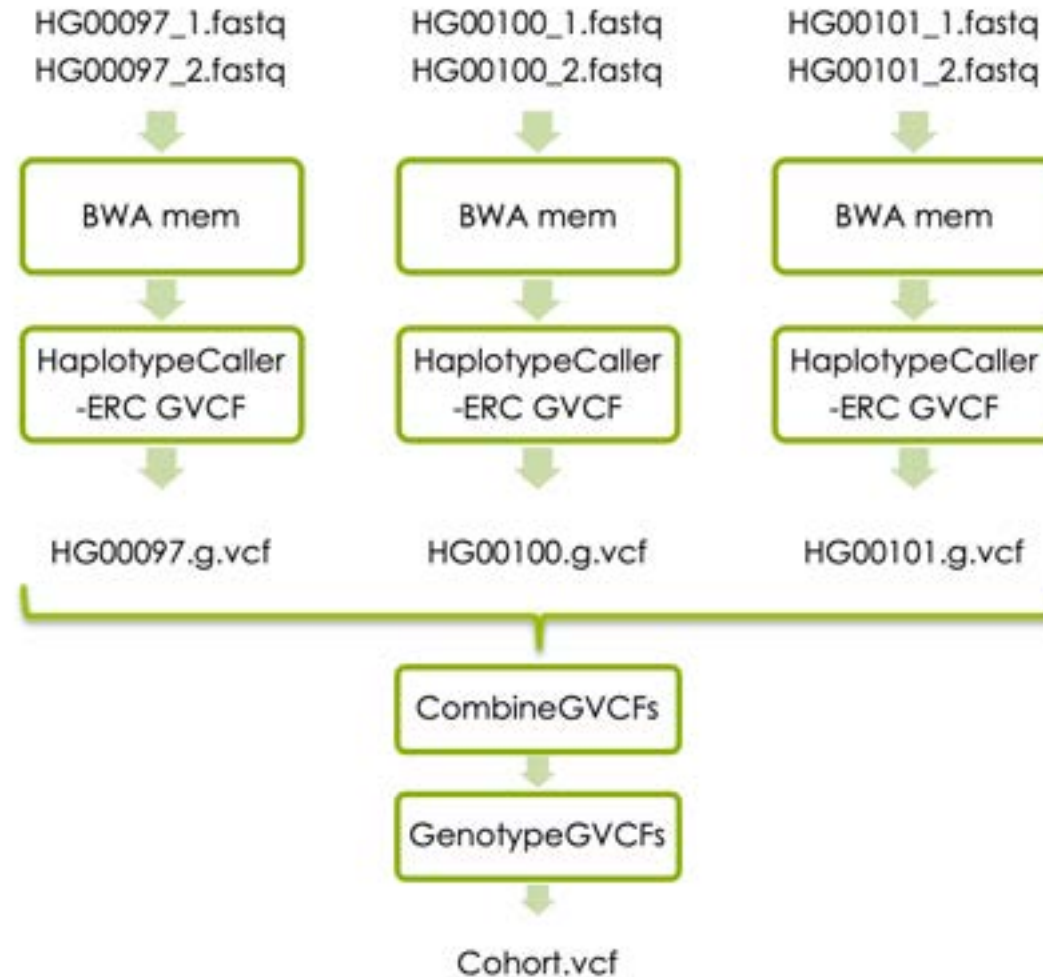


GVCF Files are valid VCFs with extra information



- GVCF has records for all sites, whether there is a variant call there or not.
- The records include an accurate estimation of how confident we are in the determination that the sites are homozygous-reference or not.
- Adjacent non-variant sites merged into blocks

Basic variant calling in cohort



Variant Call Format (VCF)



```
##fileformat=VCFv4.2
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele at this location">
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##contig=<ID=2,length=243199373>
##source=CombineGVCFs
##source=GenotypeGVCFs
##source=HaplotypeCaller
```

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	HG00097	HG00100	HG00101
2	136045826	.	G	A	167.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:8,0:8	0/0:13,0:13	0/1:1,5:6
2	136046443	.	CGT	C	129.27	.	AC=3;AF=0.500;AN=6	GT:AD:DP	0/0:8,0:8	0/1:3,1:4	1/1:0,4:4
2	136047387	.	T	C	186.27	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:6,0:6	0/0:16,0:16	0/1:4,6:10
2	136048649	.	C	G	127.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:13,0:13	0/0:9,0:9	0/1:1,4:5
2	136052318	.	C	T	107.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:7,0:7	0/0:13,0:13	0/1:3,3:6



Today's lab



1000 Genomes data



- Low coverage WGS data
- 3 samples
- Small region on chromosome 2

About the samples:

