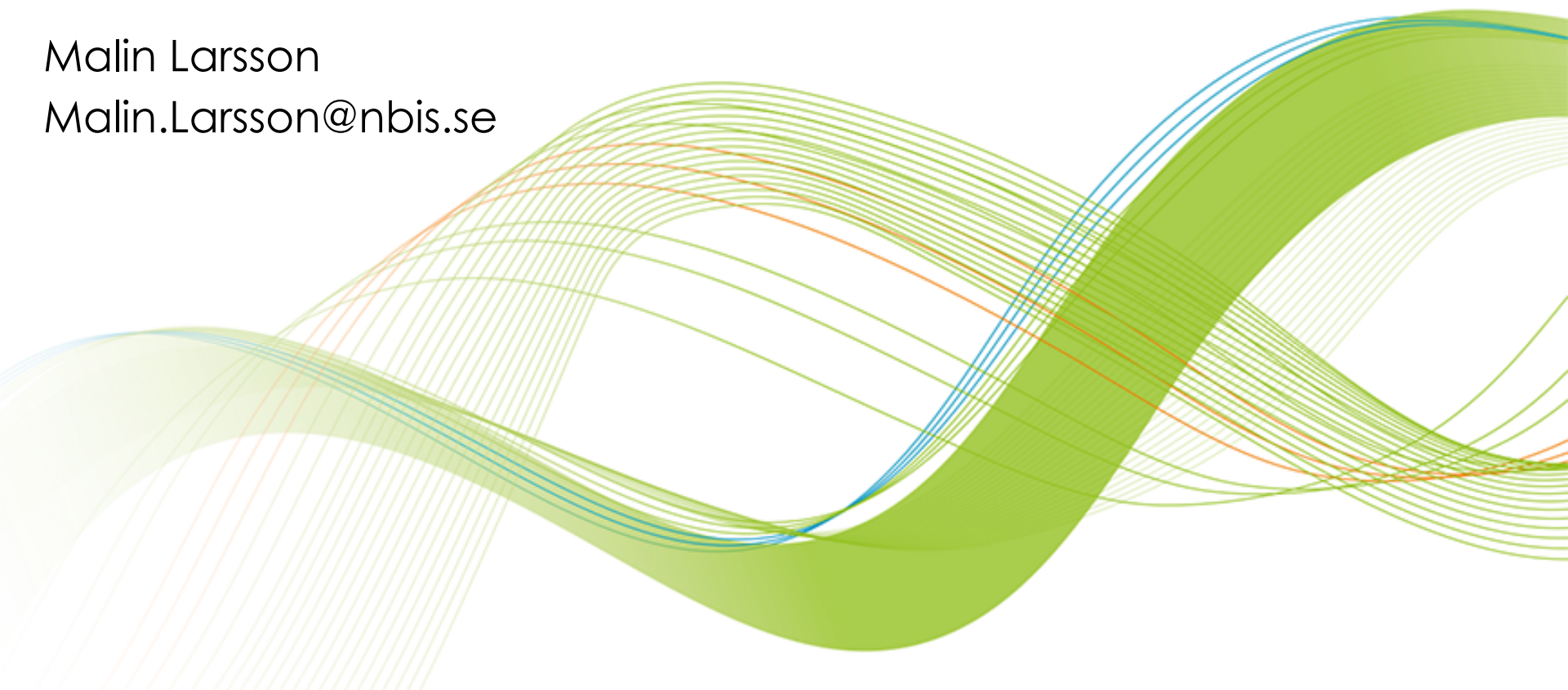


---

# Variant Calling Workflows

Malin Larsson

Malin.Larsson@nbis.se

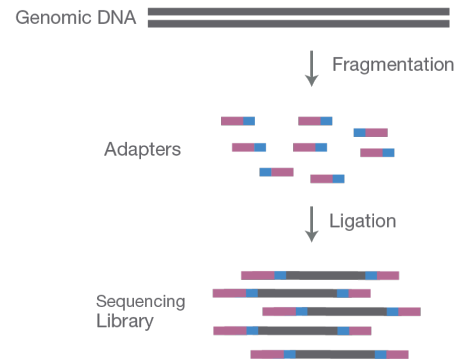


# Talk Overview

- The reference genome
- Genetic variation
- Workflows
- Basic variant calling in one sample
- Basic variant calling in cohort
- GATK Best practices
- Introduction to exercise

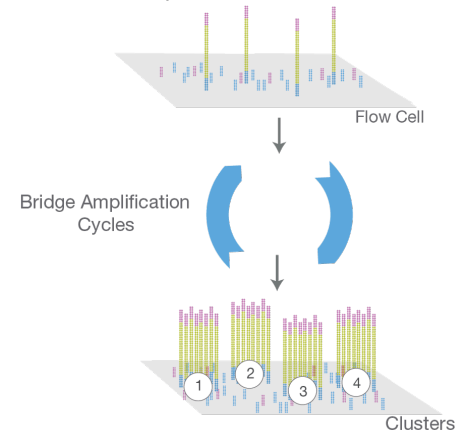
# Illumina Sequencing

## A. Library Preparation



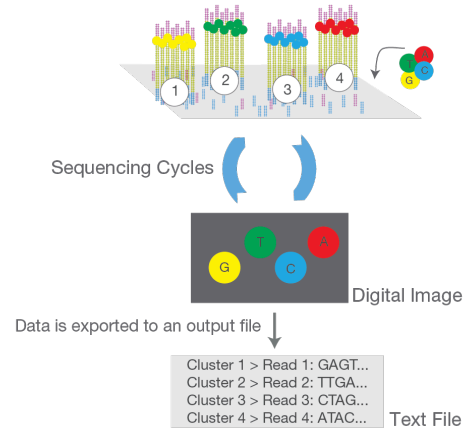
NGS library is prepared by fragmenting a gDNA sample and ligating specialized adapters to both fragment ends.

## B. Cluster Amplification



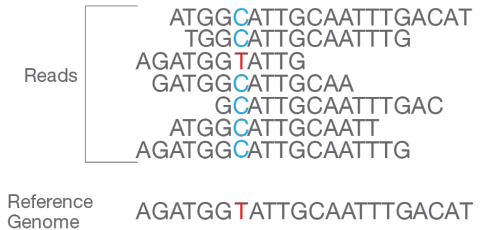
Library is loaded into a flow cell and the fragments are hybridized to the flow cell surface. Each bound fragment is amplified into a clonal cluster through bridge amplification.

## C. Sequencing



Sequencing reagents, including fluorescently labeled nucleotides, are added and the first base is incorporated. The flow cell is imaged and the emission from each cluster is recorded. The emission wavelength and intensity are used to identify the base. This cycle is repeated "n" times to create a read length of "n" bases.

