Topic 6: Variant calling using GATK

Paul Battlay
Paul.Battlay@monash.edu

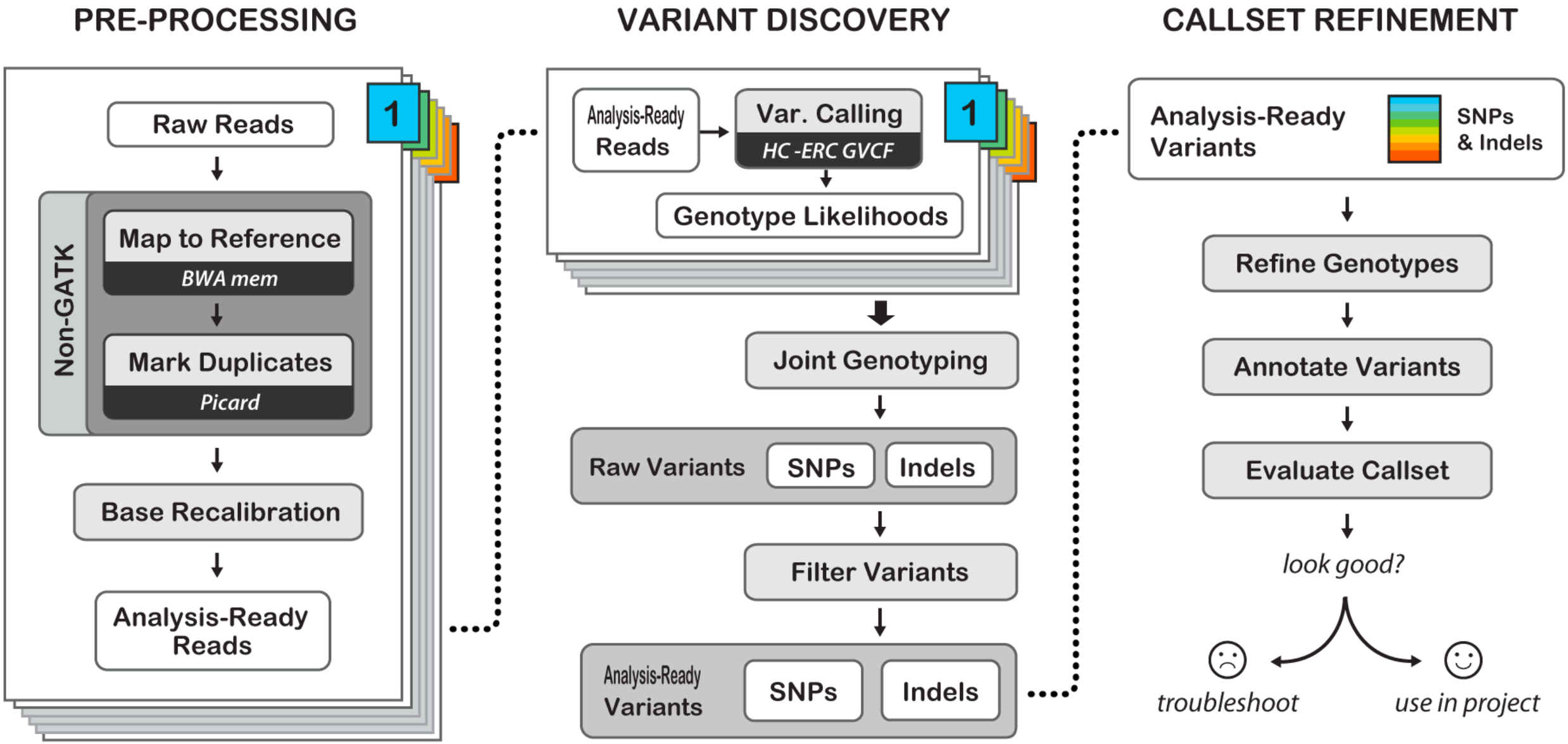
Variant calling

- You have aligned sequencing reads from individual samples to a reference genome
- We now want to identify (real) genetic sequence variation between each sample and the reference

Learning Goals

- Define the steps involved in variant calling and what they do
- Understand joint genotyping haplotype-based variant calling
- Understand the reason for variant filtering and the ways it can be done

GATK Best Practises: ‘Gold standard’ variant calling



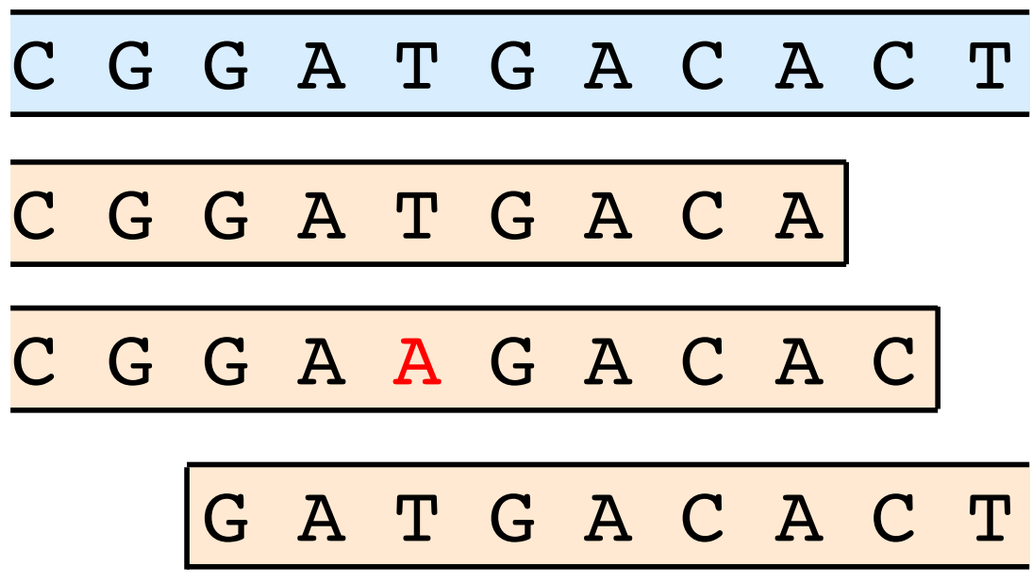
Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

GATK Best Practises: ‘Gold standard’ variant calling

- Dataset availability
  - ‘Truth sets’ of known variants
- Computational time and resources
- Project goals

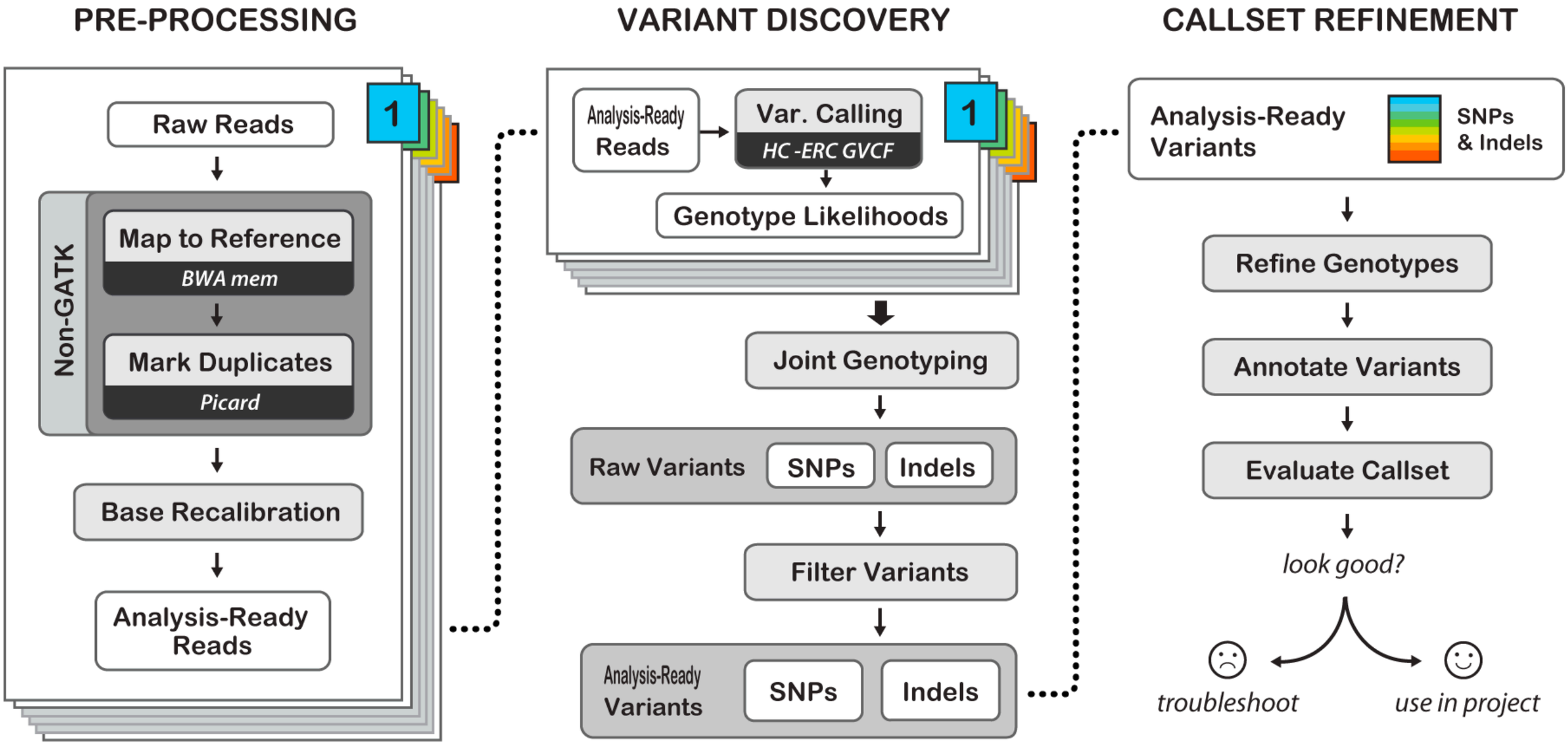
Genetic variants or errors?