<https://www.internationalgenome.org/data-portal/sample>

The Lactase enzyme

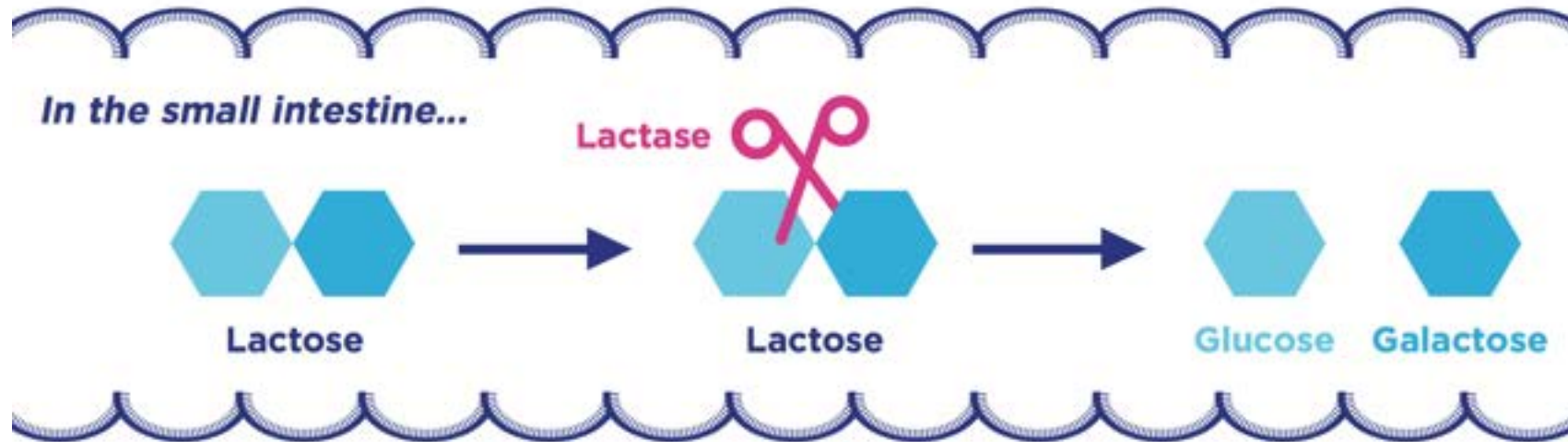
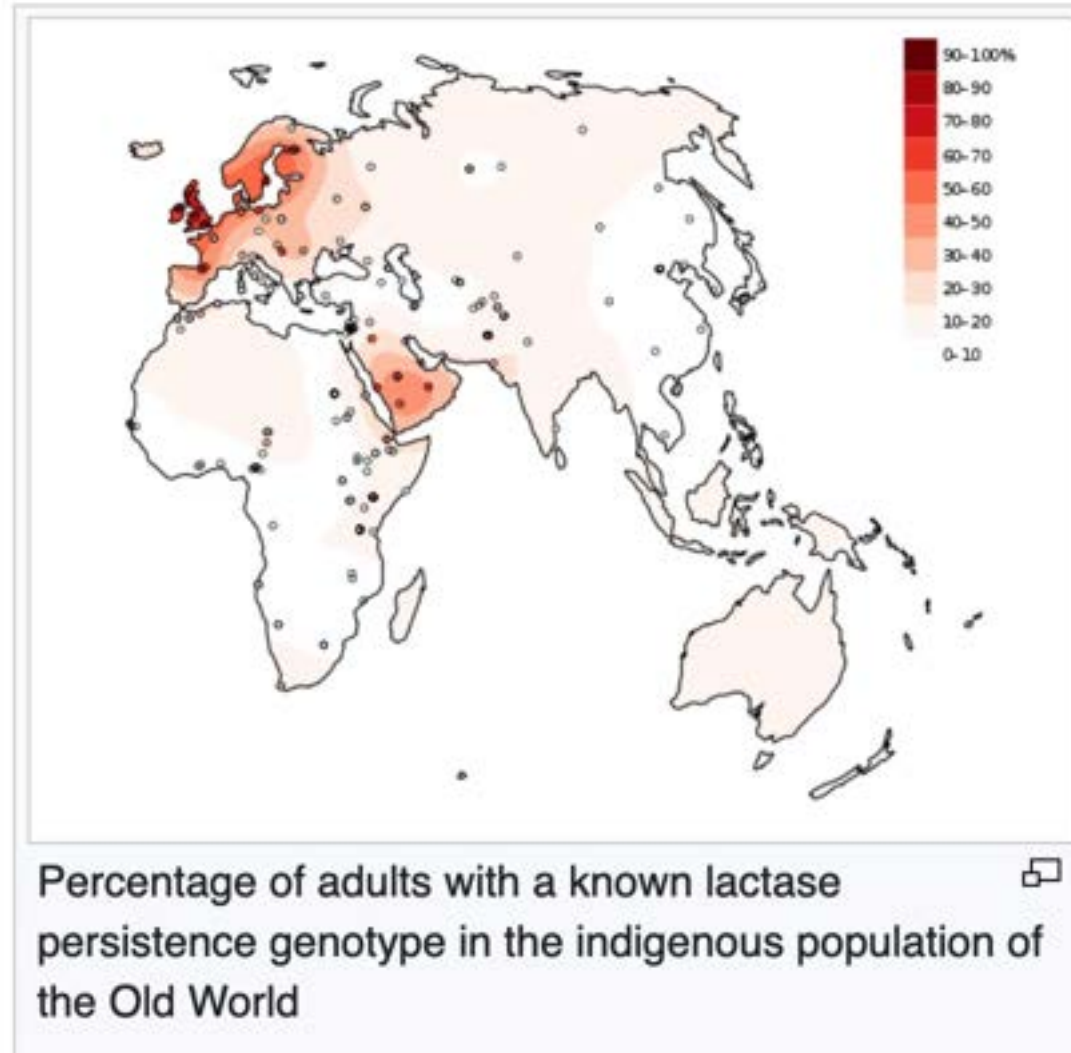