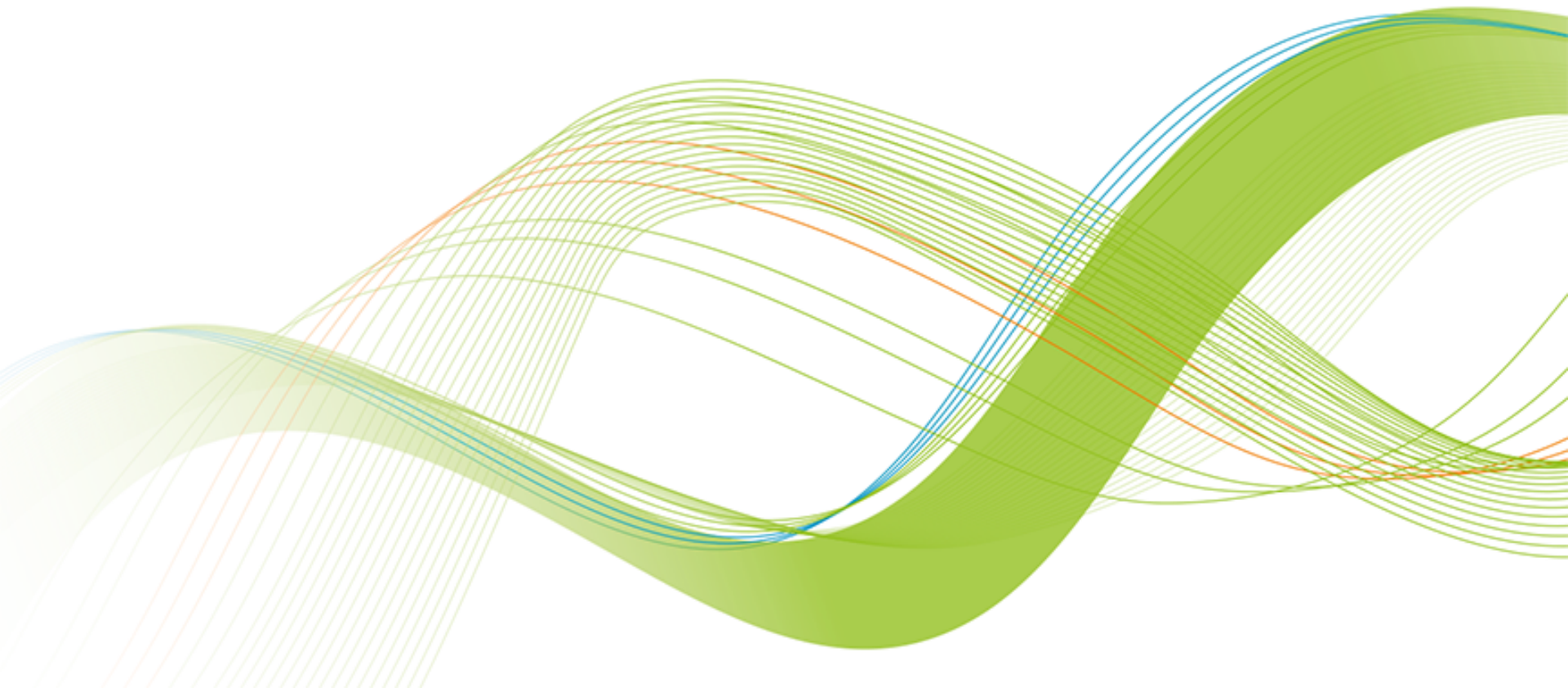
## D. Alignment and Data Analysis



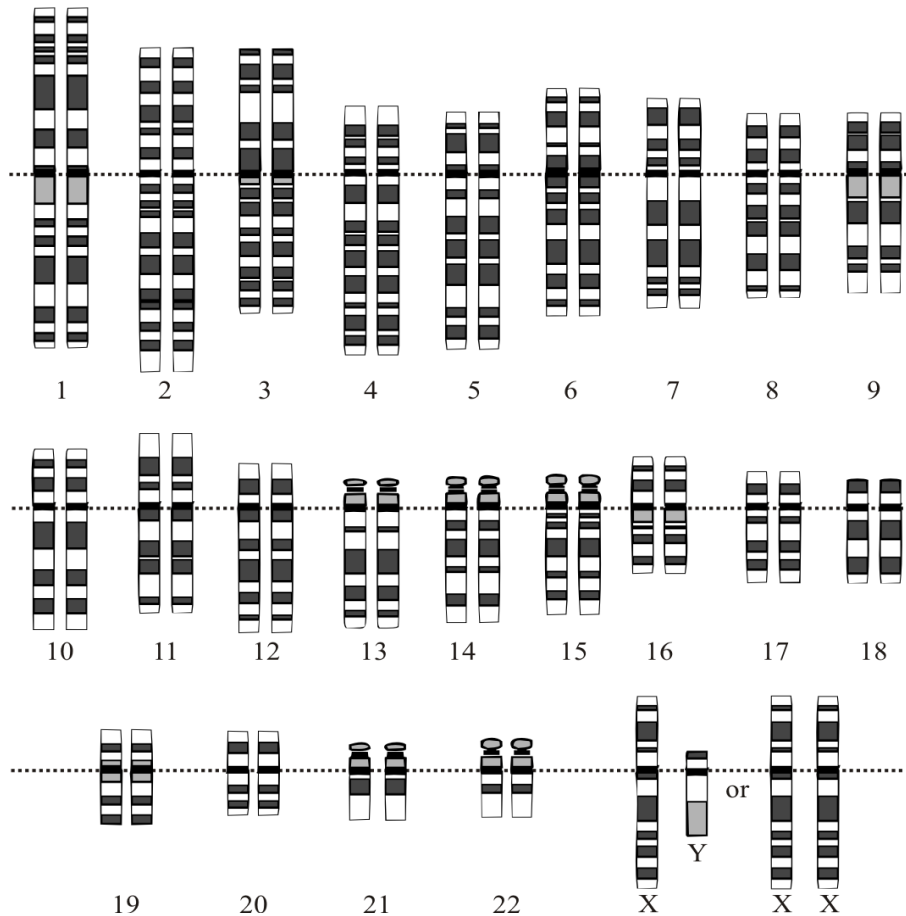
Reads are aligned to a reference sequence with bioinformatics software. After alignment, differences between the reference genome and the newly sequenced reads can be identified.

---

# The reference genome sequence



# Each chromosome...



# ...represented by a sequence

>chr1

```
GATCACAGGTCTATCACCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTTCGTCTG
GGGGGTGTGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCTATGTCGCAGTATCTGTC
TTTGATTCTGCCTCATTCTATTATTTATCGCACCTACGTTCAATATTACAGGCGAACATACCTAC
TAAAGTGTGTTAATTAATTAATGCTTGTAGGACATAATAATAACAATTGAATGTCTGCACAGCCGC
TTTCCACACAGACATCATAACAAAAAATTTCCACCAAACCCCCCCCCTCCCCCGCTTCTGGCCACA
GCACTTAAACACATCTCTGCCAAACCCCAAAAACAAAGAACCCTAACACCAGCCTAACCCAGATTTC
AAATTTTATCTTTAGGCGGTATGCACTTTTAAACAGTCACCCCCCACTAACACATTATTTTCCCCT
CCCCTCCCATACTACTAATCTCATCAATACAACCCCCGCCCATCCTACCCAGCACACACACACCCG
CTGCTAACCCCATAACCCGAACCAACCAAAACCCCAAAGACACCCCCCACAGTTTATGTAGCTTACC
TCCTCAAAGCAATACACTGAAAATGTTTAGACGGGCTCACATCACCCCATAAAACAATAGGTTTGG
TCCTAGCCTTTCTATTAGCTCTTAGTAAGATTACACATGCAAGCATCCCCGTTCCAGTGAGTTCAC
CCTCTAAATCACACGATCAAAAGAGGCGGTATGCACTTTTAAACAGTCACCCCCAGGCGGTATGCA
```

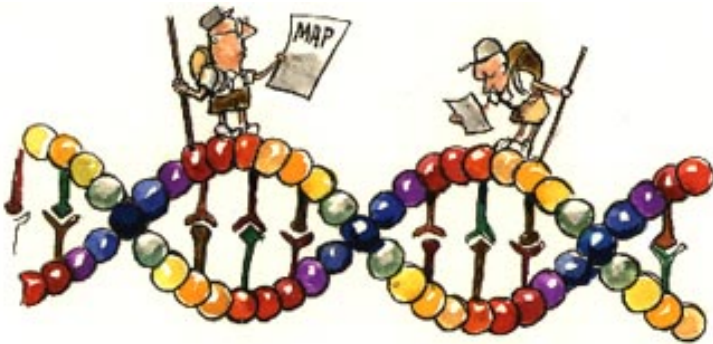
# The reference genome

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

In 2001: The International Human Genome Sequencing Consortium published the first draft of the human genome sequence. It contained 150,000 gaps.

HG19: 250 gaps

HG38 is the latest version of the human reference genome, but we will work with HG19.



# Keep track of the Reference version

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

- mapping
- visualizing
- variant calling
- annotation
- etc

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

The same reference sequence must be used in all downstream analyses.

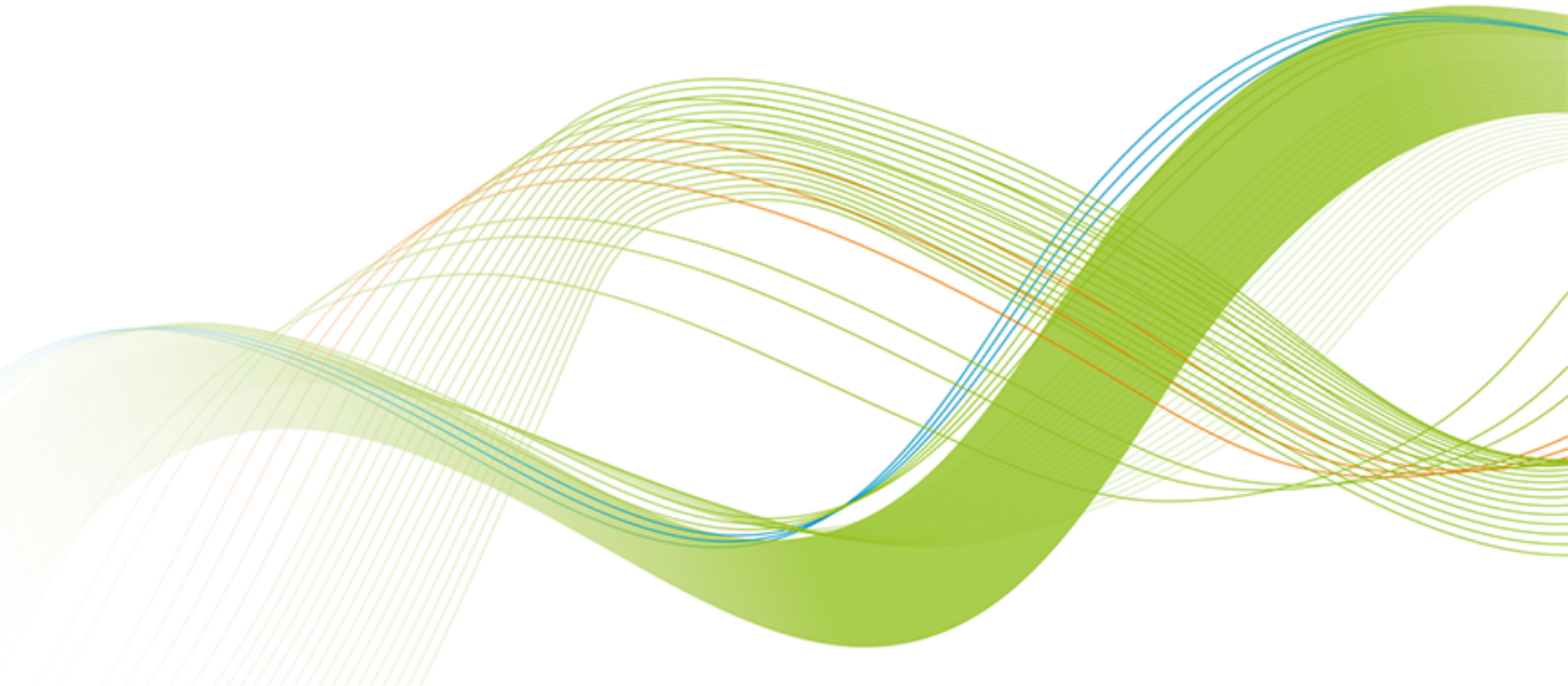


# File Indices

- Most large files we work with, such as the reference genome, need an index
- Allows efficient random access
- Different indices for different file-types
- Bwa index = Burrows-Wheeler transform of reference genome (several files)
- Needs index: fasta, bam vcf files