- Sequencing error?
- Alignment error?
- True genetic variant?



Real mutations are hidden in the noise

Aligned BAMs are not ready for variant calling



Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

Picard MarkDuplicates

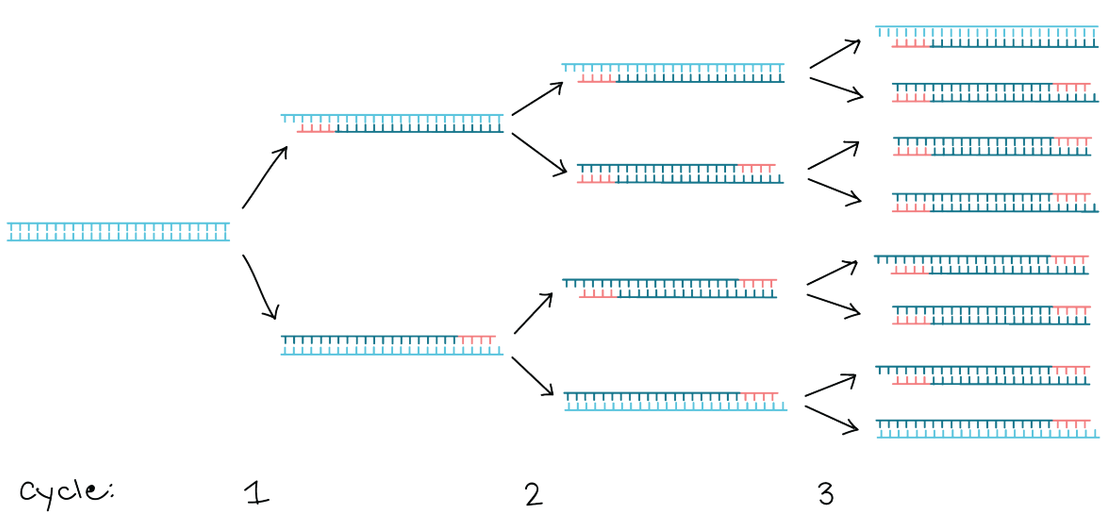
- Picard- a suite of utilities developed alongside GATK

- MarkDuplicates utility marks duplicate reads in your BAM file

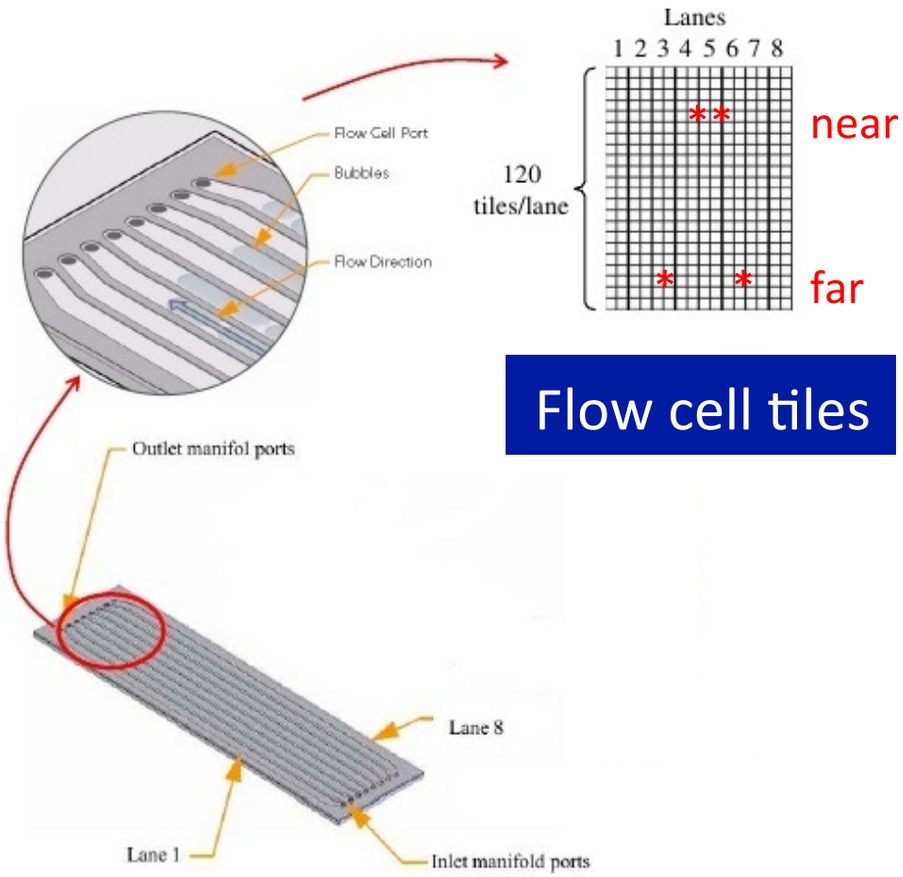
- Marked duplicates are excluded in downstream analyses

Where does the duplication come from?

- **PCR DUPLICATES**
  - **Increases with cycles**
- **OPTICAL DUPLICATES**
  - **Are nearby clusters on a flow cell lane**



https://www.khanacademy.org/science/biology/biotech-dna-technology/dna-sequencing-pcr-electrophoresis/a/polymerase-chain-reaction-pcr

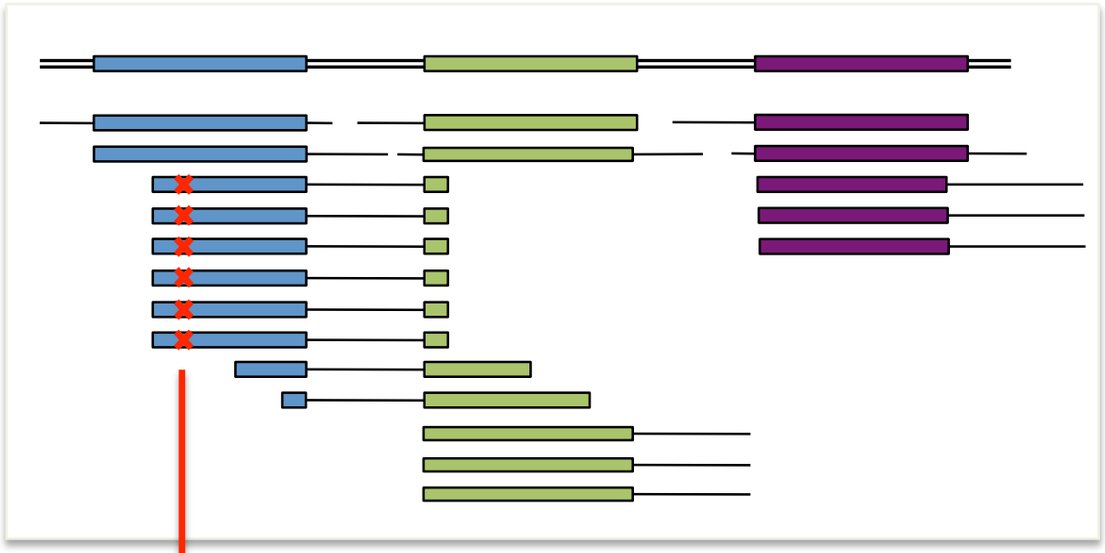


Flow cell tiles

http://www.slideshare.net/jandot/next-generation-sequencing-course-part-2-sequence-mapping
http://www.slideshare.net/cosentia/illumina-gaiix-for-high-throughput-sequencing