Figure 2. Lactose digestion in the intestine.

- All mammals produce lactase as infants
- Some human produce lactase in adulthood
- Genetic variation upstream of the *LCT* gene cause the lactase persistent phenotype (lactose tolerance)

The Lactase enzyme



Part 1:

Variant calling in one sample

Basic variant calling in one sample



HG00097_1.fastq
HG00097_2.fastq

FASTQ files

BWA mem

HG00097.bam

BAM files

HaplotypeCaller

HG00097.vcf

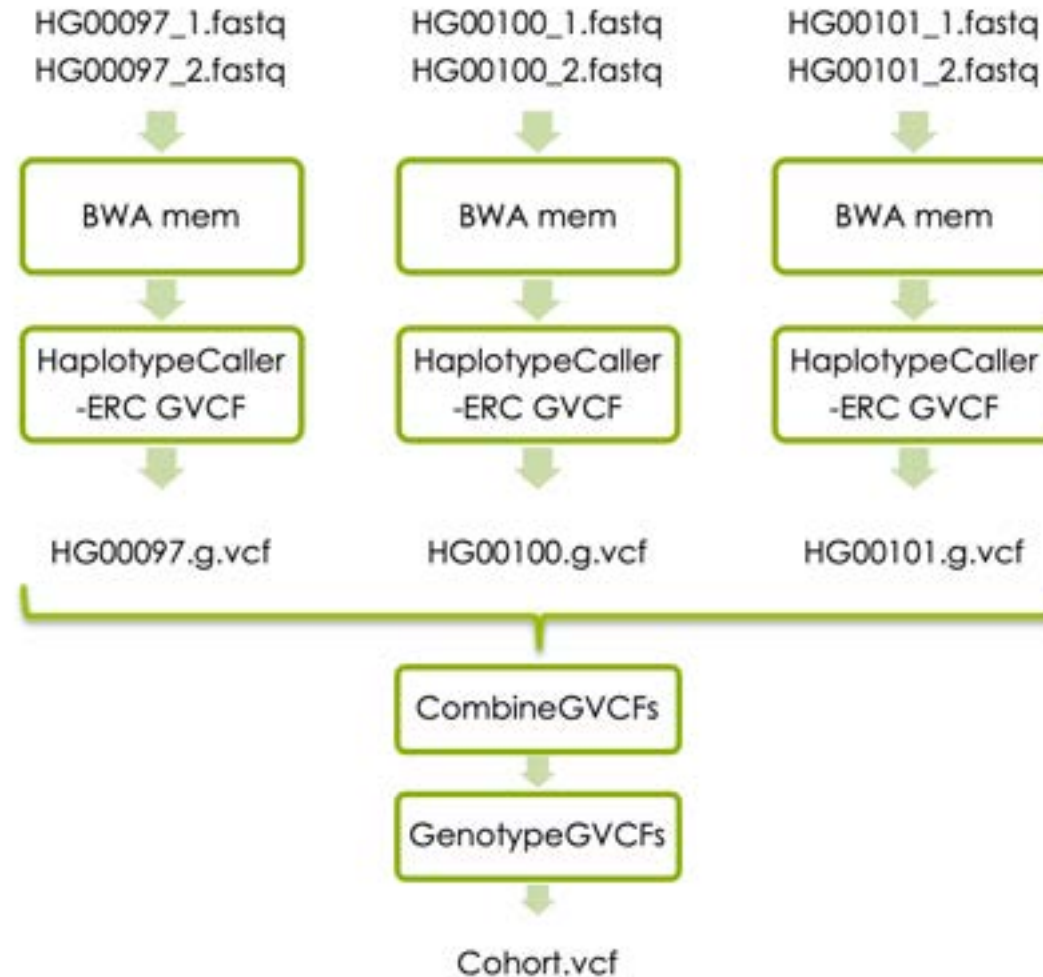
VCF files



Part 2:

Variant calling in cohort

Joint variant calling workflow





1. Create a new output file in each process
2. Don't overwrite the input file
3. Use informative file names
4. Include information of the process + sample
5. Correct name extension e.g. .bam, .vcf, ...