---

# Genetic variation

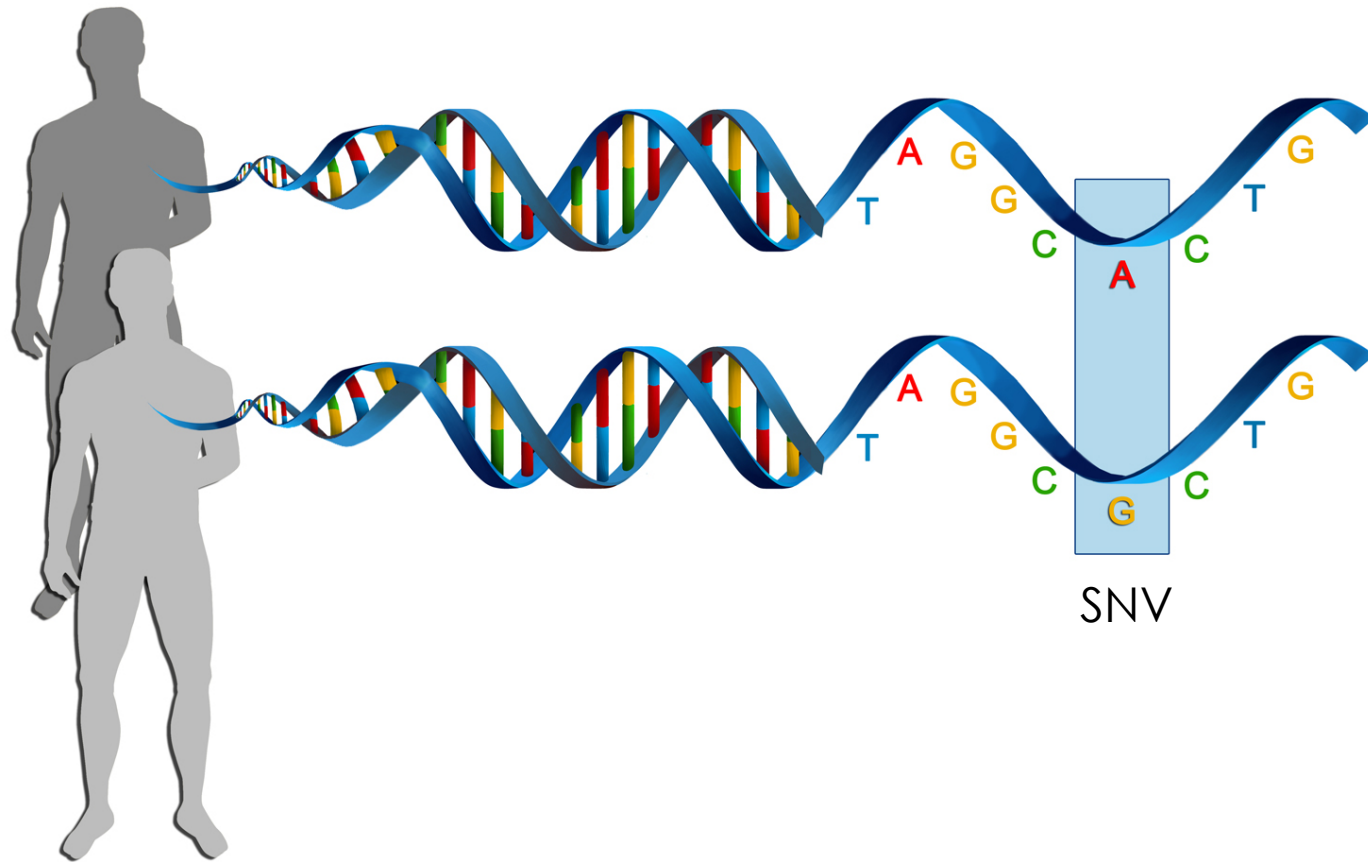


# Genetic Variation



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

# Single Nucleotide Variants (SNVs)



# Reference- and Alternative Alleles

GGCTTTTCCAACAGGTATATCTTCCCCGCTAGCTAGCTAGCTACTTCAAAT

**Reference allele** AGCTAGCTA

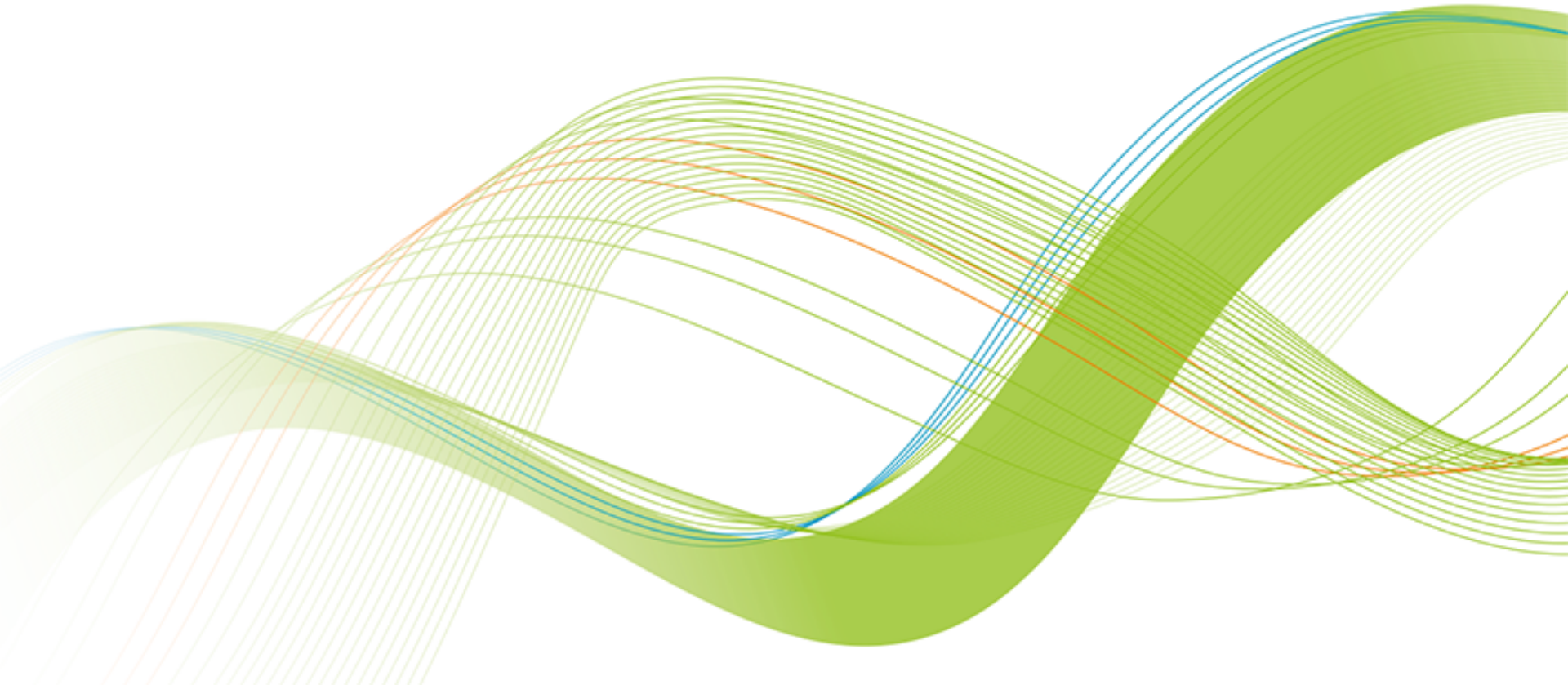
**Alternative allele** AGCTGGCTA

**Reference allele** = the allele in the reference genome

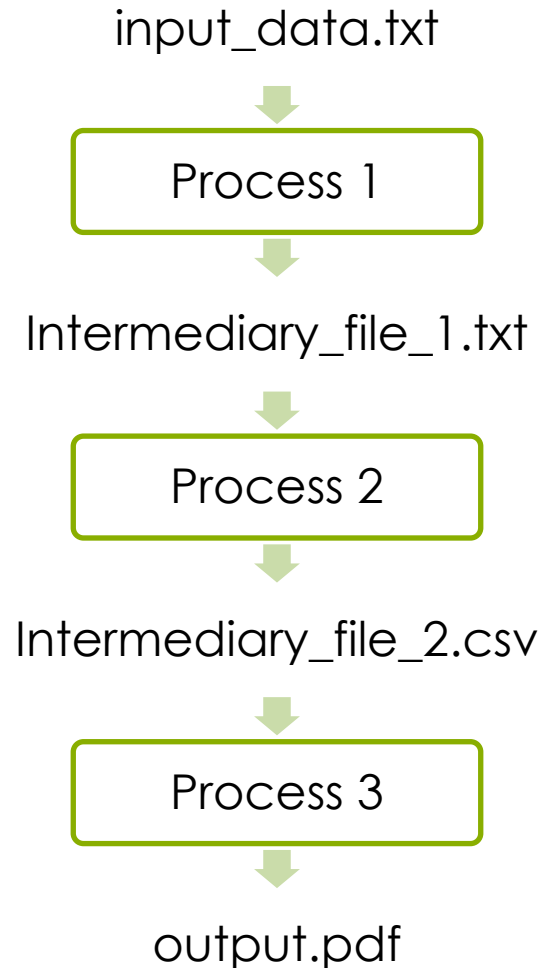
**Alternative allele** = the allele NOT in the reference genome

---

# Introduciton to workflows



# A bioinformatics workflow

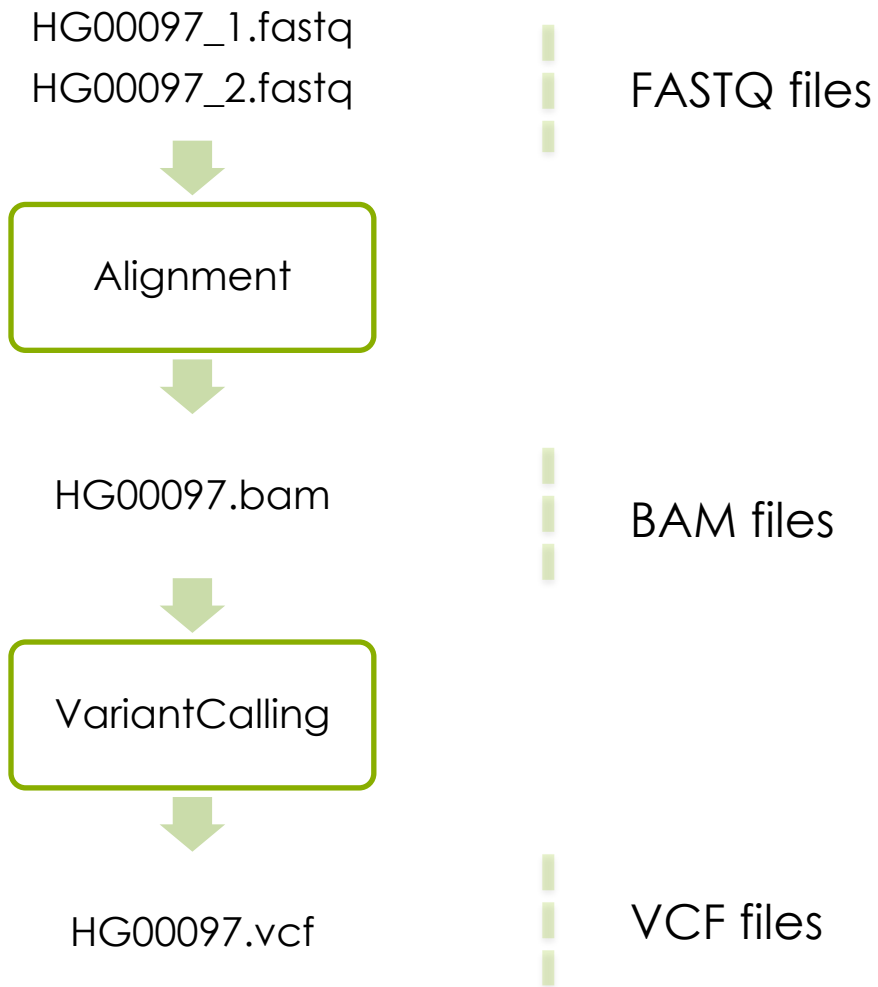


# Workflow conventions

- Each process has an input and output file(s)
- Create a new output file in each process – don't overwrite the input file
- Use informative file names
- Include information of the process in output file name