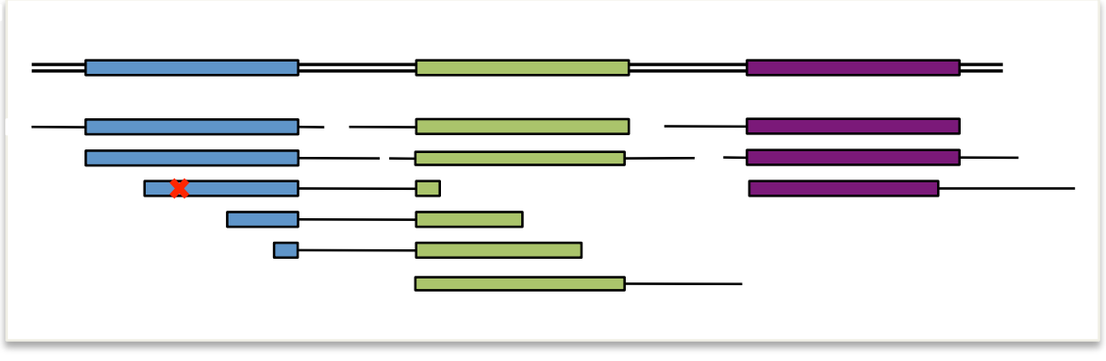
The reason why duplicates are bad

✖ = sequencing error propagated in duplicates



FP variant call (bad)

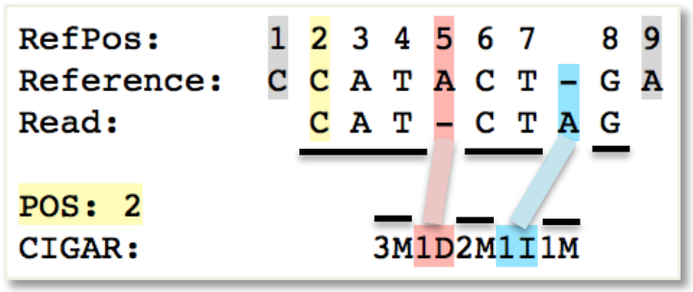
After marking duplicates, the GATK will only see :



… and thus be more likely to make the right call

Easy to identify: duplicate reads have the same starting position and same CIGAR string

CIGAR string- compressed information in a BAM file about where and how a read maps to the reference



Easy to identify: duplicate reads have the same starting position and same CIGAR string

Why wouldn't we do this for GBS?

Base Quality Score Recalibration (BQSR)

- Base quality sores reflect the confidence in each base emitted by the sequencing machine

- Variant calling relies on base qualities

- There are systematic biases in quality scores

  - Machine cycle

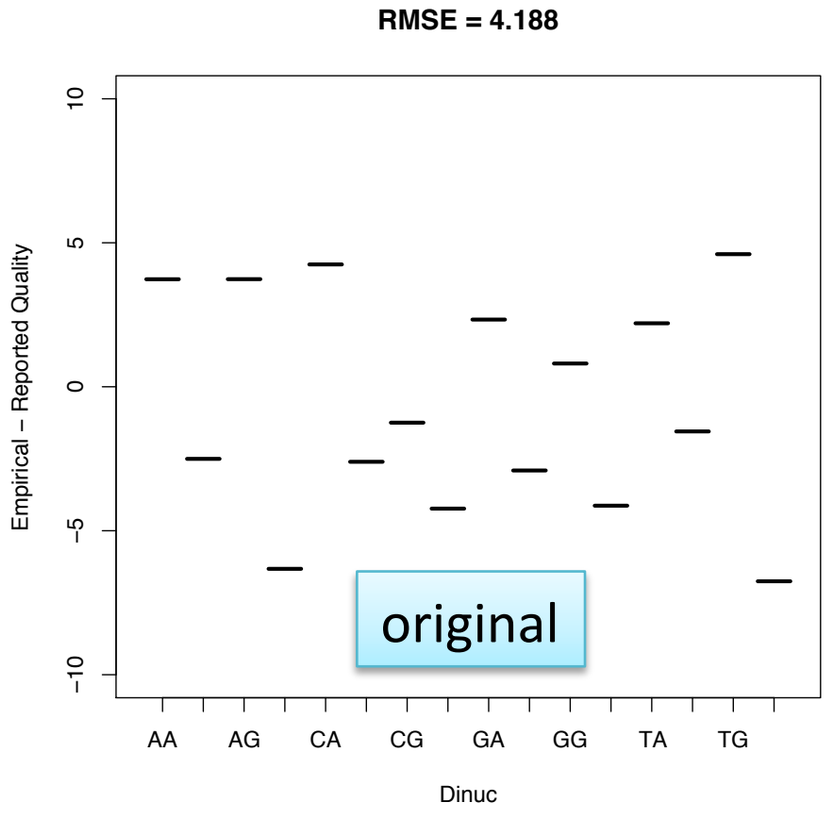
  - Nucleotide context (previous base)

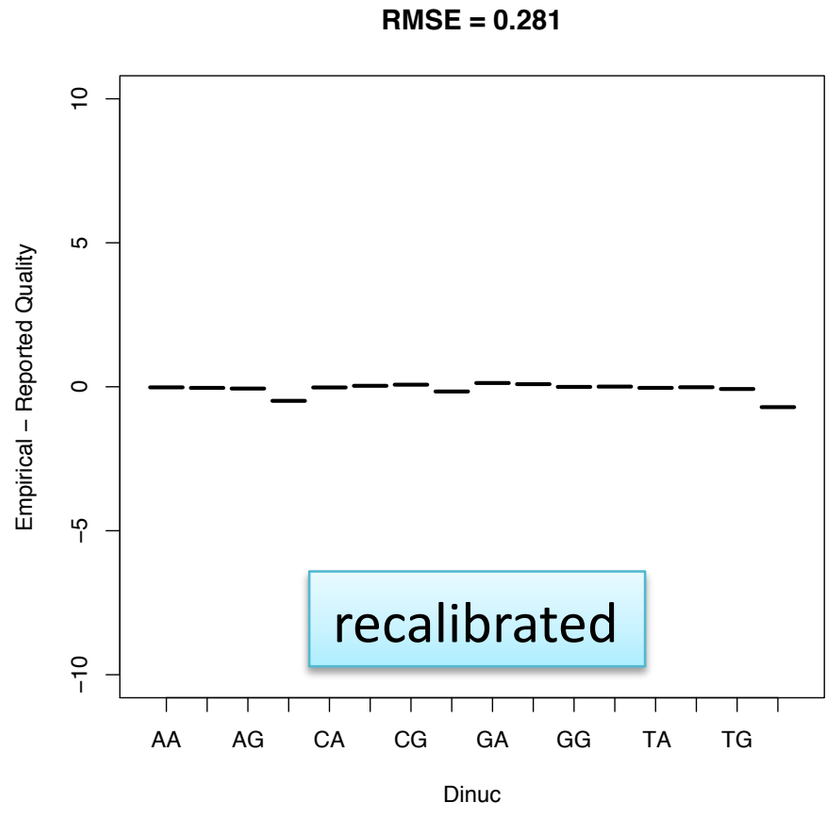
  - Read group (library; lane)

Base Quality Score Recalibration (BQSR)