Part 3:

**Follow GATK best practices for short
variant discovery**



gatk

User Guide Tool Index Blog Forum DRAGEN-GATK Events Download GATK Sign in

Genome Analysis Toolkit

Variant Discovery in High-Throughput Sequencing Data

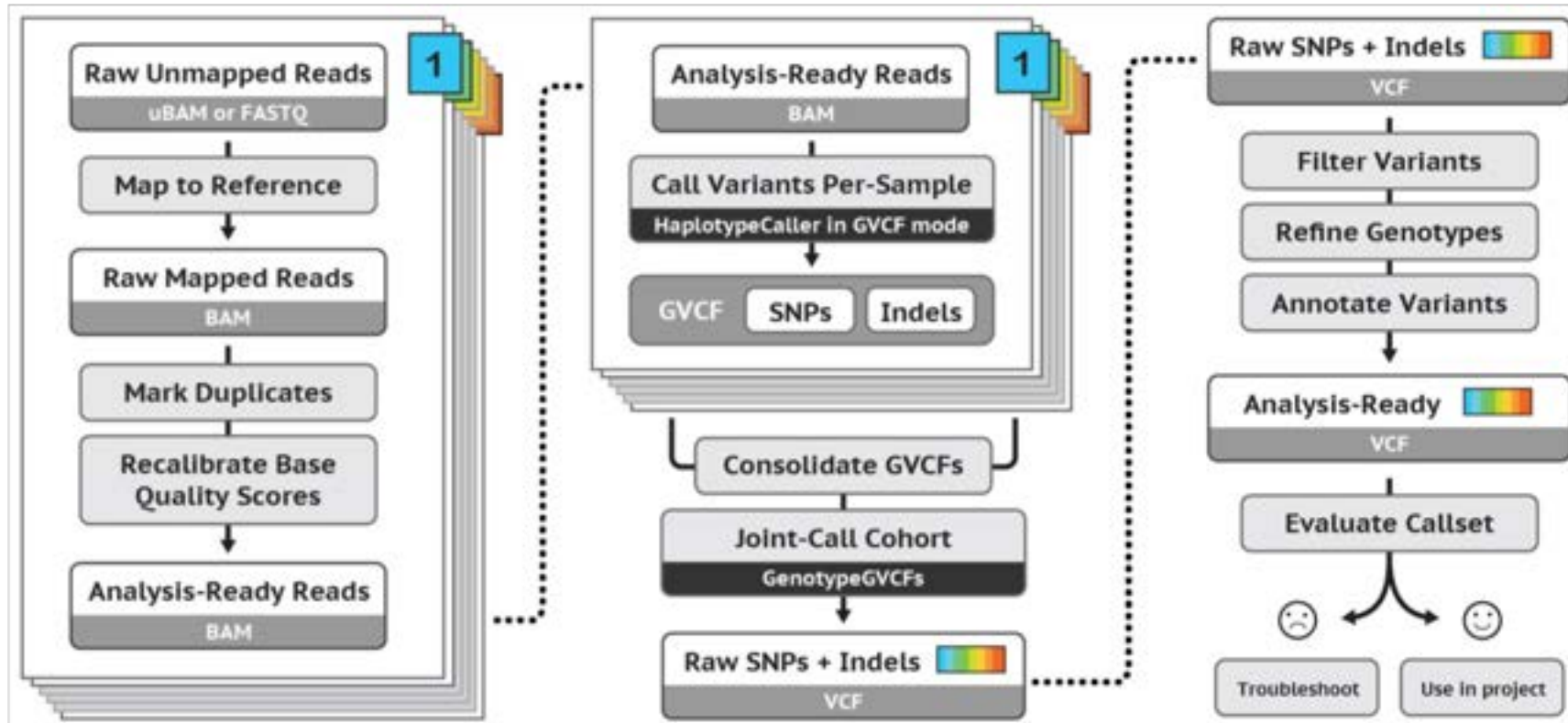
Sequencing → **gatk best practices™** → VARIANTS

Developed in the Data Sciences Platform at the [Broad Institute](#), the toolkit offers a wide variety of tools with a primary focus on variant discovery and genotyping. Its powerful processing engine and high-performance computing features make it capable of taking on projects of any size. [Learn more](#)