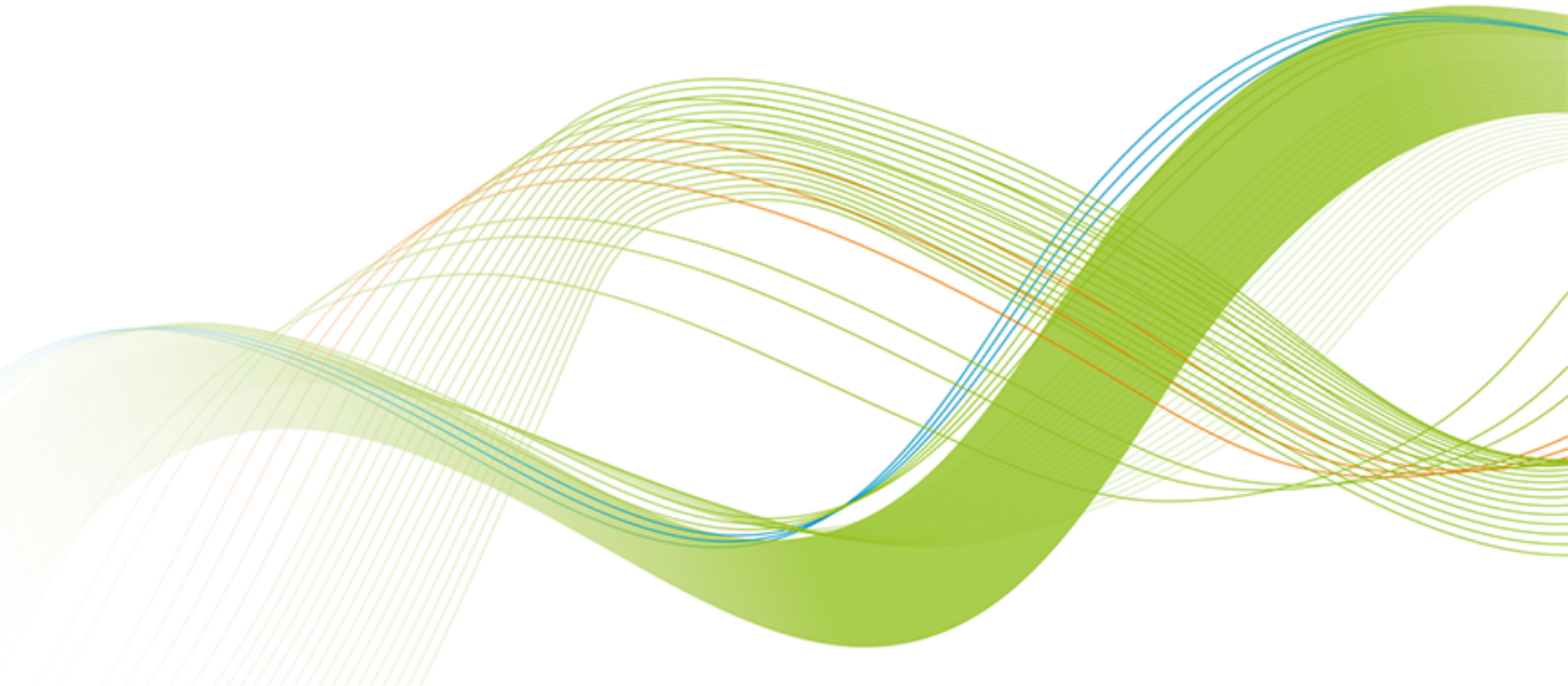


# Basic variant calling in one sample

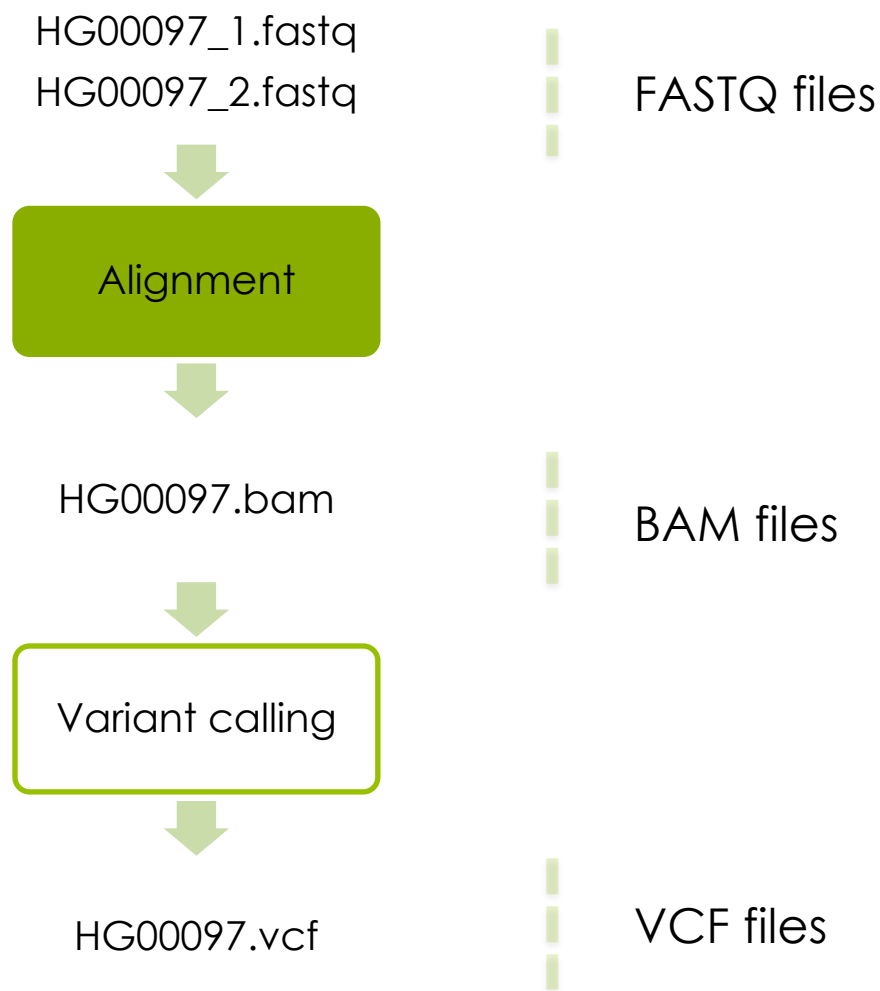


---

# Basic Variant Calling in one sample



# Basic variant calling in one sample



# Alignment

```
module load bwa
```

AACAGGTATATCTTCCCCGCTAGCTAGCTAGCTA**GCTAGCTAGCTAGCTACCCT**CTTCCTTAGGGACT  
GCTAGCTAGCTAGCTACCCT

---

# Burrows-Wheeler Aligner

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

## Burrows-Wheeler Aligner

### Introduction

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 sequences ranged from 70bp to 1Mbp. BWA-MEM and BWA-SW share similar features such as long-read support and split alignment, but BWA-MEM, which is the latest, is 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 Illumina reads.

### FAQ

#### How can I cite BWA?

The short read alignment component (bwa-short) has been published:

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: [20080505](#)]

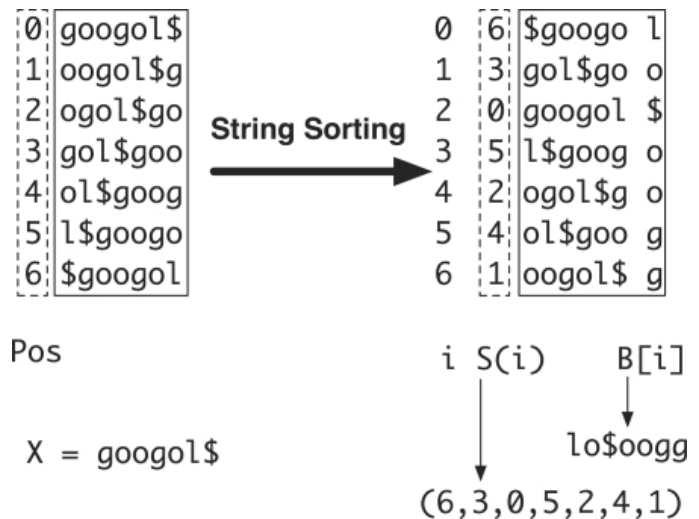
### BWA:

[SF project page](#)  
[SF download page](#)  
[Mailing list](#)  
[BWA manual page](#)  
[Repository](#)

### Links:

[SAMtools](#)  
[MAQ](#)

## Burrows-Wheeler transform of reference genome



# Alignment

module load bwa



# Output from mapping - Sam format

## 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<FFBFBFFFBFBFB
Read_001	147	2	3843625	0	101M	=	3843448	-278	TTATTTTCATTGAGCAGTGGT	FBBI7IIFIB<BBBB<BBFF
Read_002	163	2	4210055	0	101M	=	4210377	423	TGGTACCAAAACAGAGATAT	0IIFBFFFIIIFFIFFBFBF
Read_003	99	2	4210066	0	101M	=	4210317	352	CAGAGATATAGATCAATGGA	0IIFFFIFFFFIFIFIIF



Read name  
(usually more  
complicated)



Reference sequence name



Start position



Sequence



Quality

# Convert to Bam

Bam file is a binary representation of the Sam file



# Read groups

- Link *sample id, library prep, flowcell* and *sequencing run* to the reads.
- Good for error tracking!
- Often needed for variant calling
- Detailed description in tutorial or <https://gatkforums.broadinstitute.org/gatk/discussion/6472/read-groups>