- Machine learning to correct for biases in base quality scores

Example of bias: qualities reported depending on nucleotide context

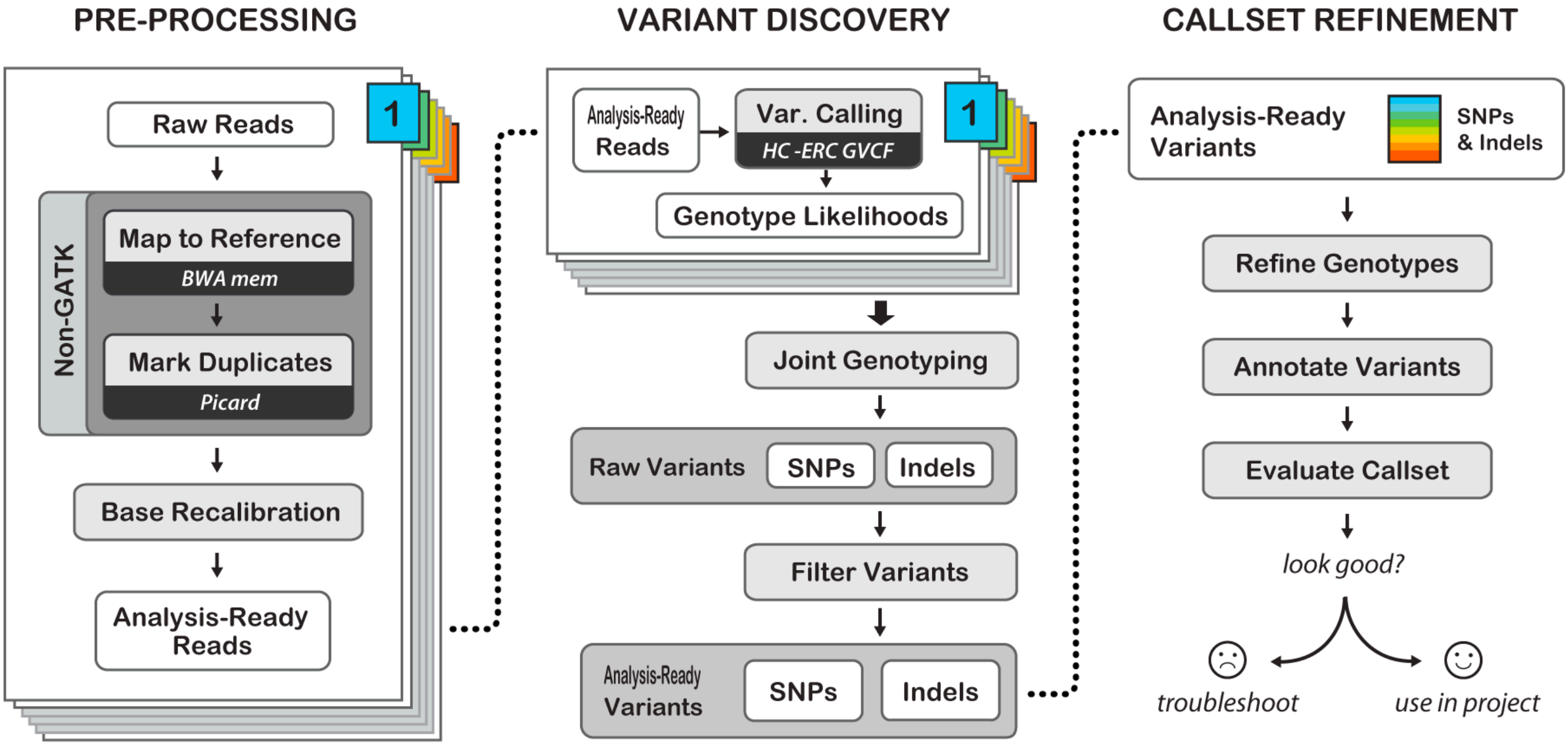




Base Quality Score Recalibration (BQSR)

- Requires a set of known variants
- Bootstrap from your data
  - Call variants without BQSR
  - Stringent filter for high-quality variants
  - Use as known variants with BQSR in second round of variant calling

GATK Best Practises: ‘Gold standard’ variant calling



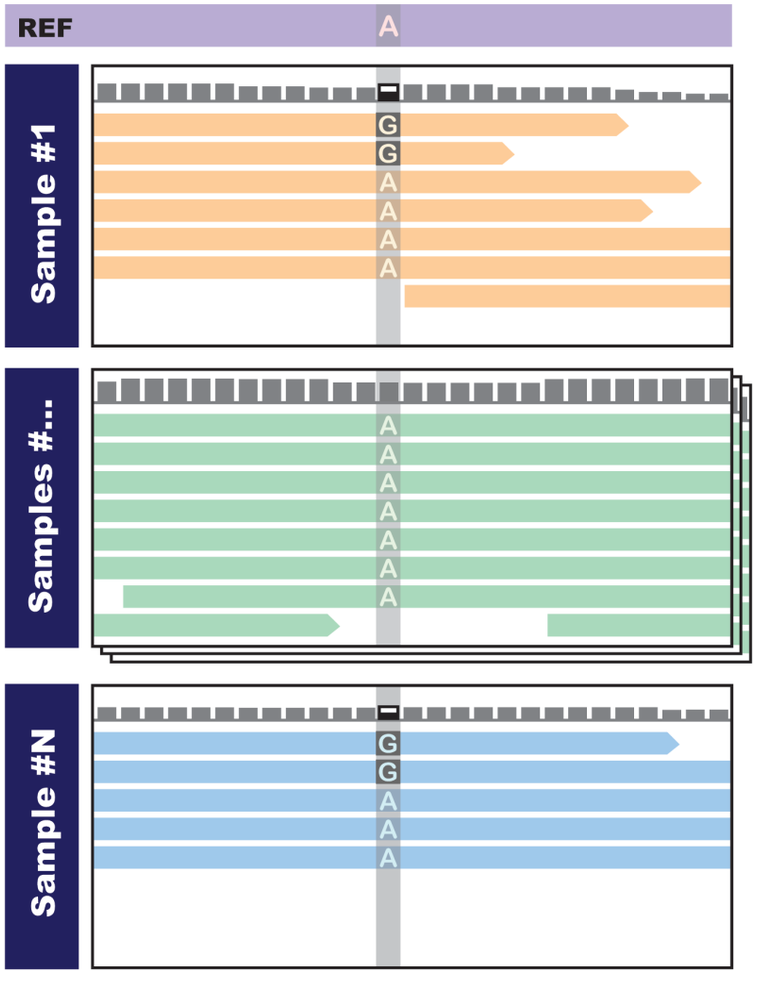
Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

Joint genotyping

- Call variants across all cohort/population samples at the same time

- Increased support for variants when similar patterns are observed in multiple samples

Joint discovery empowers discovery at difficult sites

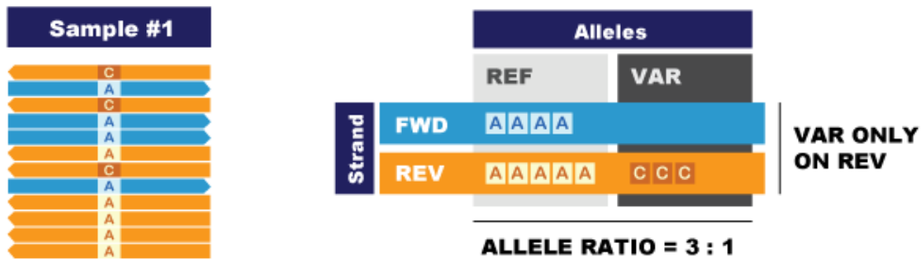


- If we analyze Sample #1 or Sample #N alone we are not confident that the variant is real

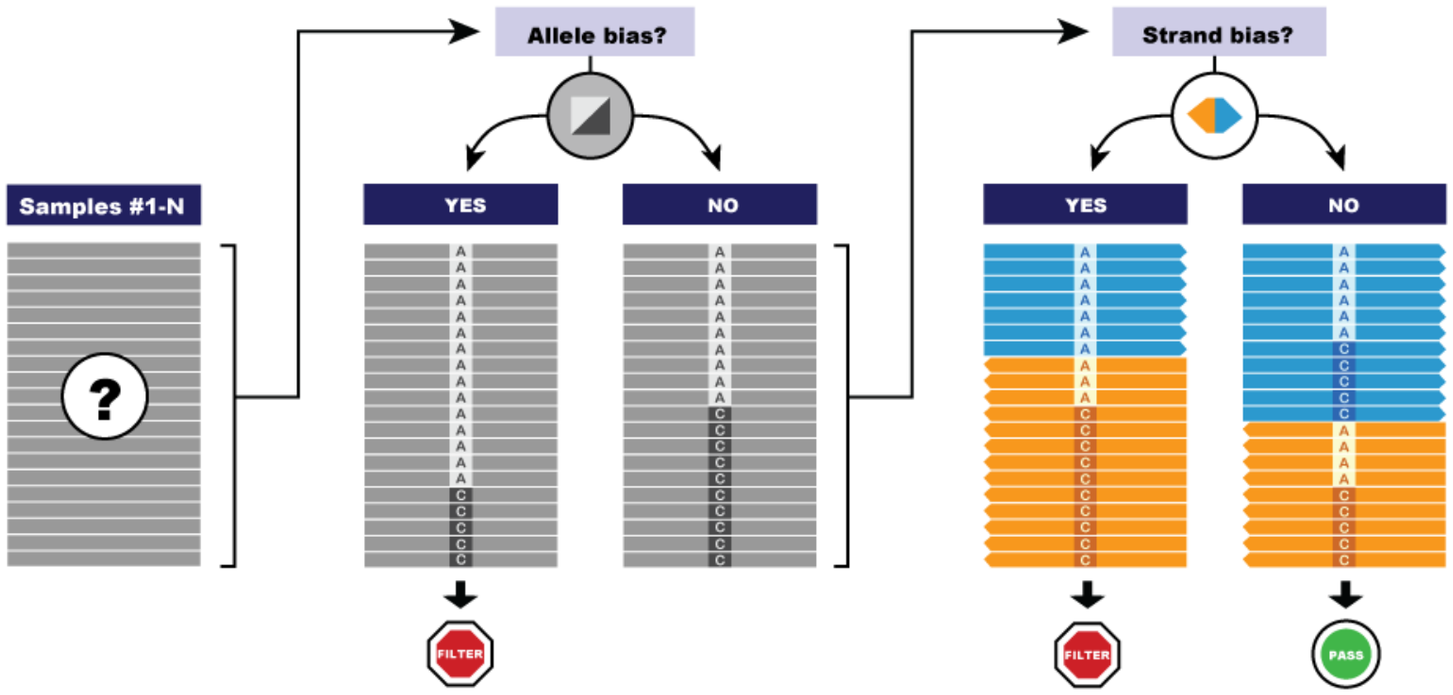
- If we see both samples then we are more confident that there is real variation at this site in the cohort