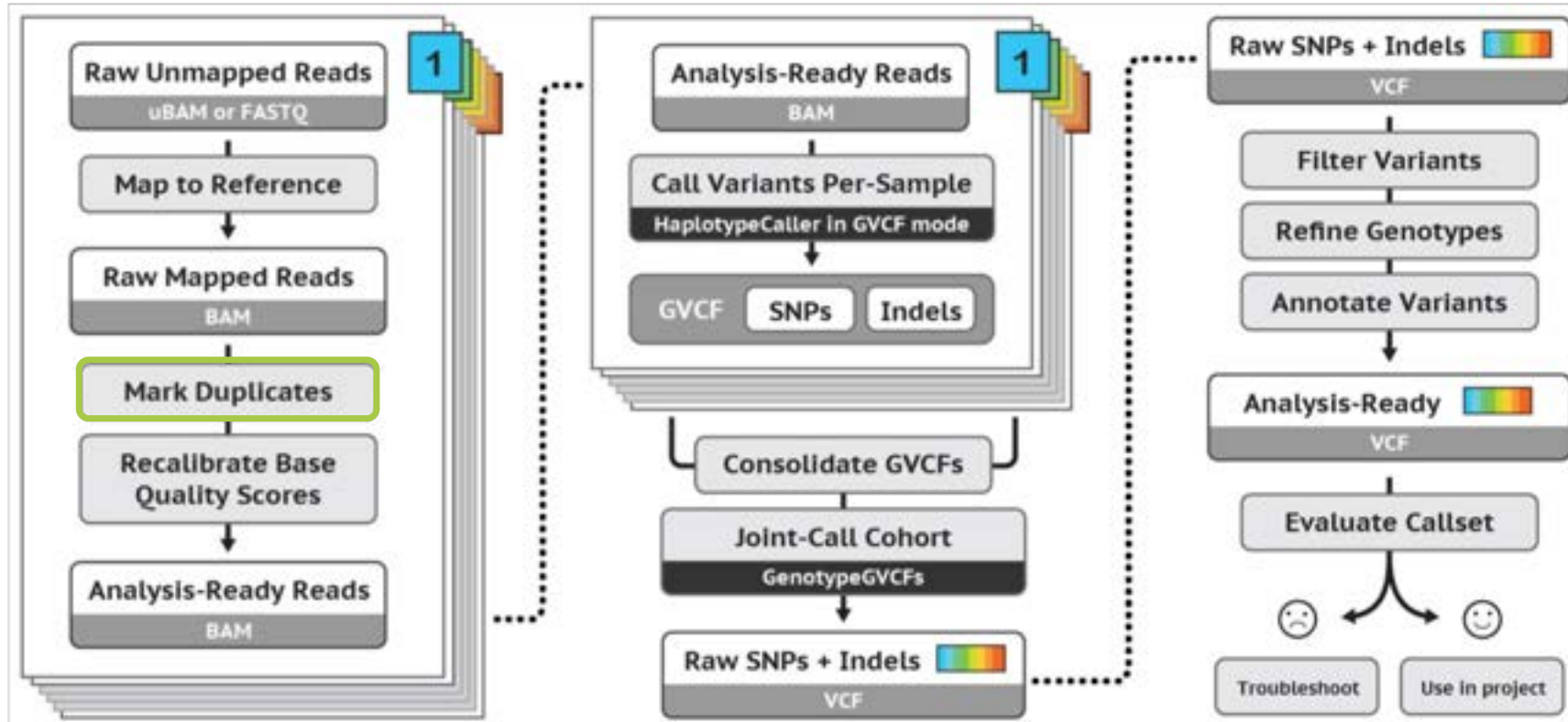
Find answers to your questions. Stay up to date on the latest topics. Ask questions and help others.

- Getting Started**
Best practices, tutorials, and other info to get you started
- Technical Documentation**
Algorithms, glossary, and other detailed resources
- Announcements**
Blog and events
- Tool Index**
Purpose, usage and options for each tool
- Forum**
Ask our team for help and report issues
- GATK Showcase on Terra**
Check out these fully configured workspaces
- DRAGEN-GATK**
Learn more about DRAGEN-GATK
- Download latest version of GATK**
The GATK package download includes all released GATK tools
- Run on Cloud**
- Run on HPC**