**RGID** = *combination of the sample id and run id*

**RGLB** = Library prep

**RGPL** = Platform (for us ILLUMINA)

**RGPU** = Run identifier *usually barcode of flowcell*

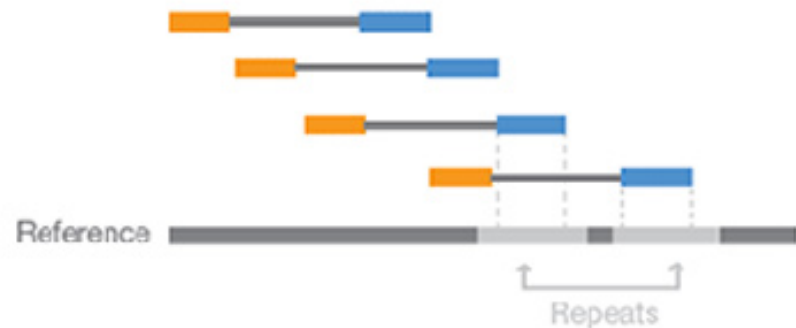
**RGSM** = Sample name

# Paired-End data

Paired-End Reads



Alignment to the Reference Sequence



Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

# Paired-end data

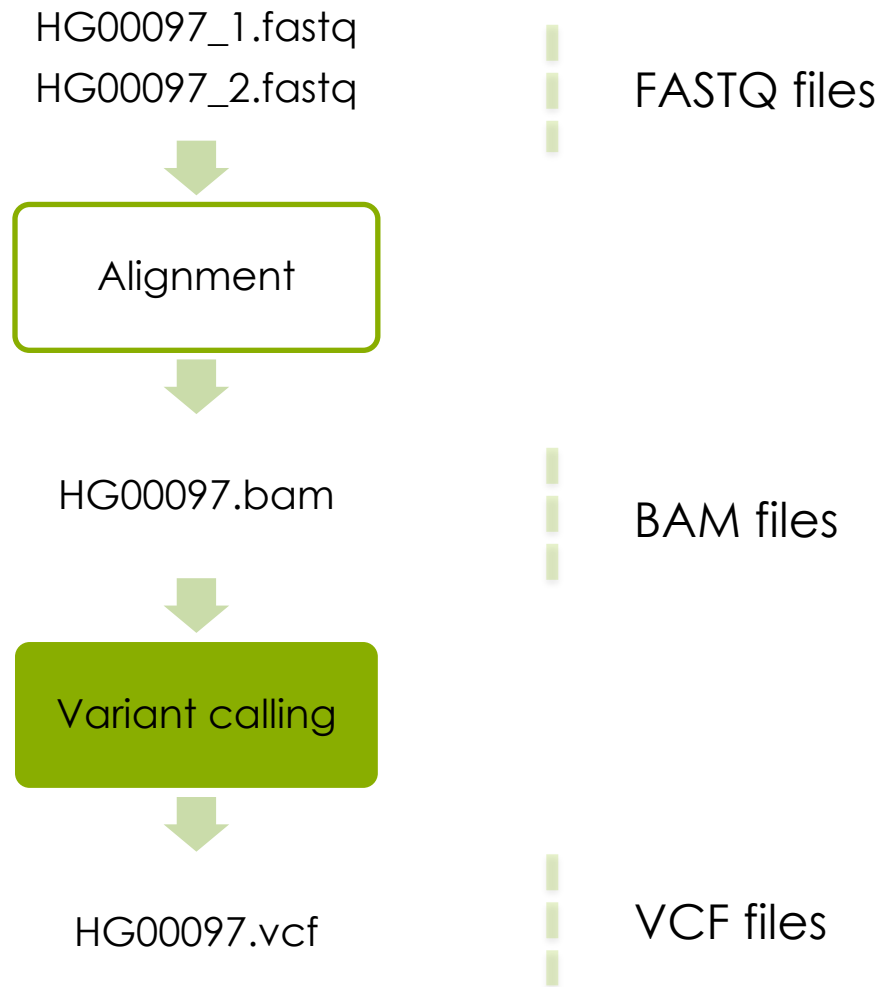
ID\_**R1**\_001.fastq

```
@HISEQ:100:C3MG8ACXX:5:1101:1160:2
197 1:N:0:ATCACG
CAGTTGCGATGAGAGCGTTGAGAAGTATAATAGG
AGTTAAACTGAGTAACAGGATAAGAAATAGTGAG
ATATGGAAACGTTGTGGTCTGAAAGAAGATGT
+
B@CFFFFFFHHHHHGJJJJJJJJJJJJFHHIIIIJJ
JIHGIIJJJJIIJIIJJJJIIJJJJJJIIIEIHHIJ
HGHHHHHDFFFEDDDDDDCDDDCDDDDDDDDCDC
```

ID\_**R2**\_001.fastq

```
@HISEQ:100:C3MG8ACXX:5:1101:1160:
2197 2:N:0:ATCACG
CTTCGTCCACTTTCATTATTCCTTTCATACATG
CTCTCCGGTTTAGGGTACTCTTGACCTGGCCTT
TTTTCAAGACGTCCCTGACTTGATCTTGAAACG
+
CCCFFFFFFFHHHHHJJJJIIJJJJJJJJJJJJJJ
JJJJJJJJIIJIIJGIJHBGHHIIIIJIIJJJJJJ
JJJHFFFFFFFDDDDDDDDDDDDDDDEDCDDDD
```

# Basic variant calling in one sample



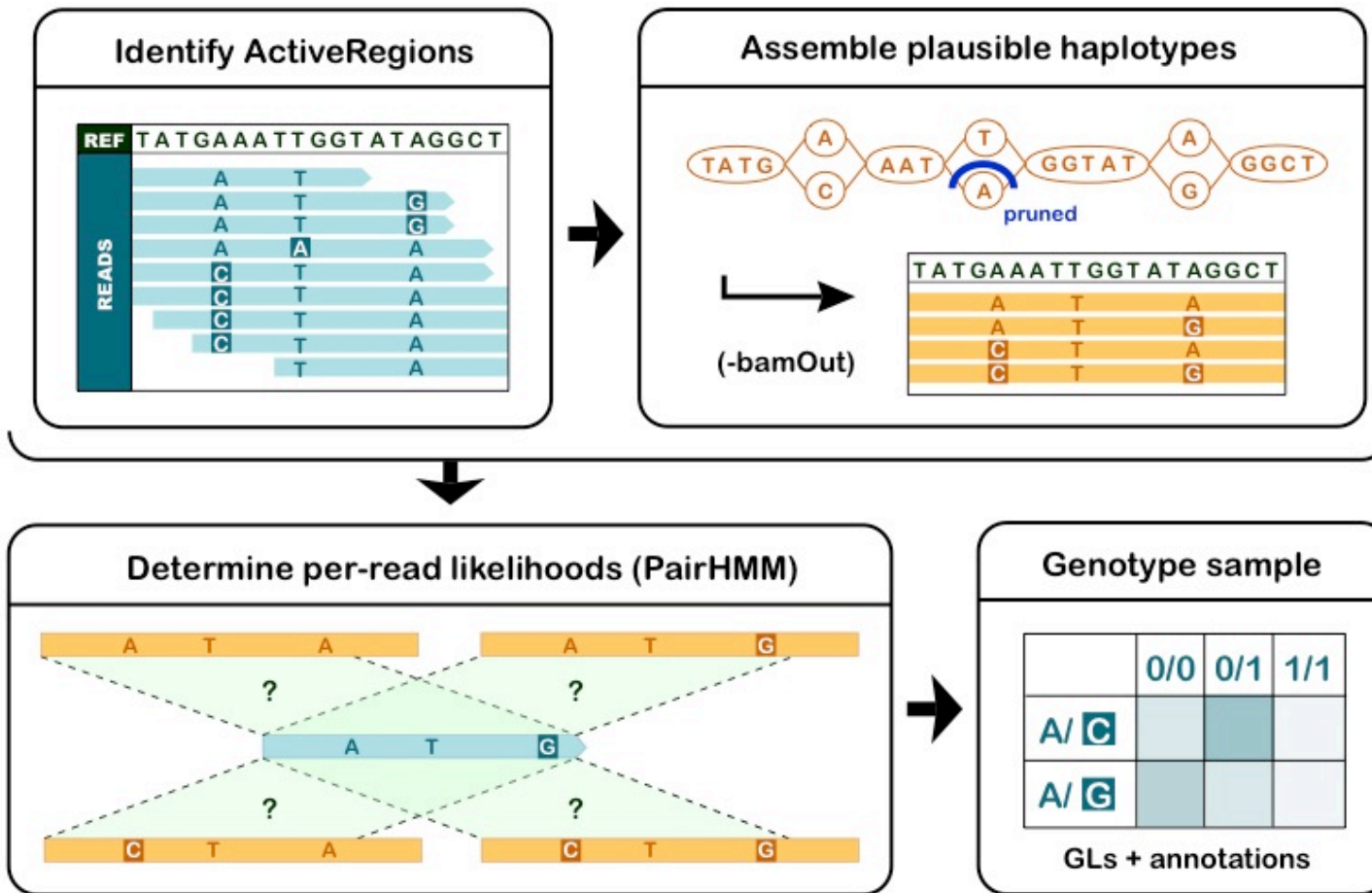
# Detecting variants in reads

Reference: ...GTGCGTAGACTGCTAGATCGAAGA...

Sample: ...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...  
...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...  
...GTGCGTAGACTGCTAGATCGAAGA...  
...GTGCGTAGACTGCTAGATCGAAGA...  
...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...  
...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...  
...GTGCGTAGACTGCTAGATCGAAGA...  
...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...  
...GTGCGTAGACTGCTAGATCGAAGA...  
...GTGCGTAGACTG<sup>A</sup>TAGATCGAAGA...

