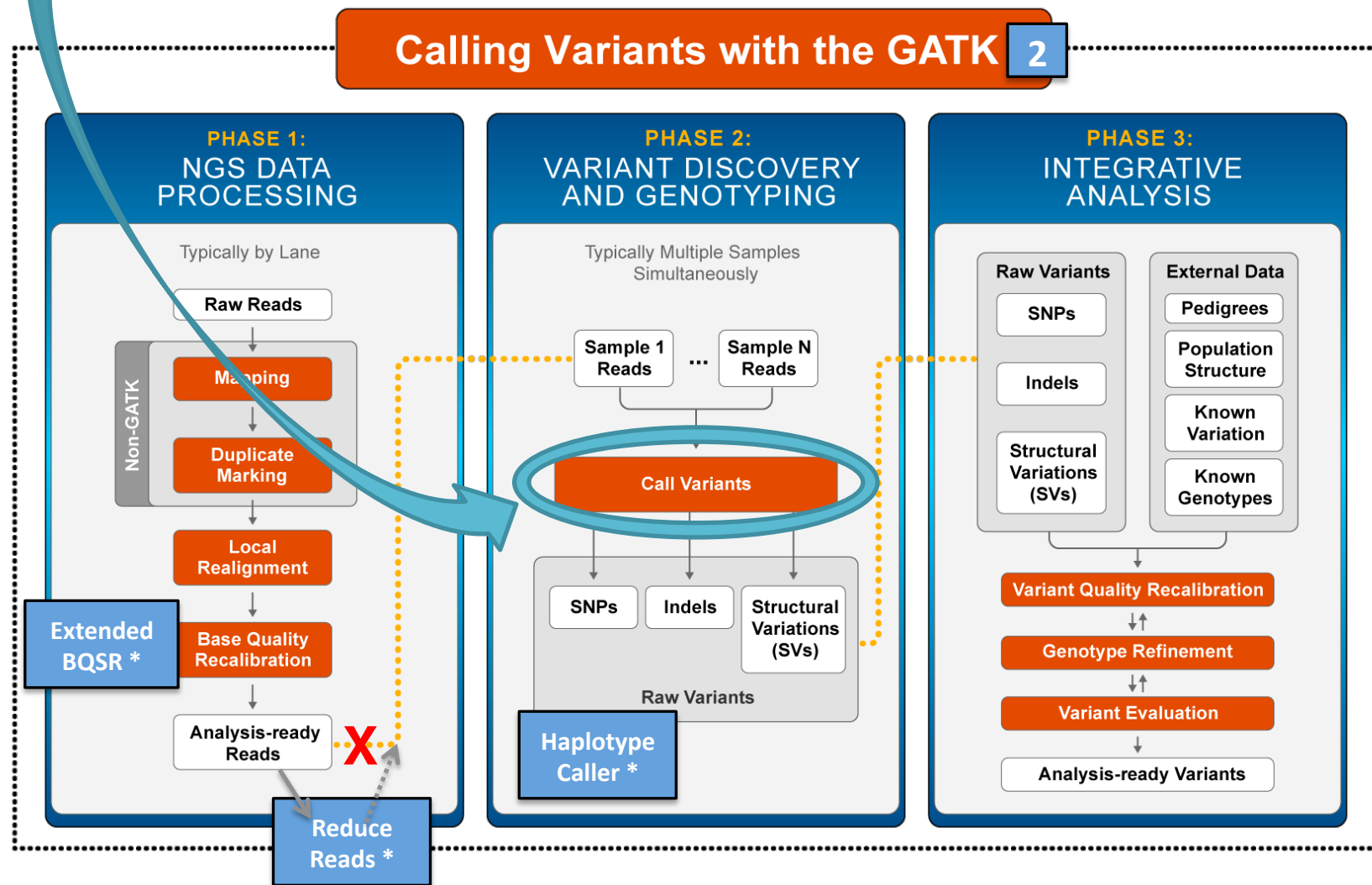


# Calling Variants

Examine the evidence for variation from  
reference via Bayesian inference

# We are here in the Best Practices workflow

## CALLING VARIANTS



\* New tools or functionalities not available in GATK-Lite

**PURPOSE**

# Real mutations are hidden in the noise



# **PRINCIPLES & PROTOCOLS**

## One problem, two approaches

- Genetic variant or random machine noise?  
= large scale Bayesian inference problem
- #1: Initial approach: very fast, independent base assumption
- #2: Evolved approach: more computationally intensive, involves local *de-novo* assembly of the variable region