Joint discovery helps resolve bias issues

A. Single sample showing strand and allelic biases



B. Decision process using evidence from multiple samples to filter out sites showing systematic biases



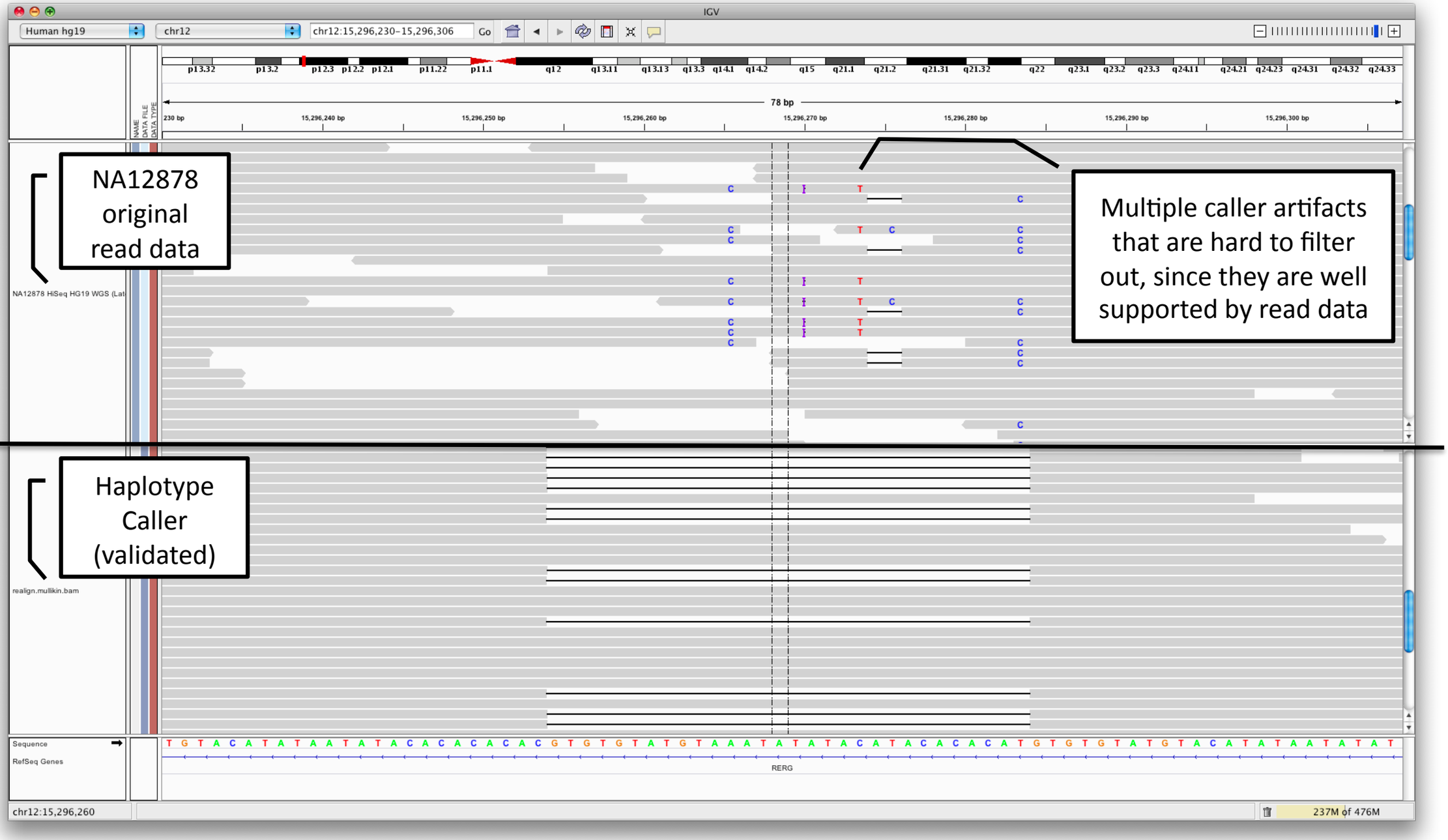
Joint genotyping

- Computational time scales exponentially with number of samples
- The ‘N+1 problem’
    - If you want to add an additional sample, you have to repeat variant calling in all samples

Haplotype-based variant calling

- Previously calling would be done site-by-site
- Genetic variants tend to co-occur with nearby variants (i.e. on haplotypes)
- Reads contain haplotype information
- Allows the analysis of multiple potential variants in a region at the same time
- Local reassembly of complicated regions to refine calls

Artifactual SNPs and small indels caused by large indel can be recovered by assembly



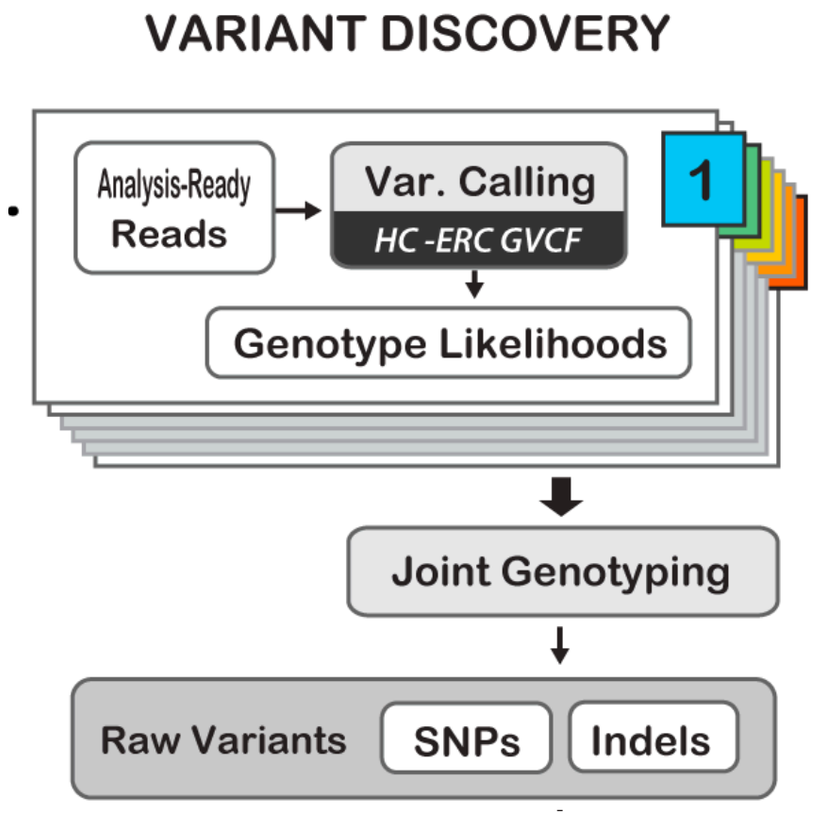
HaplotypeCaller

- Make tentative calls on genotypes for individual samples- ‘genotype likelihoods’