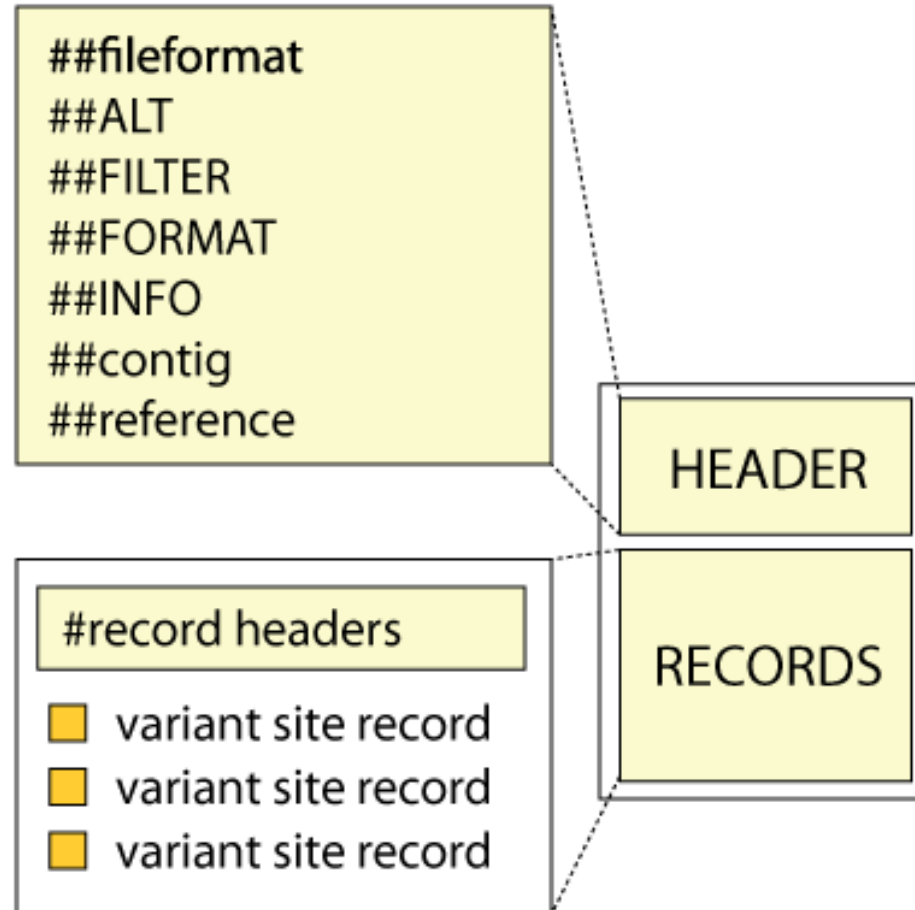
# Variant Calling

## HaplotypeCaller



For more info: <https://www.youtube.com/watch?v=NQHGkVGICpY>

# Variant Call Format (VCF)



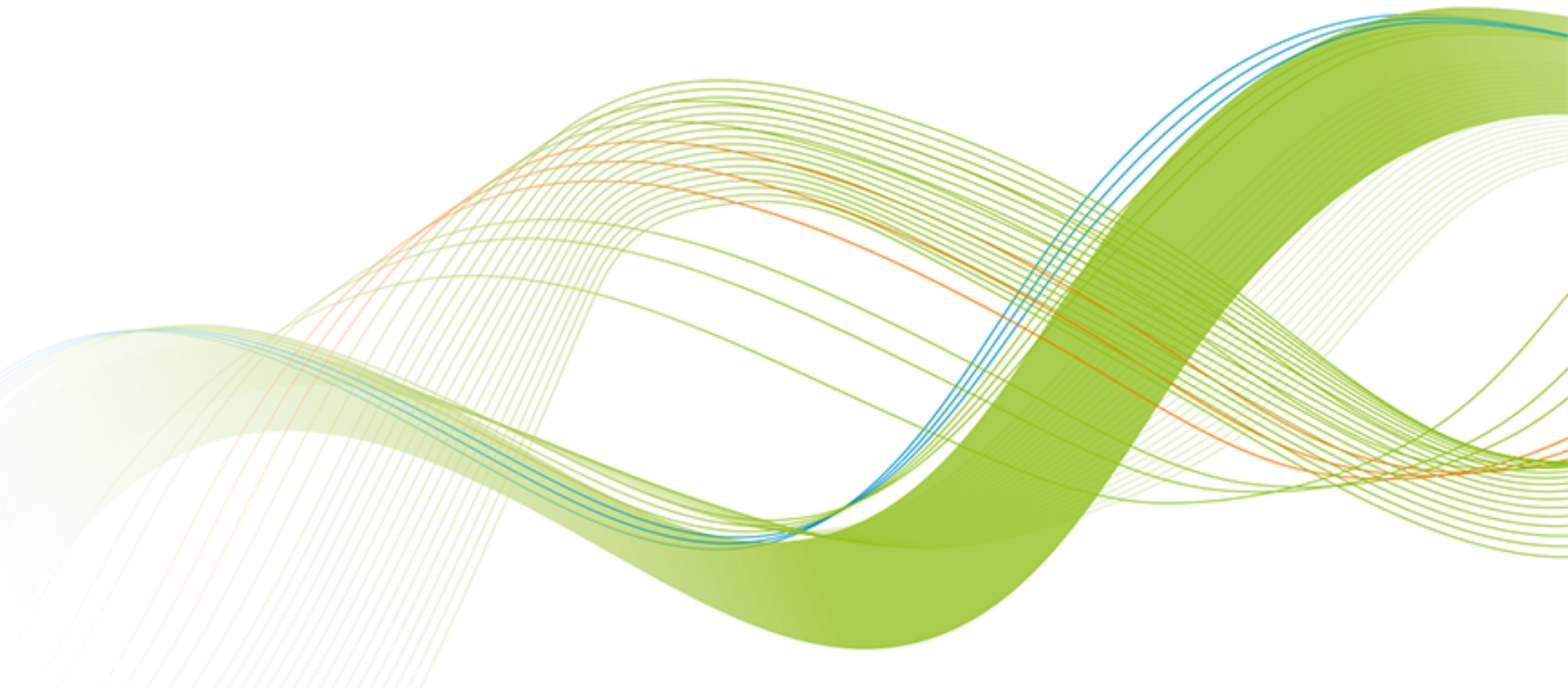
# Variant Call Format (VCF)