# Variant calling tools

- UnifiedGenotyper

Call SNPs and indels separately by considering each variant locus independently

- HaplotypeCaller

Call SNPs, indels, and some SVs simultaneously by performing a local *de-novo* assembly

## Unified Genotyper method overview

- Call SNPs and indels separately by considering each variant locus independently
  - Determine the possible SNP and indel alleles
  - Compute, for each sample, for each genotype, likelihoods of data given genotypes
  - Compute the allele frequency distribution to determine most likely allele count, and emit a variant call if determined
  - If we are going to emit a variant, assign a genotype to each sample



# SNP and Indel calling is a large-scale Bayesian modeling problem

Bayesian model

$$\begin{aligned}
 \Pr\{G|D\} &= \frac{\overbrace{\Pr\{G\}}^{\text{Prior of the genotype}} \overbrace{\Pr\{D|G\}}^{\text{Likelihood of the genotype}}}{\sum_i \Pr\{G_i\} \Pr\{D|G_i\}}, \text{ [Bayes' rule]} \\
 \Pr\{D|G\} &= \prod_j \left( \frac{\Pr\{D_j|H_1\}}{2} + \frac{\Pr\{D_j|H_2\}}{2} \right) \text{ where } \overbrace{G = H_1 H_2}^{\text{Diploid assumption}} \\
 \Pr\{D|H\} &\text{ is the haploid likelihood function}
 \end{aligned}$$

- Inference: what is the genotype  $G$  of each sample given read data  $D$  for each sample?
- Calculate via Bayes' rule the probability of each possible  $G$
- Product expansion assumes reads are independent
- Relies on a likelihood function to estimate probability of sample data given proposed haplotype

# SNP genotype likelihoods

$$\Pr\{D_j|H\} = \Pr\{D_j|b\}, \text{ [single base pileup]}$$

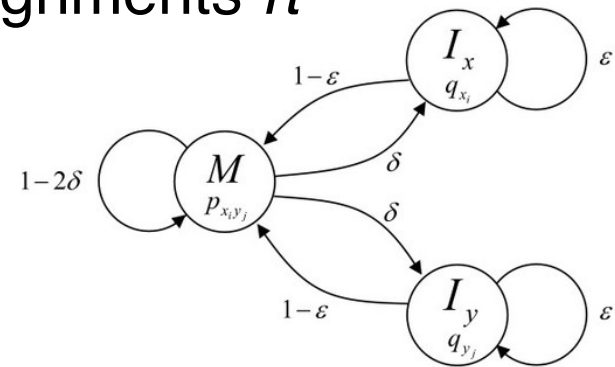
$$\Pr\{D_j|b\} = \begin{cases} 1 - \epsilon_j & D_j = b, \\ \epsilon_j & \text{otherwise.} \end{cases}$$

- All diploid genotypes (AA, AC, ..., GT, TT) considered at each base
- Likelihood of genotype computed using only pileup of bases and associated quality scores at given locus
- Only “good bases” are included: those satisfying minimum base quality, mapping read quality, pair mapping quality, NQS