GATK's best practices workflow for germline short variant discovery



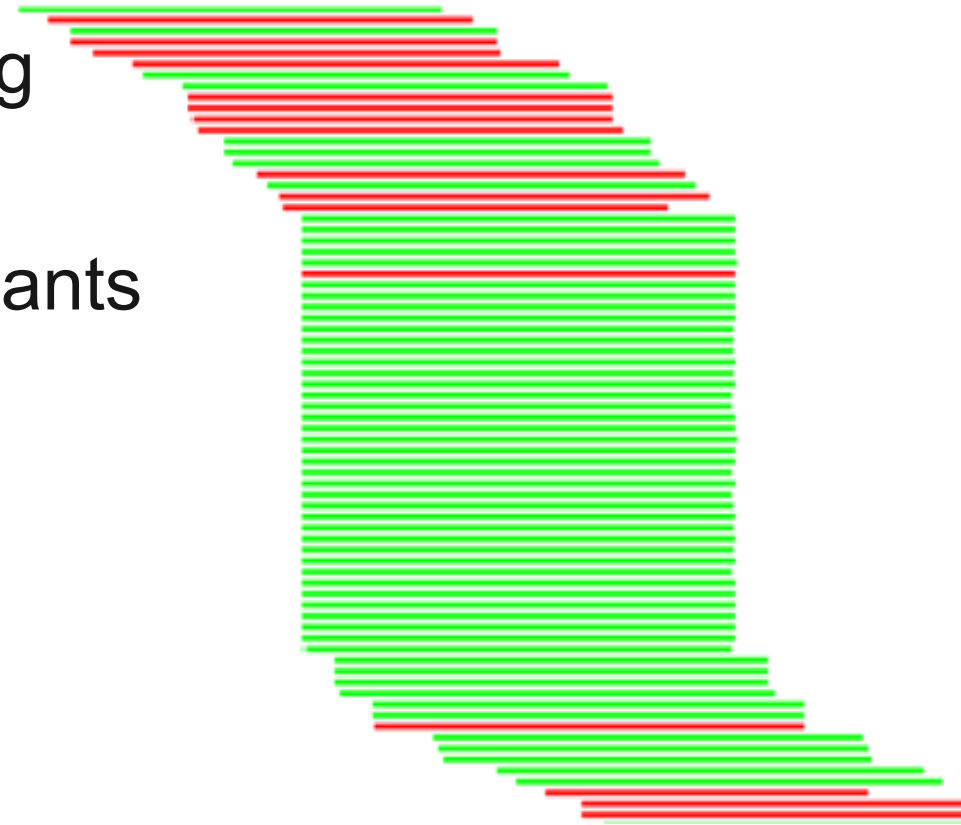
Mark Duplicates



Duplicate reads



- PCR duplicates - library preparation
- Optical duplicates - sequencing
- Don't add unique information
- Gives false allelic ratios of variants
- Should be removed/marked





Need Help?

Search our documentation

MarkDuplicates



GATK / Tool Index / 4.0.11

MarkDuplicates (Picard)