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens"...
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001
20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP 0|0:48:1
20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP 0|0:49:3
20 1230237 . T . 47 PASS NS=3;DP=13;AA=T GT:GQ:DP 0|0:54:7
20 1234567 microsat1 GTC G,GTCT 50 PASS NS=3;DP=9;AA=G GT:GQ:DP 0|1:35:4
```

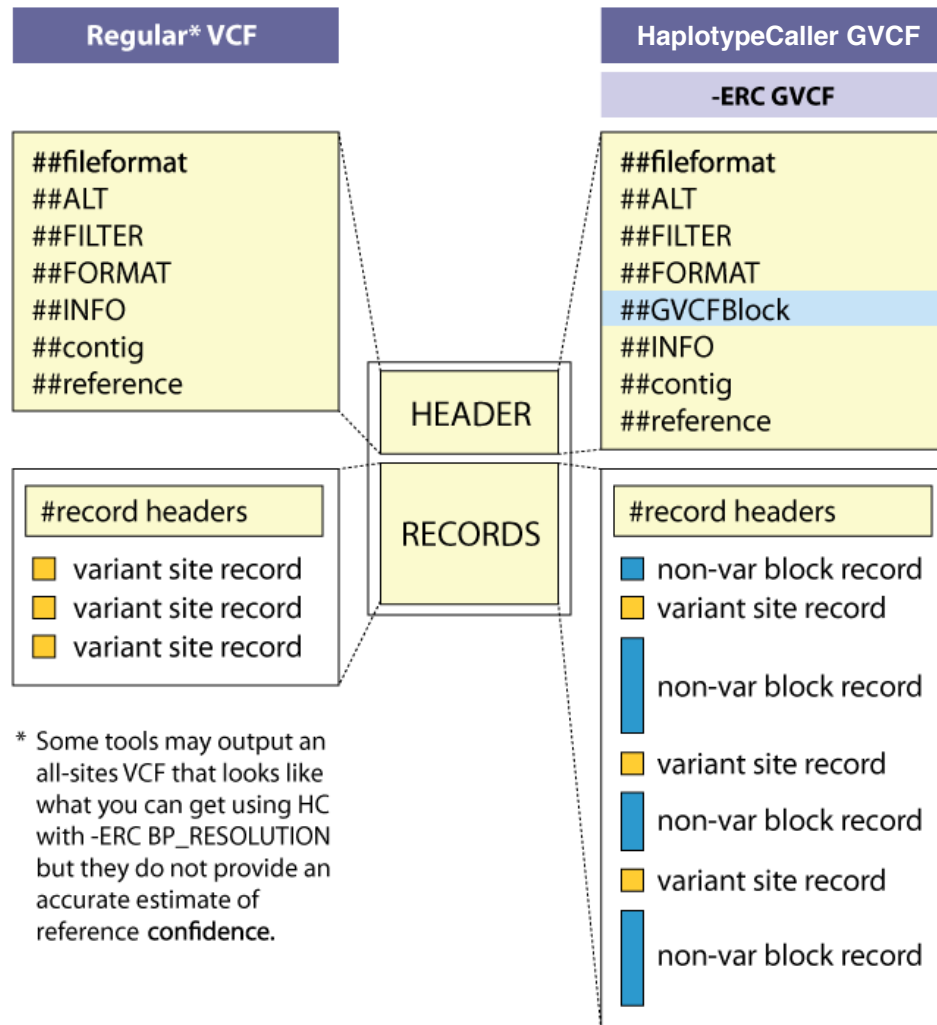


---

# Variant calling in cohort

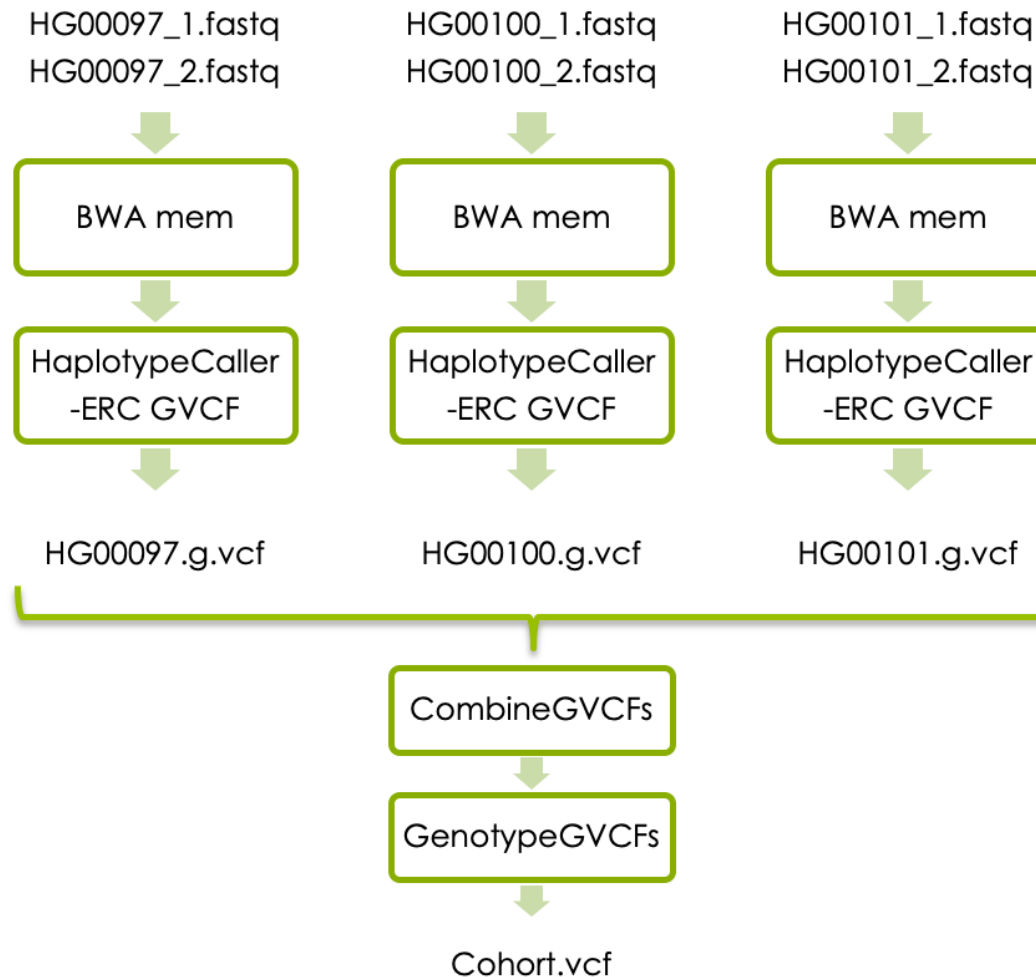


# GVCF Files are valid VCFs with extra information



- 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)

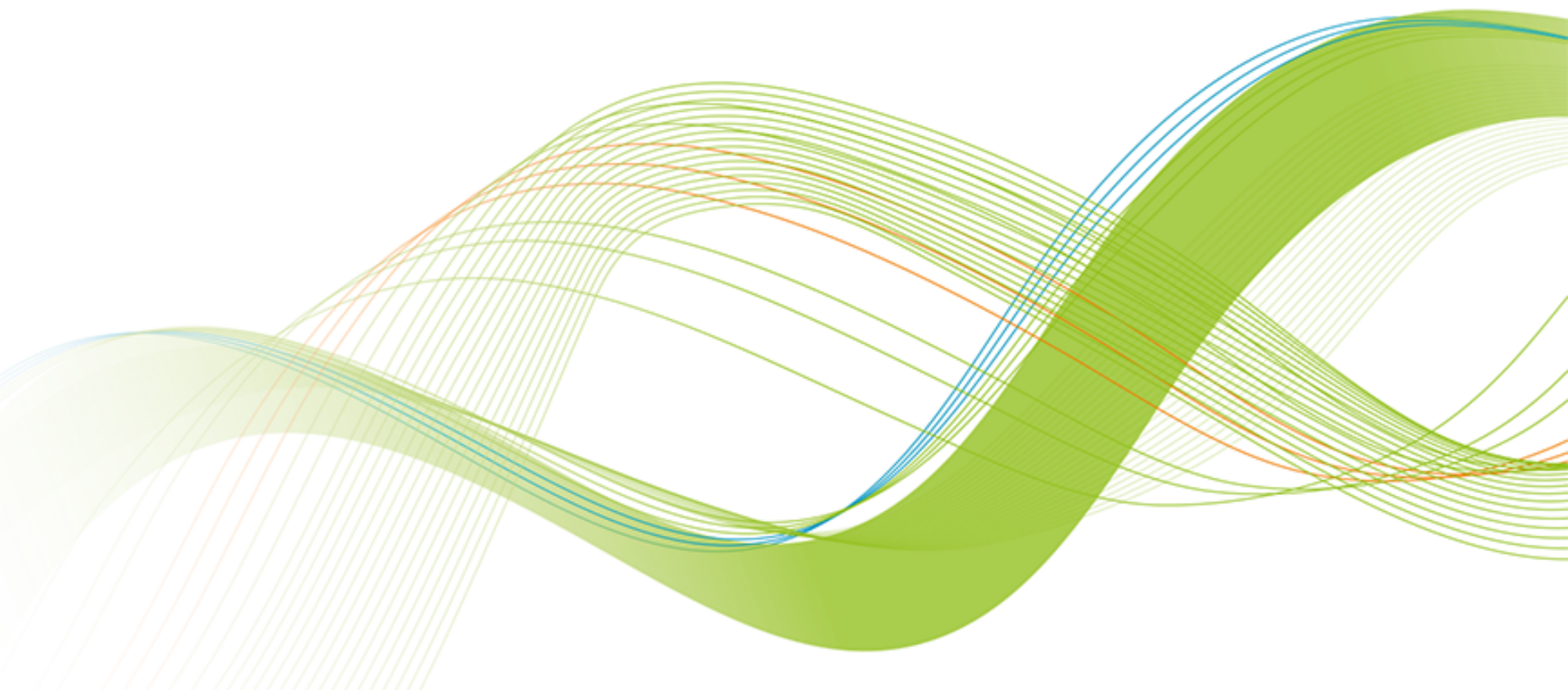
```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens"...
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
```



#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	NA000001	NA000002	NA000003
20	14370	rs6054257	G	A	29	PASS	NS=3;DP=14;AF=0.5;DB;H2	GT:GQ:DP	0 0:48:1	1 0:48:8	1 1:43:5
20	17330	.	T	A	3	q10	NS=3;DP=11;AF=0.017	GT:GQ:DP	0 0:49:3	0 1:3:5	0 0:41:3
20	1230237	.	T	.	47	PASS	NS=3;DP=13;AA=T	GT:GQ:DP	0 0:54:7	0 0:48:4	0 0:61:2
20	1234567	microsat1	GTC	G,GTCT	50	PASS	NS=3;DP=9;AA=G	GT:GQ:DP	0 1:35:4	0 2:17:2	1 1:40:3

---

# GATK's best practice germline short variant discovery



<https://gatk.broadinstitute.org>



[User Guide](#)

[Tool Index](#)