- Joint genotyping with genotype likelihoods across all samples

- More computationally efficient

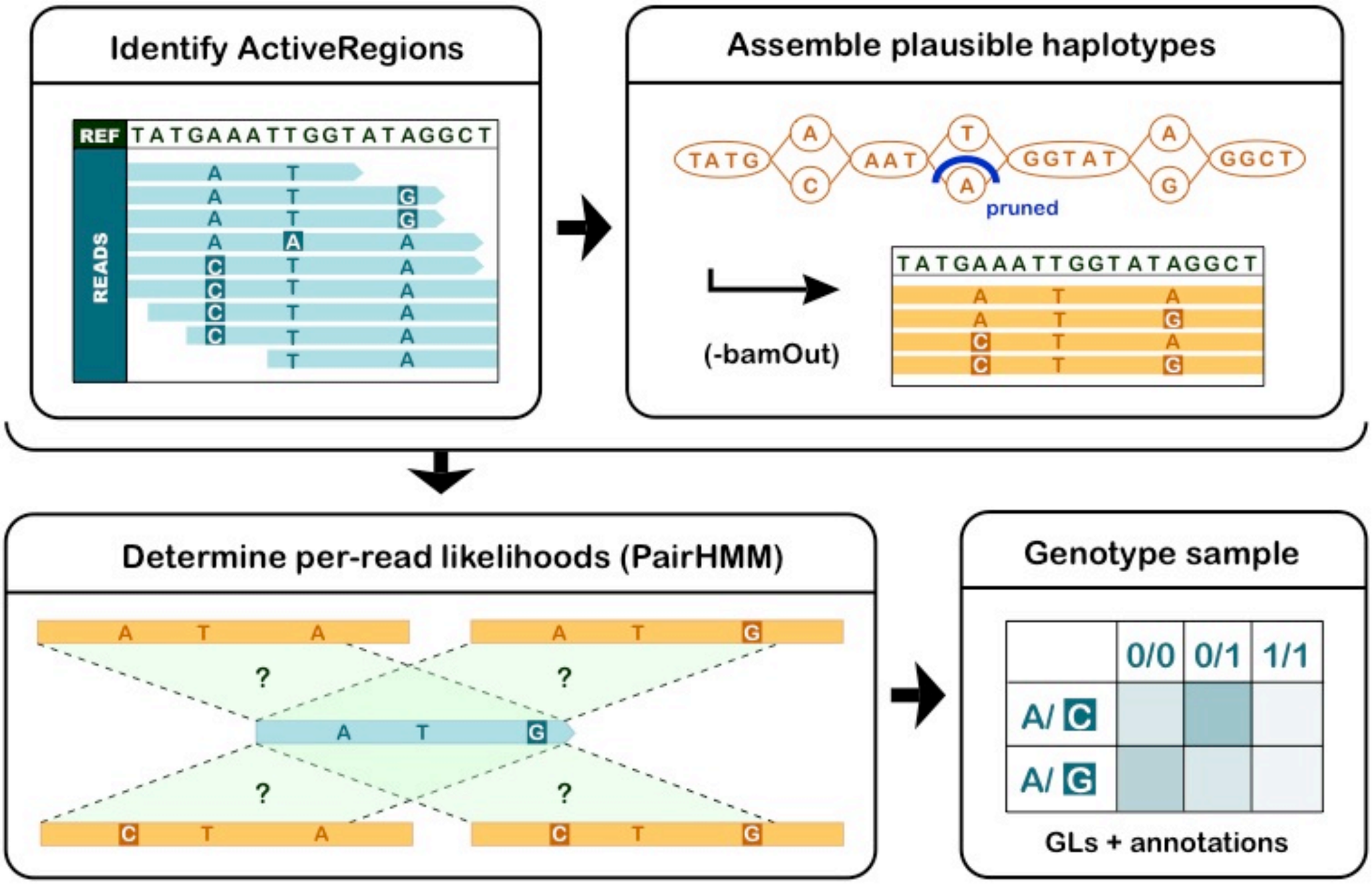
- Additional samples can be added without repeating all variant calling



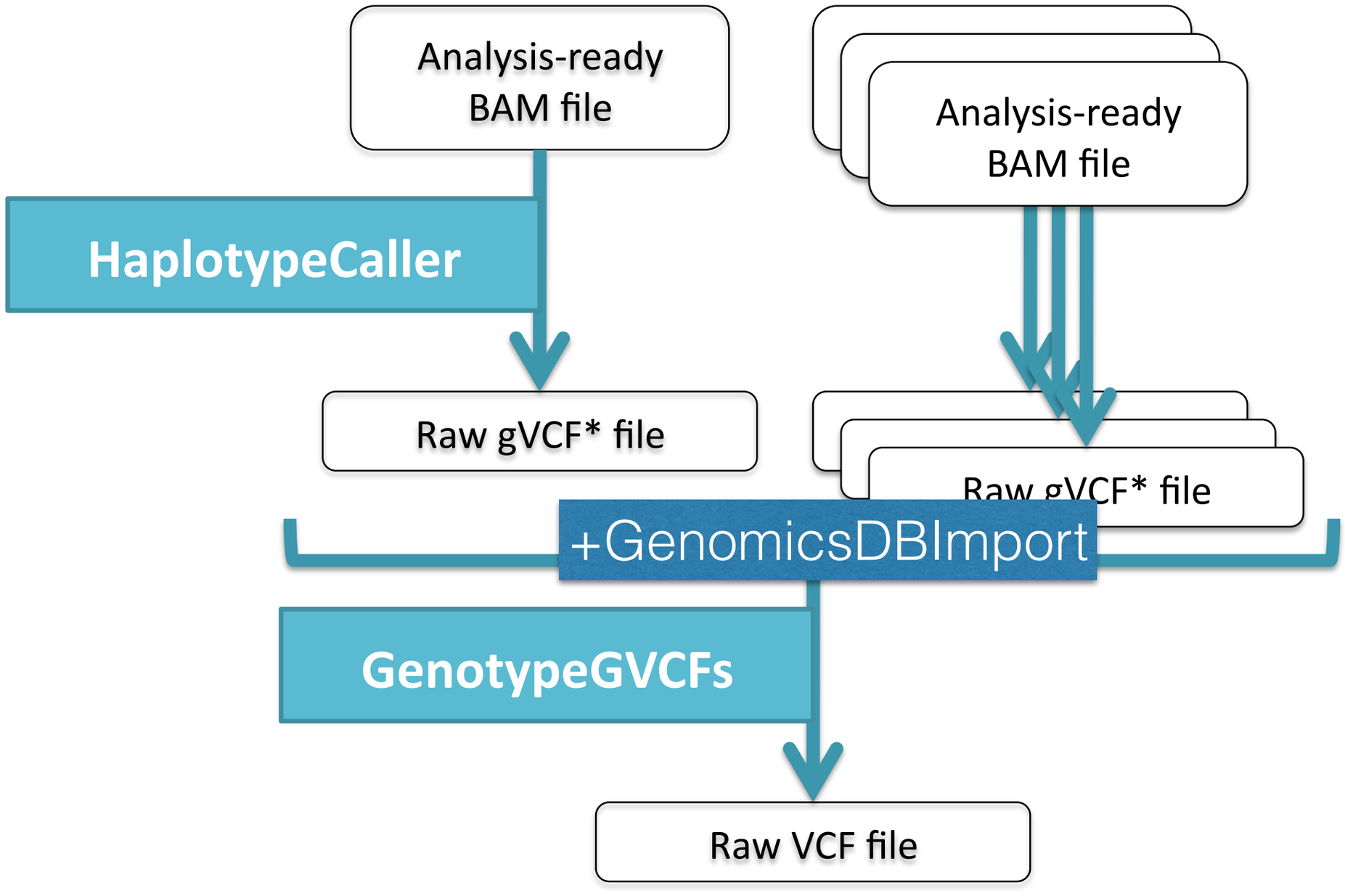
HaplotypeCaller method overview

- Call SNPs, indels, and some SVs simultaneously by doing local re-assembly and considering haplotypes
  - Determine if a region has **potential variation**
  - Make **deBruijn assembly graph** of the region
  - Paths in the graph = **potential haplotypes** to evaluate
  - Calculate **data likelihoods** given the haplotypes (PairHMM)
  - **Identify variants** on most likely haplotypes
  - Compute **allele frequency distribution** to determine most likely allele count, and emit a variant call if appropriate
  - If emitting a variant, **assign genotype** to each sample

HC method illustrated



Variant calling + joint genotyping workflow



VCF Files store variant information

```
##fileformat=VCFv4.1
##reference=1000GenomesPilot-NCBI36
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##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        NA00002        NA00003
```