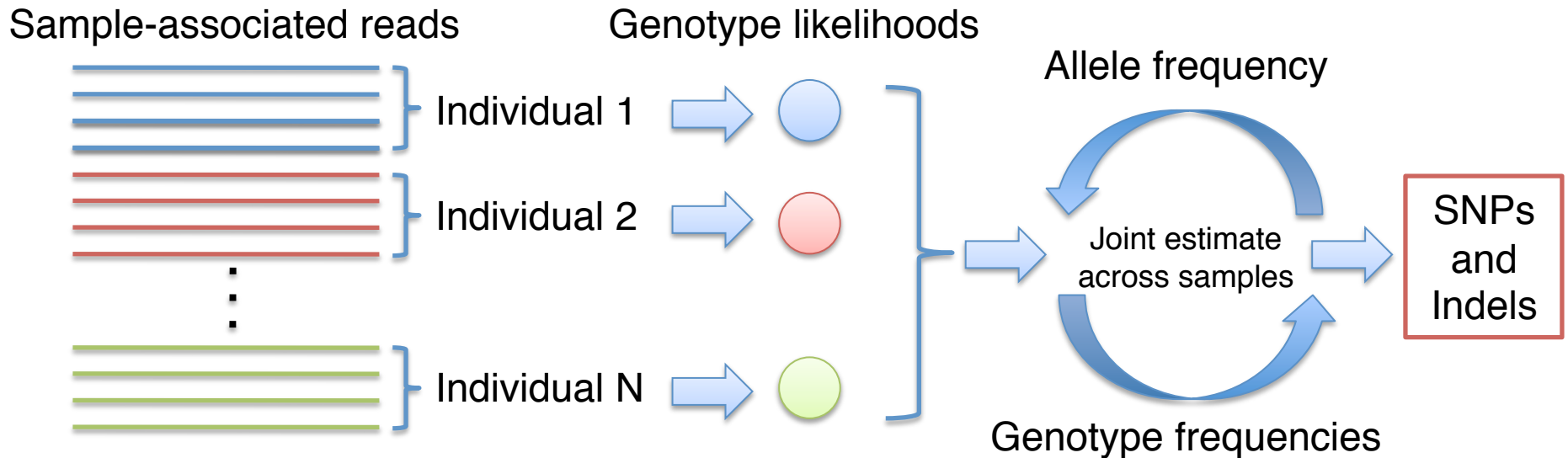
# Indel genotype likelihoods

$$\Pr\{D_j|H\} = \sum_{\substack{\text{alignments } \pi \\ \text{of } D_j \text{ to } H}} \Pr\{D_j, \pi\}$$

- Haplotypes  $H_i$  are discovered from indels in the reads
- Diploid genotypes  $G$  for all haplotype  $H_i H_j$  combinations
- For each haplotype  $H_i$ , calculate likelihood of each read  $D_j$  marginalizing over all possible alignments  $\pi$
- Sum computed by a standard HMM with context-dependent affine gap penalties using haplotype and read bases and quality scores



# Multi-sample calling integrates per sample likelihoods to jointly estimate allele frequency of variation



- Simultaneous estimation of:
  - Allele frequency (AF) spectrum  $\Pr\{AF = i \mid D\}$
  - The probability that a variant exists  $\Pr\{AF > 0 \mid D\}$
  - Assignment of genotypes to each sample

# UnifiedGenotyper

- Inputs
  - `-l` Input analysis-ready bam file
- Other parameters of interest
  - `-stand_call_conf` Qual score at which to call the variant
  - `-stand_emit_conf` Qual score at which to emit the variant as filtered
- Outputs
  - `-o` Raw mutation calls in VCF format
- Typical command line

```
java -jar GenomeAnalysisTK.jar -R human.fasta -T UnifiedGenotyper -l  
input.bam -o output.vcf
```

## HaplotypeCaller method overview

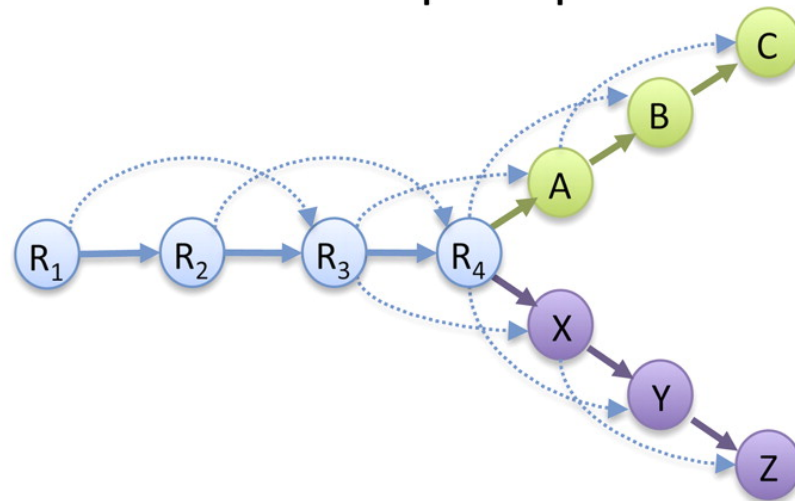
- Call SNPs, indels, and some SVs simultaneously by performing a local *de-novo* assembly
  - Determine if a region has the potential to be variable
  - Construct a deBruijn assembly of the region
  - The paths in the graph are potential haplotypes that need to be evaluated
  - Calculate haplotype likelihoods given the data using the PairHMM model
  - Determine if there are any variants on the most likely haplotypes
  - Compute the allele frequency distribution to determine most likely allele count, and emit a variant call if determined
  - If we are going to emit a variant, assign a genotype to each sample