[Blog](#)

[Forum](#)

[DRAGEN-GATK](#)

[Events](#)

[Download GATK4](#)

[Sign in](#)

# Genome Analysis Toolkit

Variant Discovery in High-Throughput Sequencing Data



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

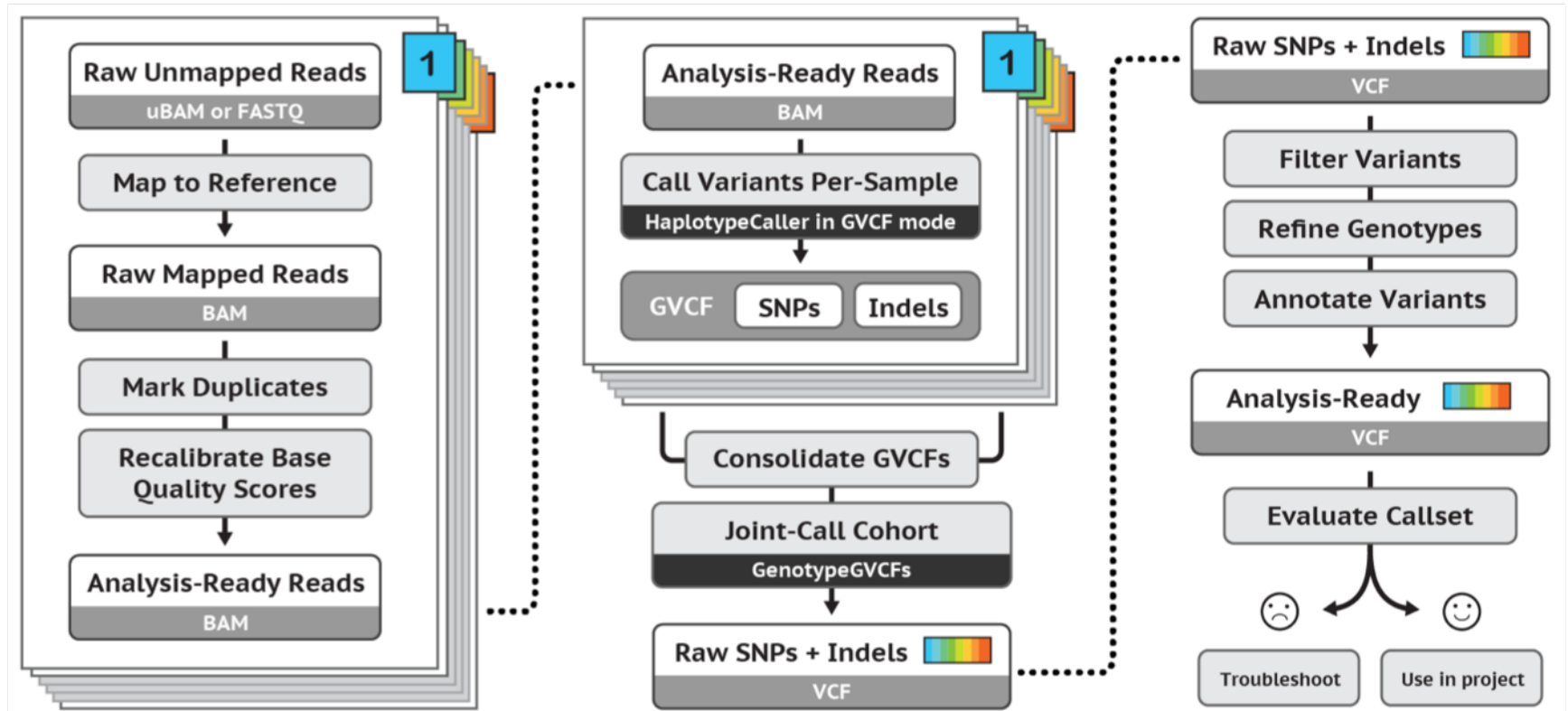


## Run on Cloud



## Run on HPC

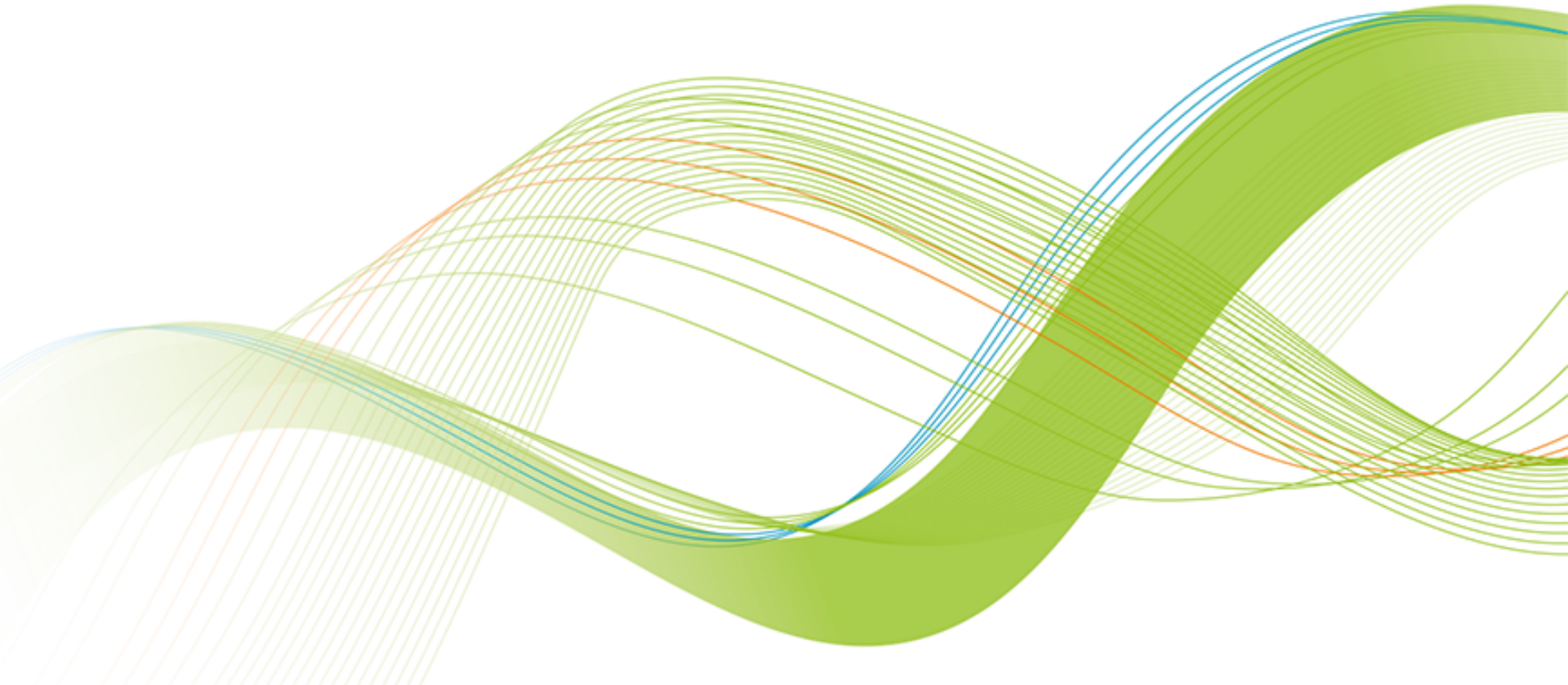
# GATK best practices workflow for variant discovery





---

# Introduction to computational exercise





# 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

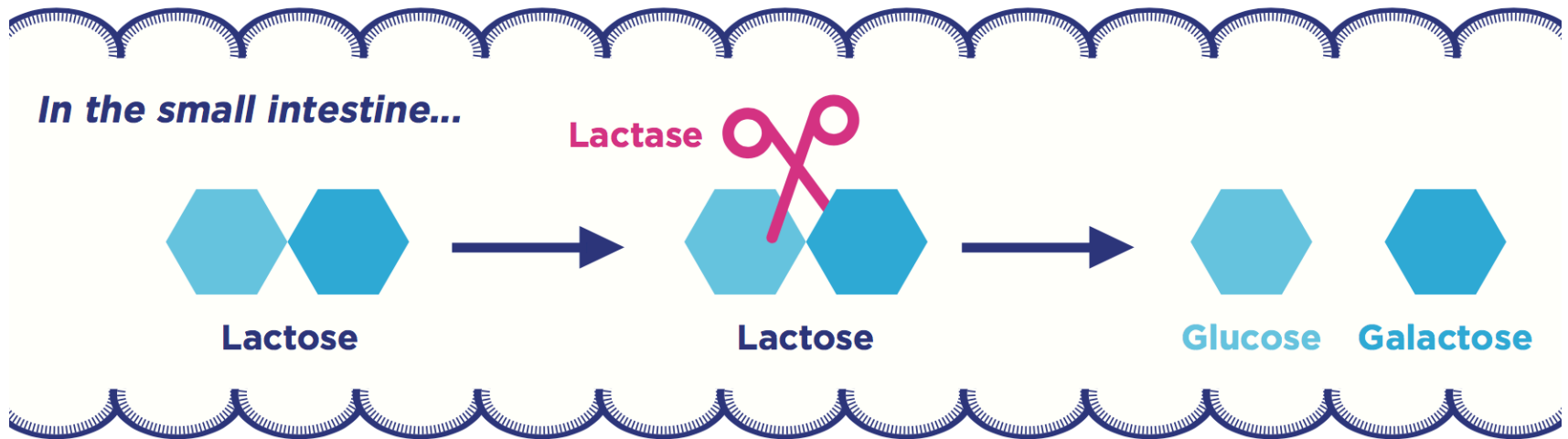


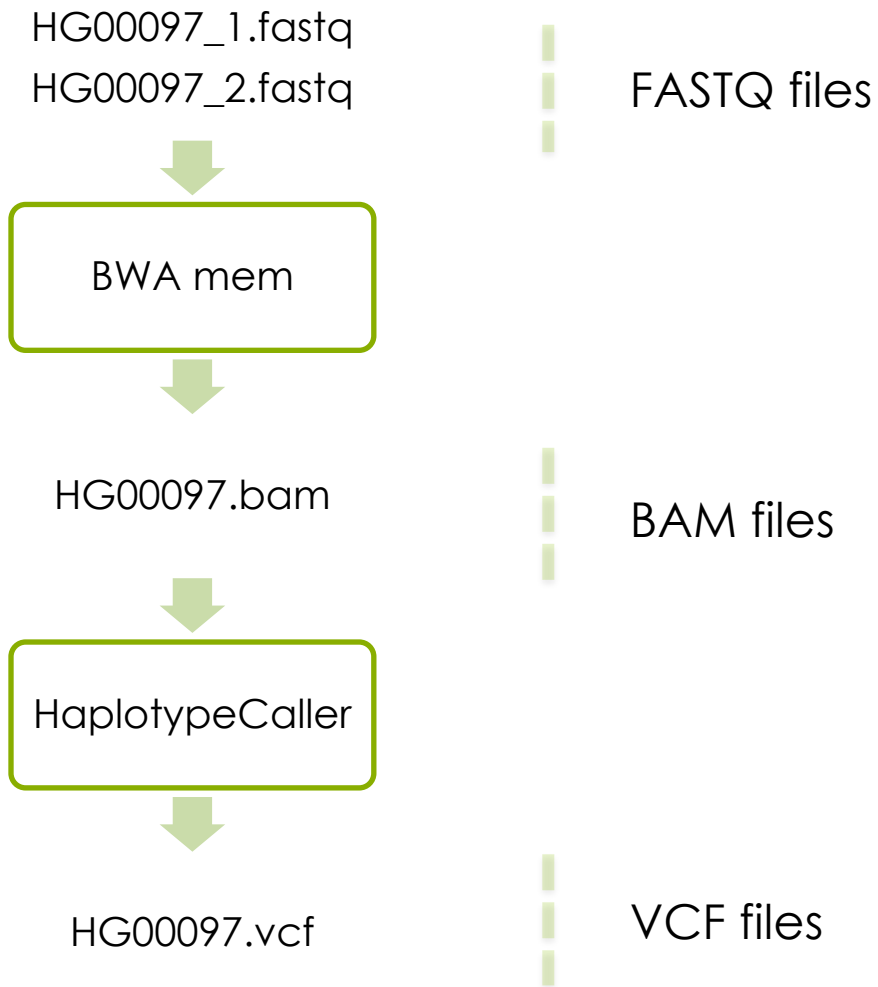
Figure 2. Lactose digestion in the intestine.

- All mammals produce lactase as infants
- Some human produce lactase in adulthood
- The *LCT* gene on chromosome 2 encodes lactase
- Genetic variation upstream of the *LCT* gene cause the lactase persistent phenotype

**part one:**

**variant calling in one sample**

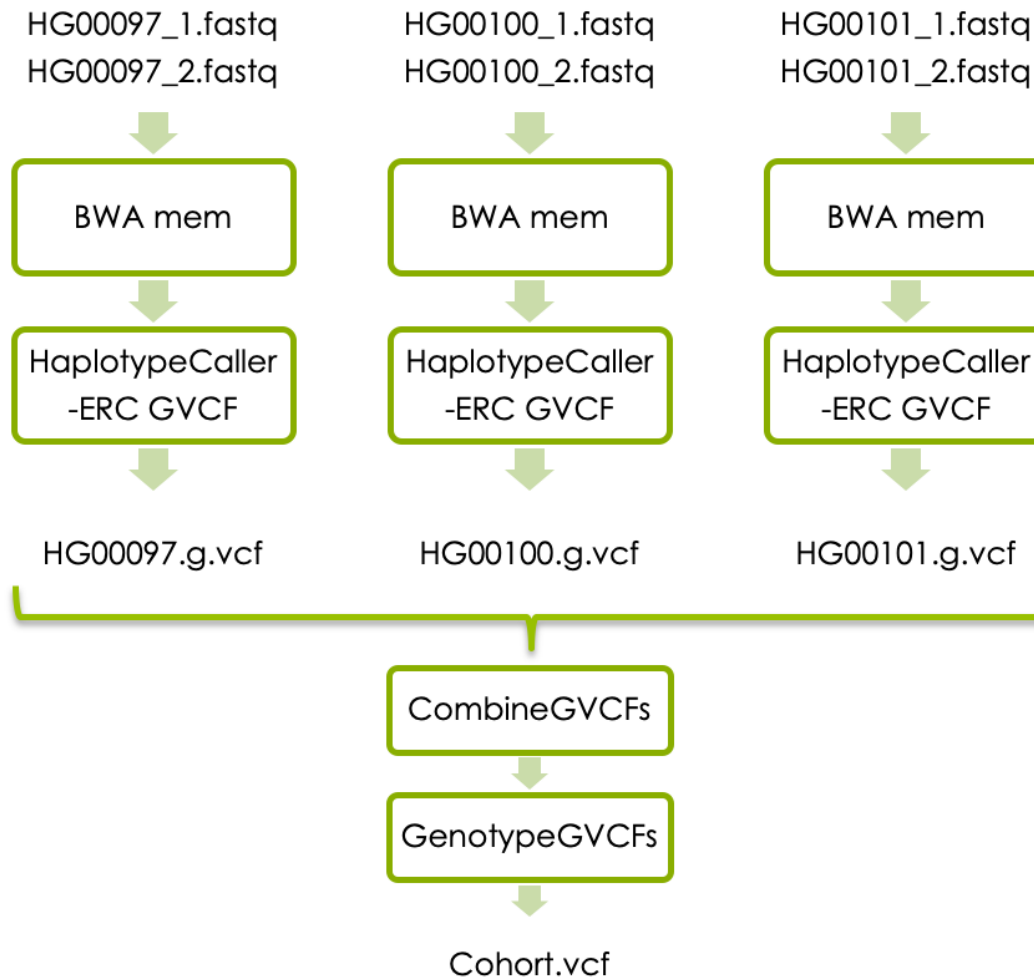
# Basic variant calling in one sample



**Part two (if you have time):**

**variant calling in cohort**

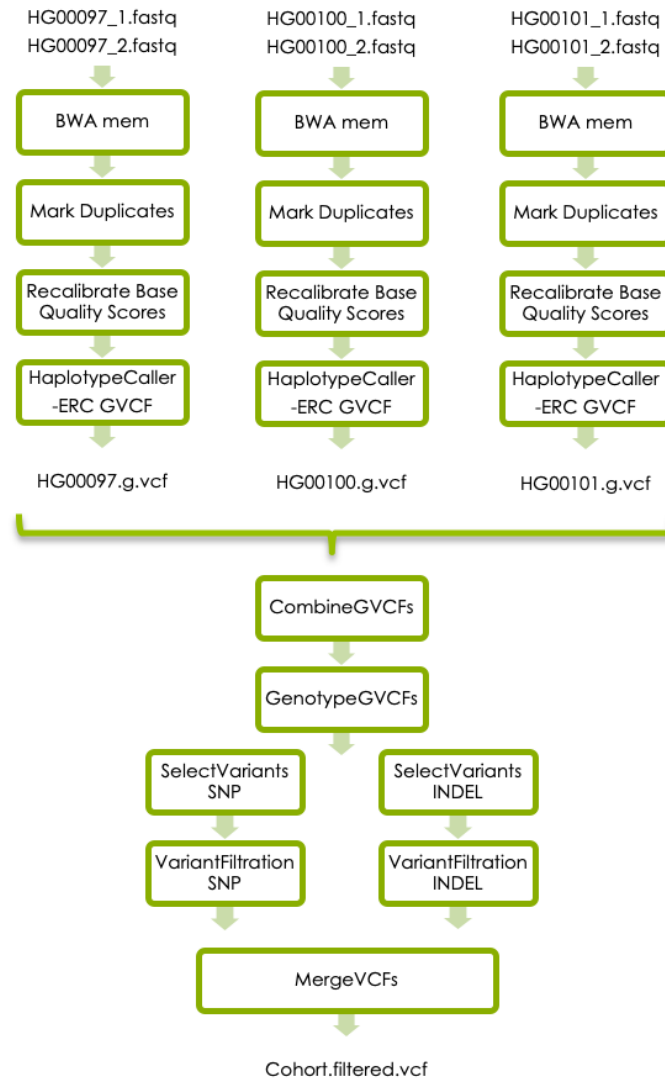
# Joint variant calling workflow



**Part three (if you have time):**

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

# GATK's best practises



First look at video  
about this linked  
from schedule!



**Questions?**