Follow



GATK Team

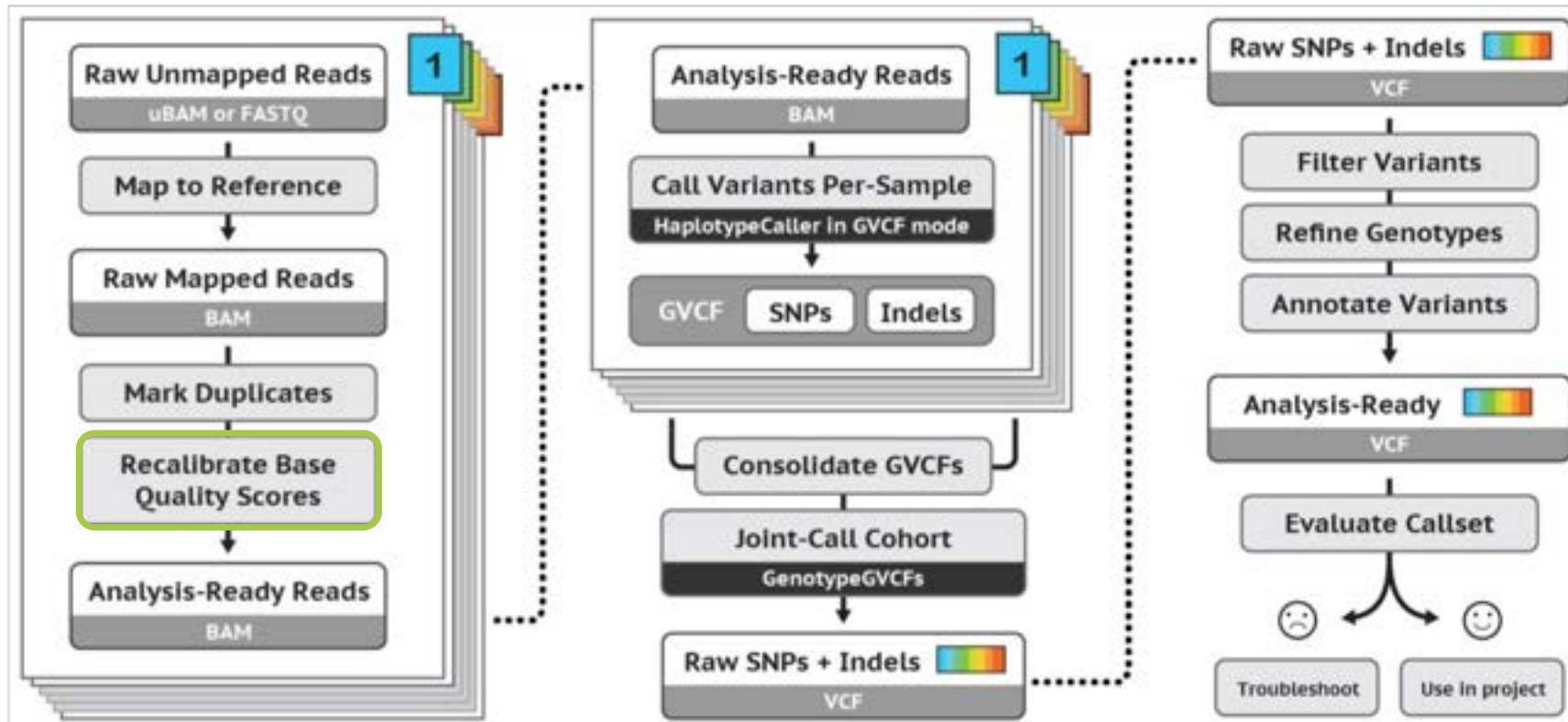
10 months ago · Updated

Identifies duplicate reads.

This tool locates and tags duplicate reads in a BAM or SAM file, where duplicate reads are defined as originating from a single fragment of DNA. Duplicates can arise during sample preparation e.g. library construction using PCR. See also [EstimateLibraryComplexity](#) for additional notes on PCR duplication artifacts. Duplicate reads can also result from a single amplification cluster, incorrectly detected as multiple clusters by the optical sensor of the sequencing instrument. These duplication artifacts are referred to as optical duplicates.

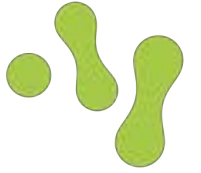
```
gatk --java-options -Xmx7g MarkDuplicates \  
-I input.bam \  
-O marked_duplicates.bam \  
-M marked_dup_metrics.txt
```

Base Quality Score Recalibration (BQSR)