# Propose haplotypes with local de novo assembly via DeBruijn graphs

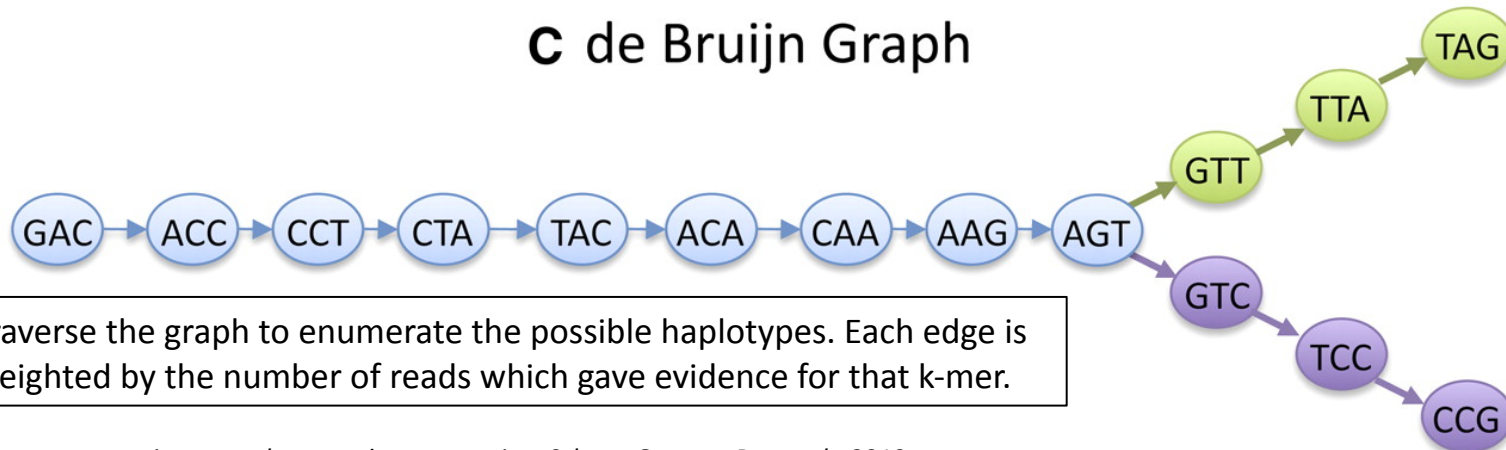
## A Read Layout

R<sub>1</sub>: GACCTACA  
R<sub>2</sub>: ACCTACAA  
R<sub>3</sub>: CCTACAAG  
R<sub>4</sub>: CTACAAGT  
A: TACAAGTT  
B: ACAAGTTA  
C: CAAGTTAG  
X: TACAAGTC  
Y: ACAAGTCC  
Z: CAAGTCCG

## B Overlap Graph



## C de Bruijn Graph



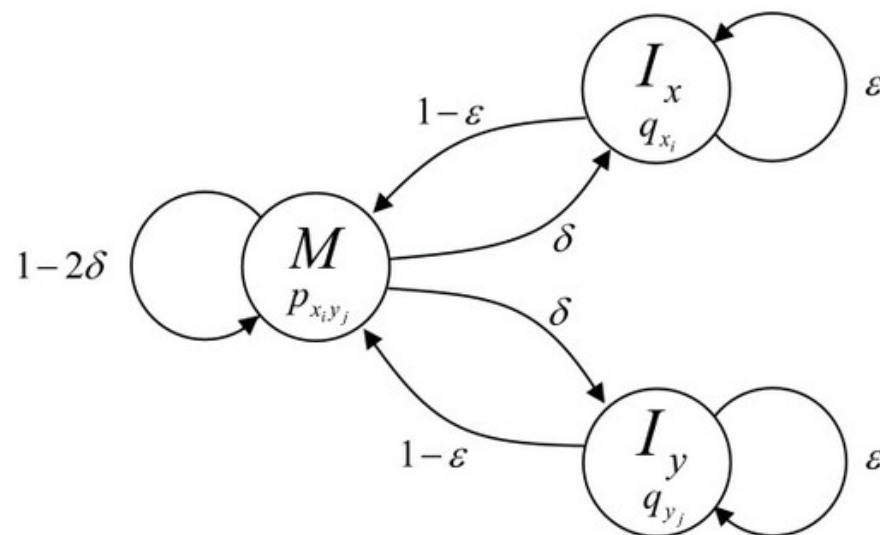
# Evaluate haplotypes with Pair HMM

Bayesian model

$$\Pr\{G|D\} = \frac{\overbrace{\Pr\{G\}}^{\text{Prior of the genotype}} \overbrace{\Pr\{D|G\}}^{\text{Likelihood of the genotype}}}{\sum_i \Pr\{G_i\} \Pr\{D|G_i\}}, \text{ [Bayes' rule]}$$

$$\Pr\{D|G\} = \prod_j \left( \frac{\Pr\{D_j|H_1\}}{2} + \frac{\Pr\{D_j|H_2\}}{2} \right) \text{ where } \overbrace{G = H_1 H_2}^{\text{Diploid assumption}}$$