Header

```
20     14370   rs6054257 G       A       29    PASS    DP=14;AF=0.5;DB
    GT:GQ:DP 0|0:48:1 1|0:48:8 1/1:43:5
20     1110696 rs6040355 A       G,T     67    PASS    DP=10;AF=0.333,0.667;DB
    GT:GQ:DP 1|2:21:6 2|1:2:0   2/2:35:4
20     1230237 .         T       .       47    PASS    DP=13
    GT:GQ:DP 0|0:54:7 0|0:48:4 0/0:61:2
20     1234567 microsat1 GTCT    G,GTACT 50    PASS    DP=9
    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Variant records

INFO field variant annotations

MQ RMS Mapping Quality

DP Approximate read depth; some reads may have been filtered

QD Variant Confidence/Quality by Depth

FS Phred-scaled p-value using Fisher's exact test to detect strand bias

MQRankSum Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities

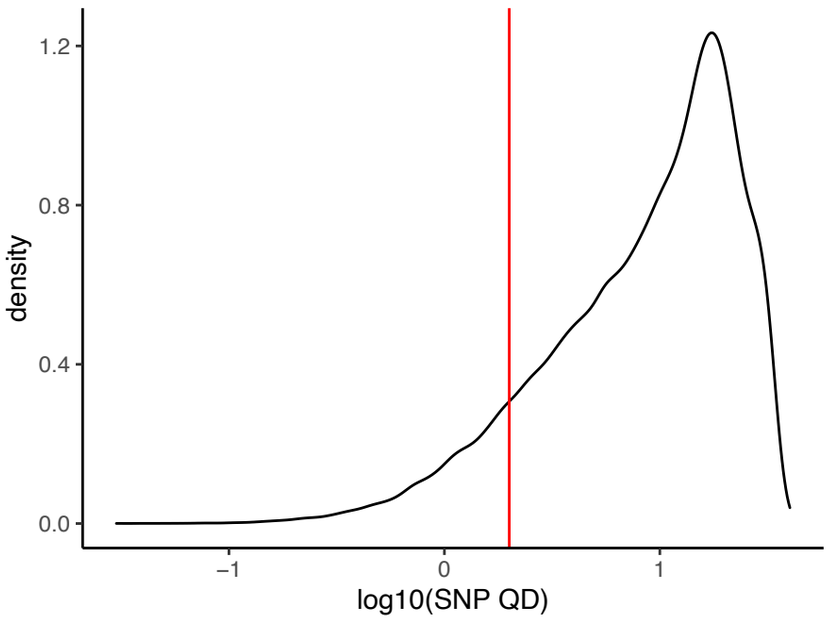
BaseQRankSum Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities

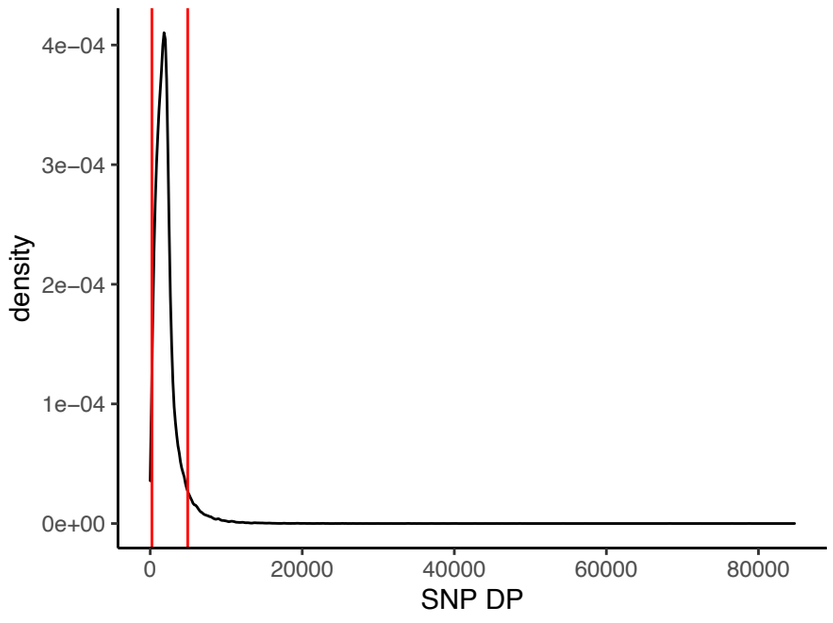
Variant filtering

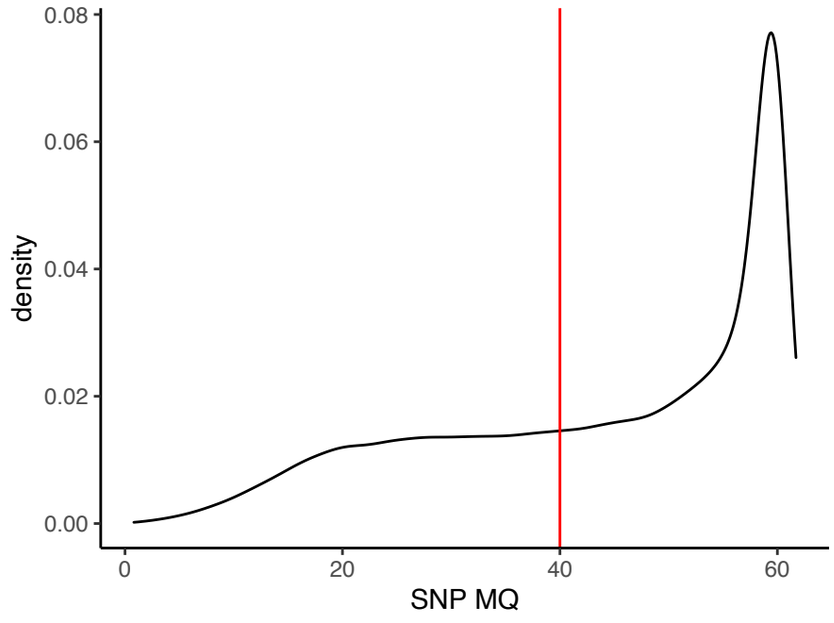
- Variant callers (especially HaplotypeCaller) are very permissive- they will identify a lot of false positives
- They provide variant annotations to help us identify and remove (filter) false positives
  - Hard filtering
  - Variant quality score recalibration (VQSR)

Hard Filters

- Best guess based on variant annotation distributions
- GATK recommendations



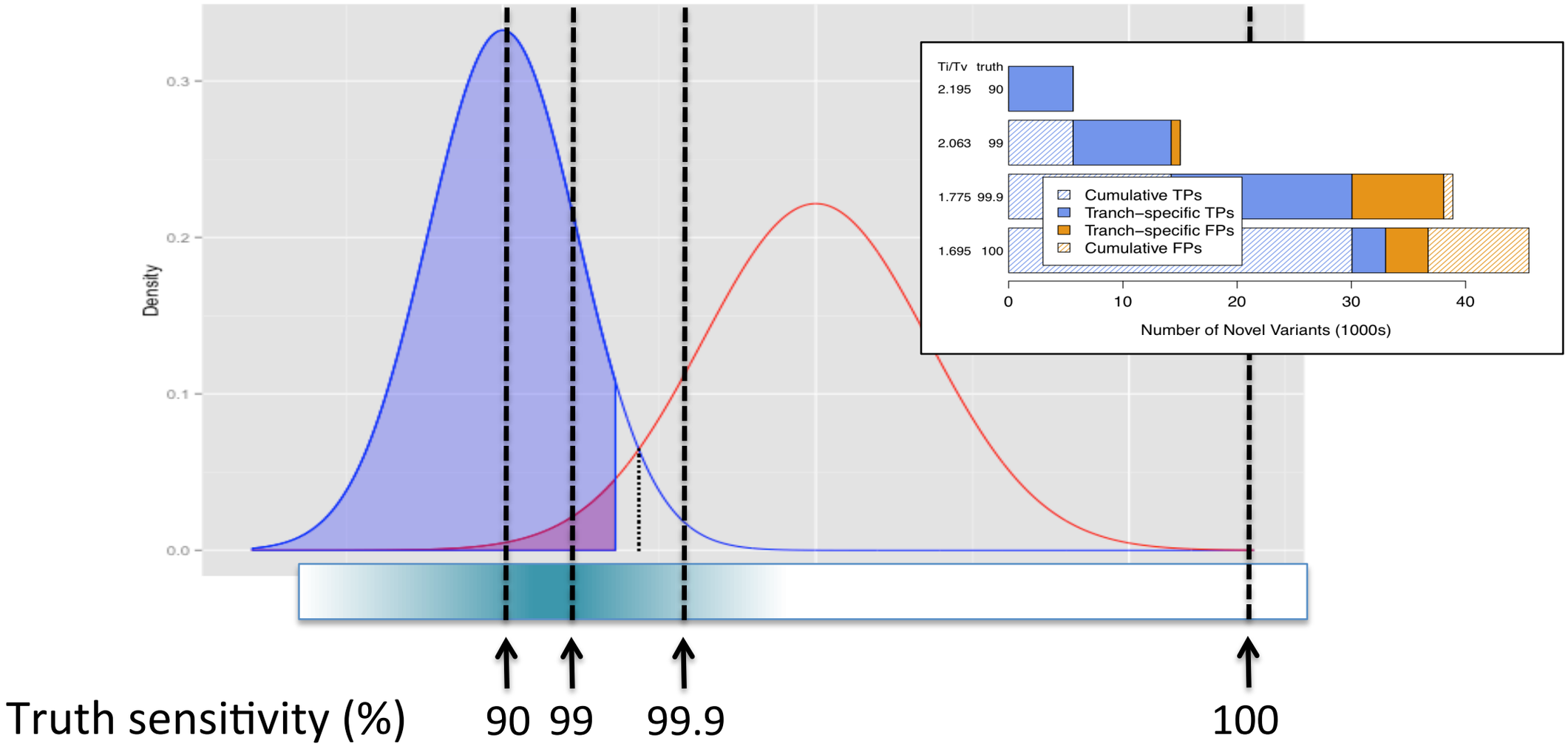




Variant quality score recalibration (VQSR)

- Like BQSR, but for variant quality scores
- Machine-learning approach trained on known true positives and known false positives
  - What are the features (variant annotations) of a true positive?
  - What are the features of a false positive?