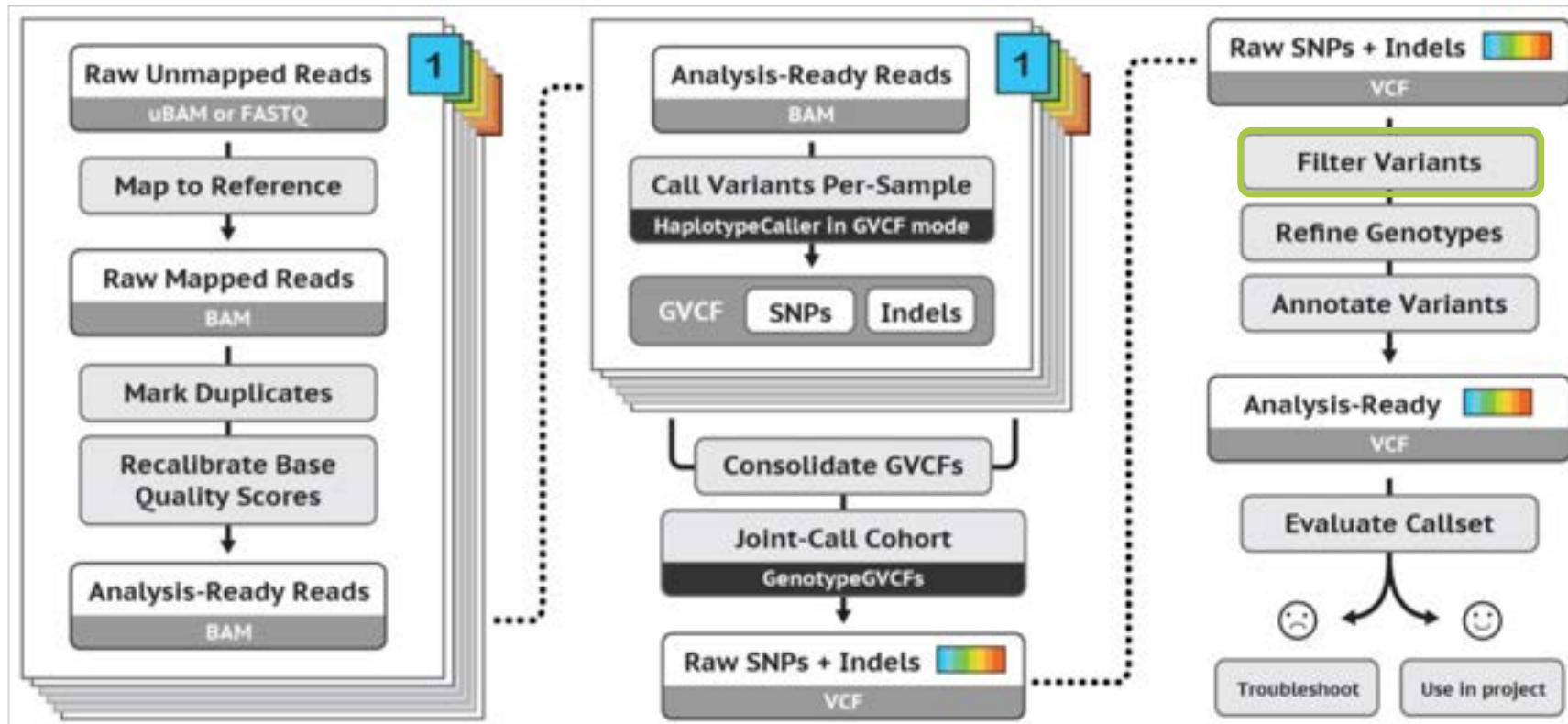




1. During base calling, the sequencer estimates a quality score for each base. This is the quality scores present in the fastq files.
2. Systematic (non-random) errors in the base quality score estimation can occur.
 - due to the physics or chemistry of the sequencing reaction
 - manufacturing flaws in the equipment
 - etc
3. Can cause bias in variant calling
4. **Base Quality Score Recalibration** helps to calibrate the scores so that they correspond to the real per-base sequencing error rate (phred scores)



Filter variants



[https://software.broadinstitute.org/gatk/best-practices/
Germline short variant discovery \(SNPs + Indels\)](https://software.broadinstitute.org/gatk/best-practices/Germline-short-variant-discovery-(SNPs+-Indels))

Variant Call Format (VCF)



```
##fileformat=VCFv4.2
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele at this location">
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##contig=<ID=2,length=243199373>
##source=CombineGVCFs
##source=GenotypeGVCFs
##source=HaplotypeCaller
```

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	HG00097	HG00100	HG00101
2	136045826	.	G	A	167.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:8,0:8	0/0:13,0:13	0/1:1,5:6
2	136046443	.	CGT	C	129.27	.	AC=3;AF=0.500;AN=6	GT:AD:DP	0/0:8,0:8	0/1:3,1:4	1/1:0,4:4
2	136047387	.	T	C	186.27	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:6,0:6	0/0:16,0:16	0/1:4,6:10
2	136048649	.	C	G	127.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:13,0:13	0/0:9,0:9	0/1:1,4:5
2	136052318	.	C	T	107.26	.	AC=1;AF=0.167;AN=6	GT:AD:DP	0/0:7,0:7	0/0:13,0:13	0/1:3,3:6



Variant quality score recalibration (VQSR):

For large data sets (>1 WGS or >30WES samples)

GATK has a machine learning algorithm that can be trained to recognise "likely false" variants

We do recommend to use VQSR when possible!

Hard filters:

For smaller data sets

Hard filters on information in the VCF file

For example: Flag variants with "Q < 40.0"

GATK's best practises





gatk

User Guide Tool Index Blog Forum DRAGEN-GATK Events Download GATK Sign in

Genome Analysis Toolkit

Variant Discovery in High-Throughput Sequencing Data

Sequencing → **gatk best practices™** → VARIANTS

Developed in the Data Sciences Platform at the [Broad Institute](#), the toolkit offers a wide variety of tools with a primary focus on variant discovery and genotyping. Its powerful processing engine and high-performance computing features make it capable of taking on projects of any size. [Learn more](#)

Find answers to your questions. Stay up to date on the latest topics. Ask questions and help others.

- Getting Started**
Best practices, tutorials, and other info to get you started
- Technical Documentation**
Algorithms, glossary, and other detailed resources
- Announcements**
Blog and events
- Tool Index**
Purpose, usage and options for each tool
- Forum**
Ask our team for help and report issues
- GATK Showcase on Terra**
Check out these fully configured workspaces
- DRAGEN-GATK**
Learn more about DRAGEN-GATK
- Download latest version of GATK**
The GATK package download includes all released GATK tools
- Run on Cloud**
- Run on HPC**



Home Pipelines Modules Tools Docs Events About Join nf-core

nf-core/sarek

Analysis pipeline to detect germline or somatic variants (pre-processing, variant calling and annotation) from WGS / targeted sequencing

Annotation | Cancer | EBV | Exonics | Exonics | RNA-sequencing | Somatic | Sarek-Sarek | Variant-calling | Whole-exome-sequencing | Whole-exome-sequencing

[Launch version 3.1.2](#)

<https://github.com/nf-core/sarek>

Introduction Results Usage docs Parameters Output docs Releases & Statistics 3.1.2

Introduction

nf-core/sarek is a workflow designed to detect variants on whole genome or targeted sequencing data. Initially designed for Human, and Mouse, it can work on any species with a reference genome. Sarek can also handle tumour / normal pairs and could include additional relapses.

The pipeline is built using [Nextflow](#), a workflow tool to run tasks across multiple compute infrastructures in a very portable manner. It uses Docker/Singularity containers making installation trivial and results highly reproducible. The [Nextflow DSL2](#) implementation of this pipeline uses one container per process which makes it much easier to maintain and update software dependencies. Where

Run with
nf-core Nextflow Tower

```
nf-core launch nf-core/sarek -r 3.1.2
```

Video introduction

nf-core/sarek