$\Pr\{D|H\}$  is the haploid likelihood function

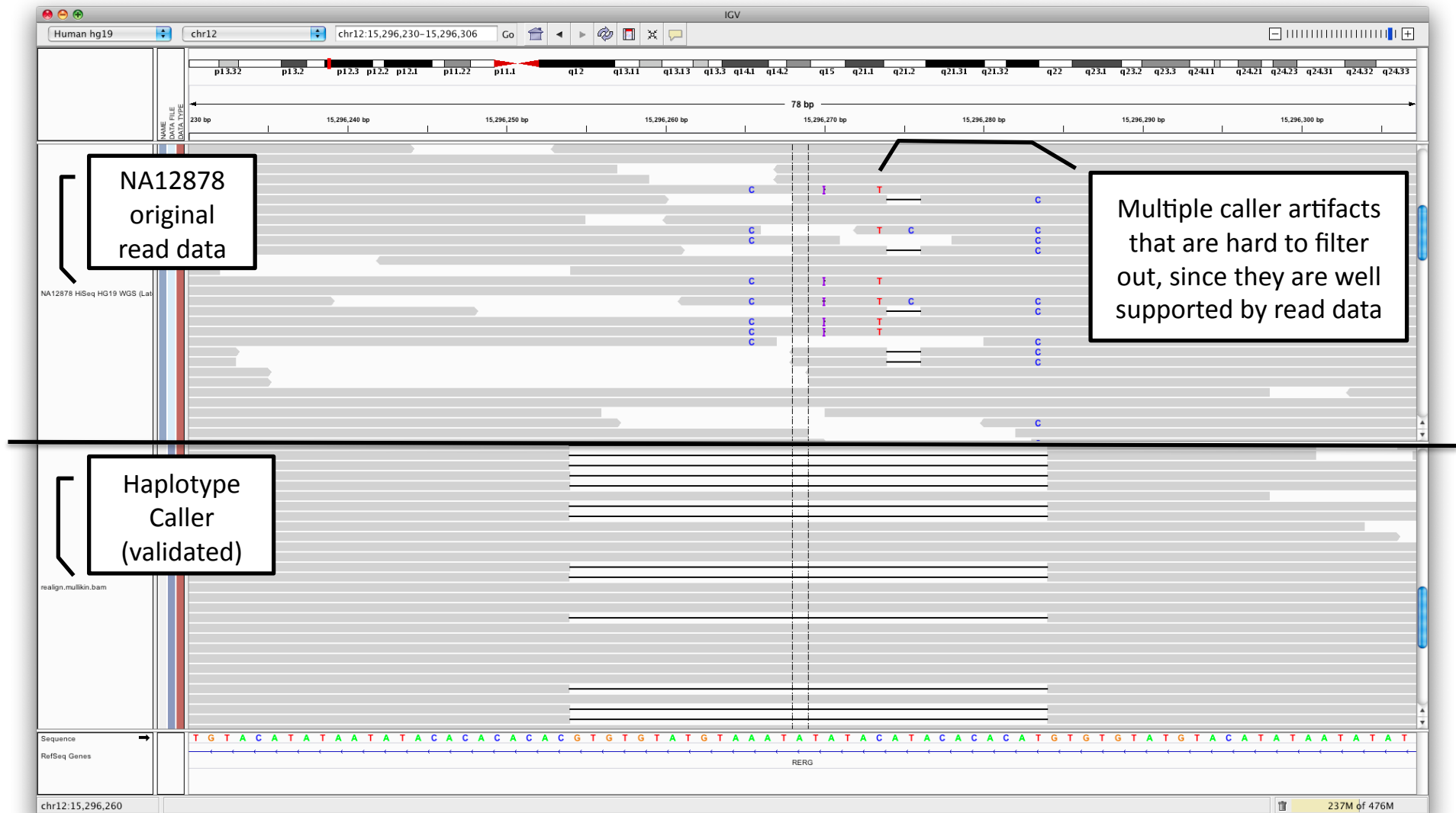


Empirical gap penalties derived from data using new BQSR.