Tranches : slices of sensitivity threshold values



sensitivity vs. specificity

Generating true and false positive sets

- Not available outside of Human/model organisms
- Bootstrap approach (like BQSR)
    - Use hard filtering produce high-quality and low-quality SNP sets

Other variant callers

- Alternative filtering approach- call variants with multiple callers and take the intersect

- Or just use a different variant caller- depending on your project requirements

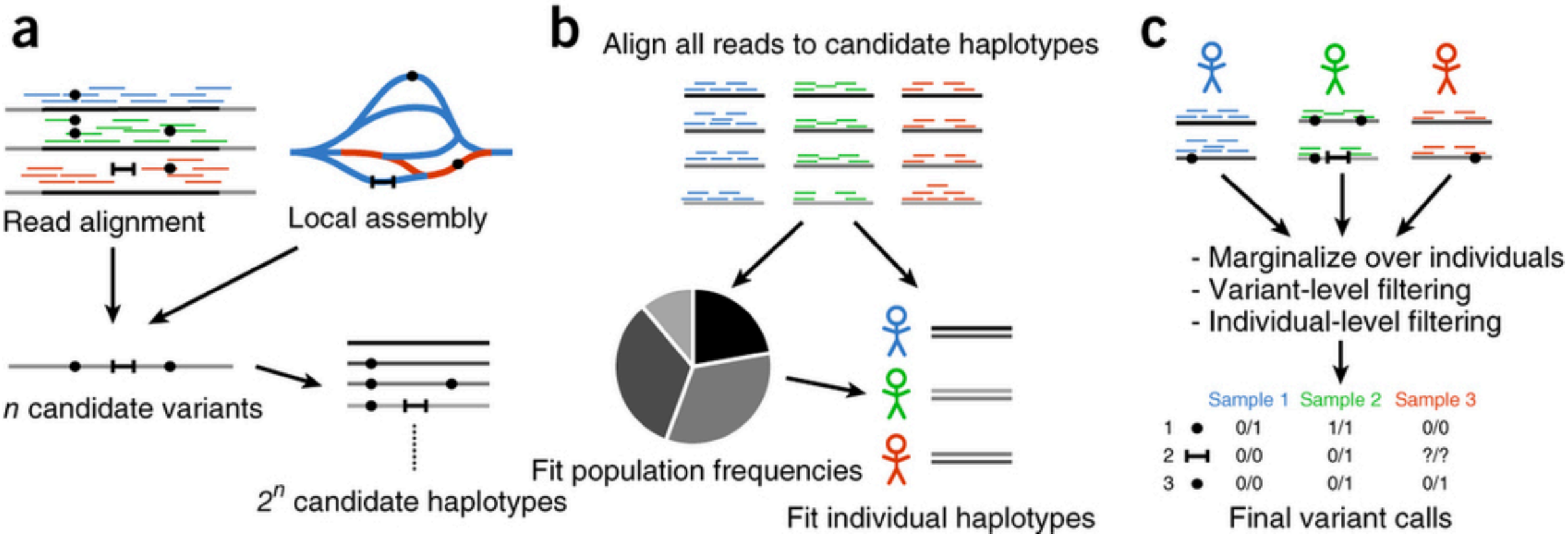
UnifiedGenotyper and Samtools

- Joint genotyping (N+1 problem) but still much faster than HaplotypeCaller
- Site-by-site calling (not haplotype-based)
    - But local realignments can be done in a separate step (GATK IndelRealigner)
- Performs almost as well as HaplotypeCaller with SNPs; much worse with indels

FreeBayes

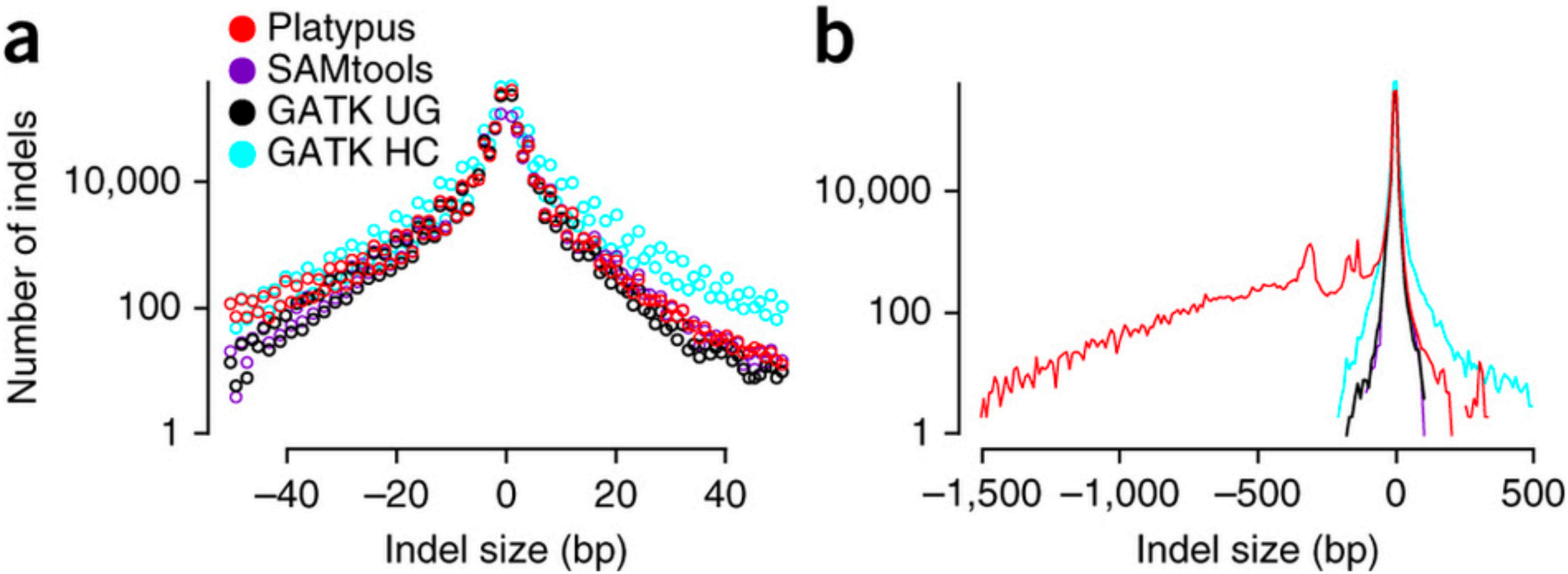
- Free and open-source
- Haplotype-based calling
- Joint genotyping
- Faster than HaplotypeCaller (everything is)

Platypus



Includes local assembly, better for large indels

Platypus



Includes local assembly, better for large indels

ANGSD

- Site-by-site analysis but calculates genotype likelihoods rather than calling variants

- Includes a suite of tools to do common population-genomic analyses with genotype likelihoods

- Useful for projects with low sample coverage

In summary

- **Picard MarkDuplicates- remove duplicated reads**
- BQSR- correct base quality score bias
- **Make GVCFs- genotype likelihoods for individual samples (computationally efficient, N+1)**
- **Joint genotyping using GVCFs**
- Variant filtering- hard filtering or VQSR
- Alternative callers