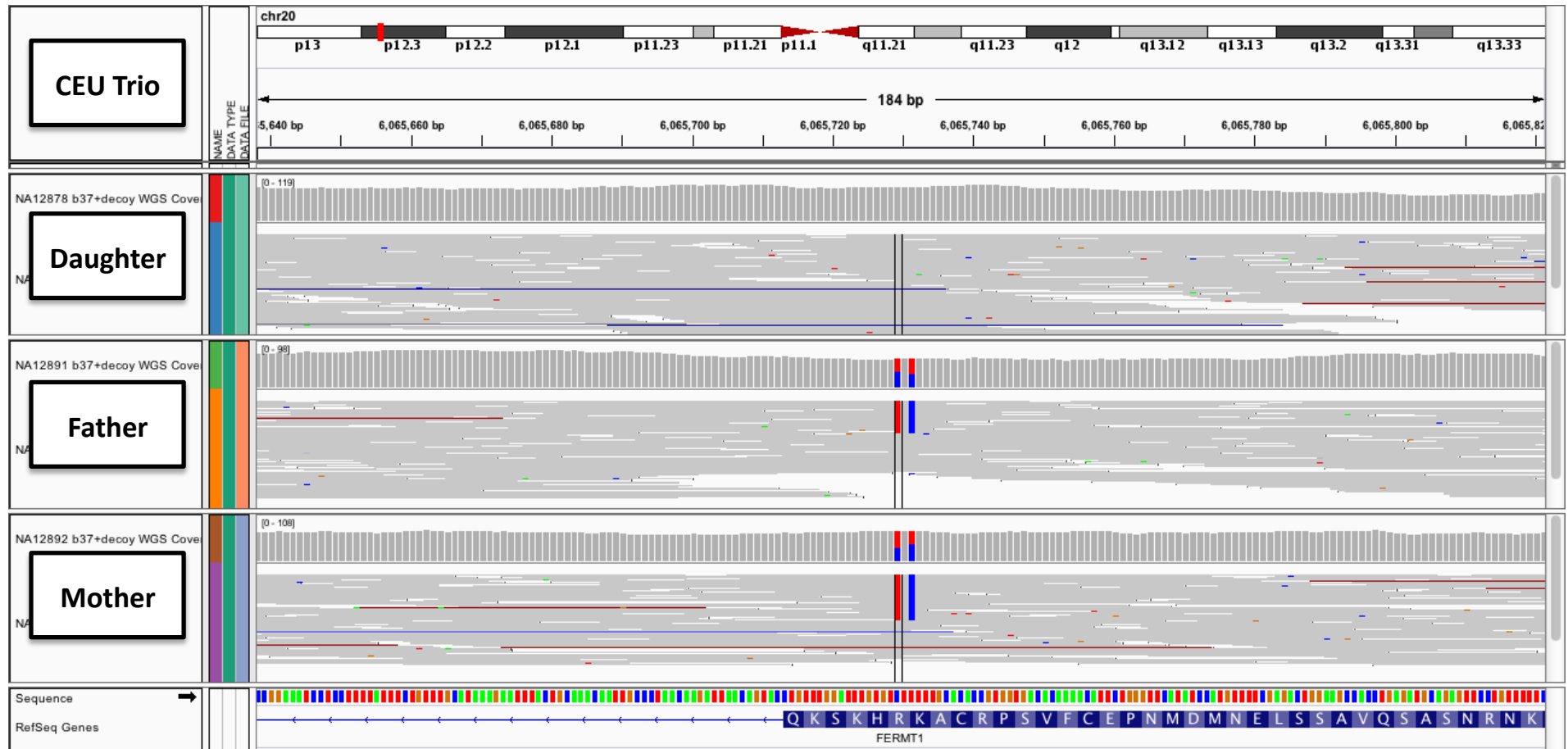
Base mismatch penalties are the base quality scores.



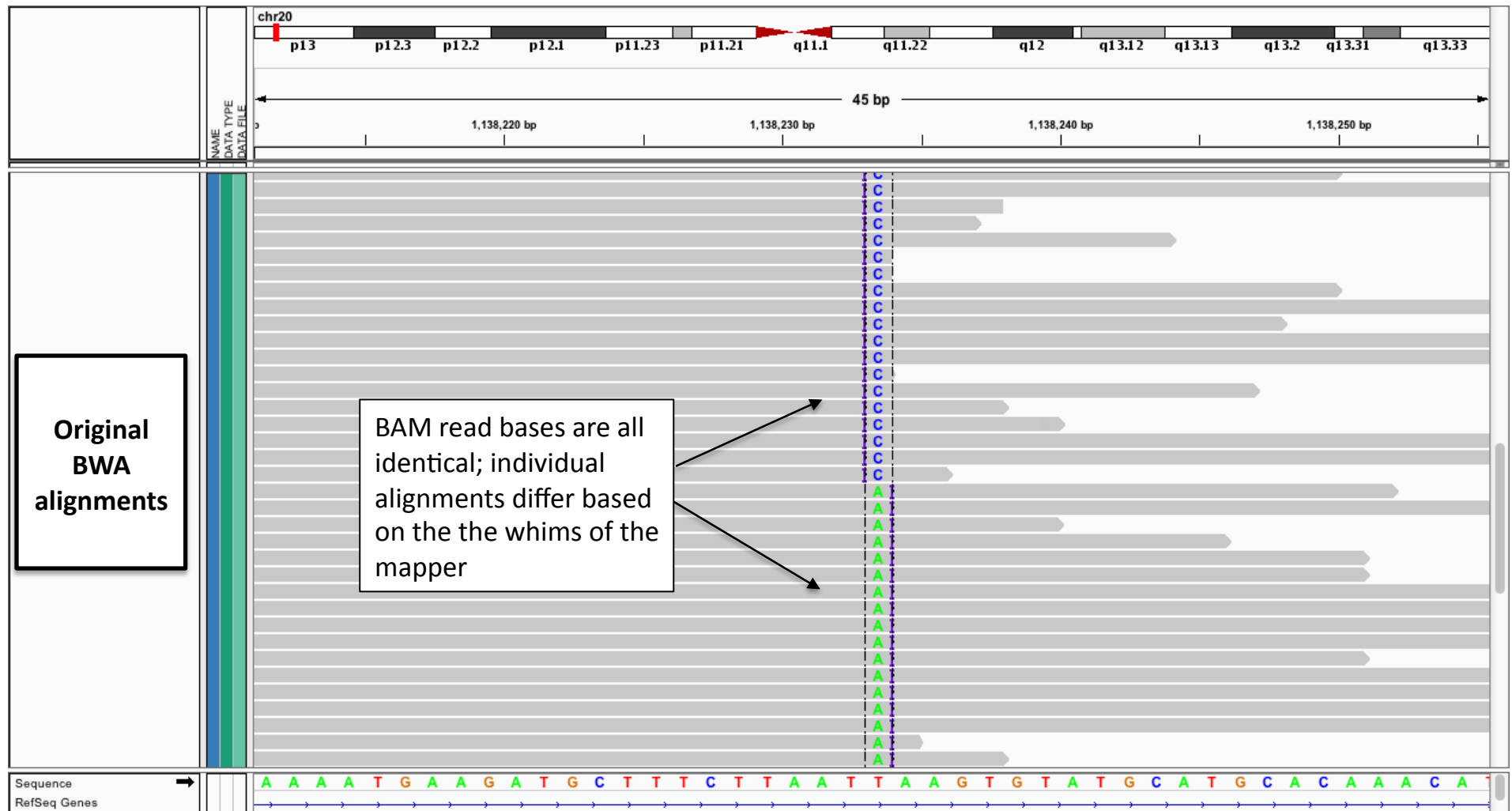
# Artifactual SNPs and small indels caused by large indel recovered by assembly



As an added bonus we now get physical phasing for free, which allows us to distinguish between e.g. MNPs and compound hets



# Allele determination is much more accurate through local assembly of candidate haplotypes



**-assembly:** 1 multi-allelic SNP and two 1bp indels are called  
**+assembly:** Only the complex substitution (TT to TAC) is called

# HaplotypeCaller

- Inputs
  - `-I` Input analysis-ready bam file
- Other parameters of interest
  - `-stand_call_conf` Qual score at which to call the variant
  - `-stand_emit_conf` Qual score at which to emit the variant as filtered
  - `-minPruning` Amount of pruning to do in the deBruijn graph
- Outputs
  - `-o` Raw mutation calls in VCF format
- Typical command line

```
java -jar GenomeAnalysisTK.jar -R human.fasta -T HaplotypeCaller -I
input.bam -minPruning 3 -o output.vcf
```

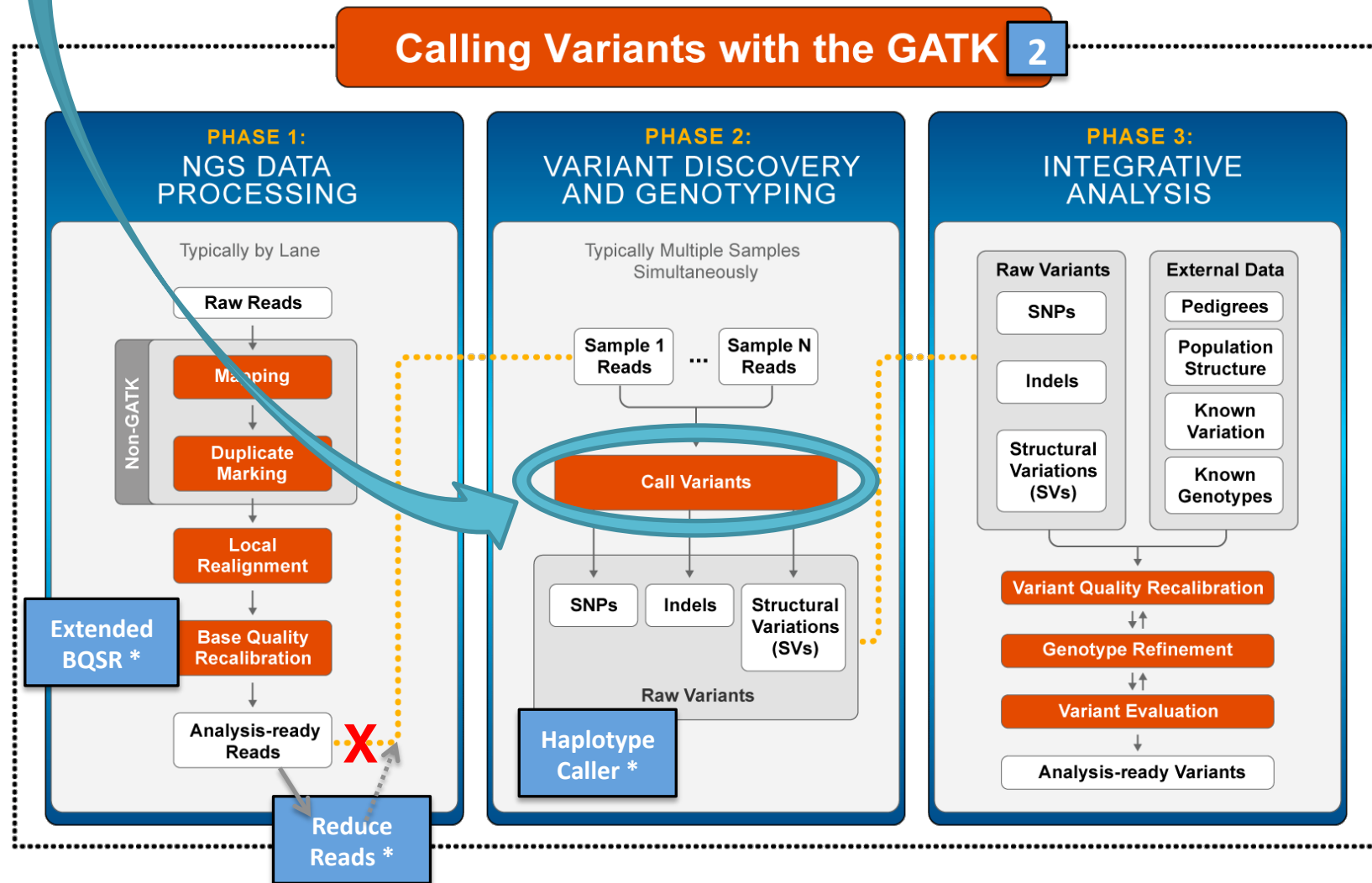
# **RESULTS**

## Did the mutation calling work properly?

- Raw callsets are often very large and full of false positive mutation calls
- Further work is needed before this callset can be used for any meaningful analysis!
- See downstream steps (VQSR etc.) on how to assess the quality of a variant callset

# We were here in the Best Practices workflow

*NEXT STEP: VARIANT RECALIBRATION*



\* New tools or functionalities not available in GATK-Lite

## Further reading

<http://www.broadinstitute.org/gatk/guide/topic?name=intro>

<http://www.broadinstitute.org/gatk/guide/topic?name=best-practices>

<http://www.broadinstitute.org/gatk/guide/article?id=1237>

[http://www.broadinstitute.org/gatk/gatkdocs/  
org\\_broadinstitute\\_sting\\_gatk\\_walkers\\_genotyper\\_UnifiedGenotyper.html](http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_genotyper_UnifiedGenotyper.html)

[http://www.broadinstitute.org/gatk/gatkdocs/  
org\\_broadinstitute\\_sting\\_gatk\\_walkers\\_haplotypecaller\\_HaplotypeCaller.html](http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_haplotypecaller_HaplotypeCaller.html)