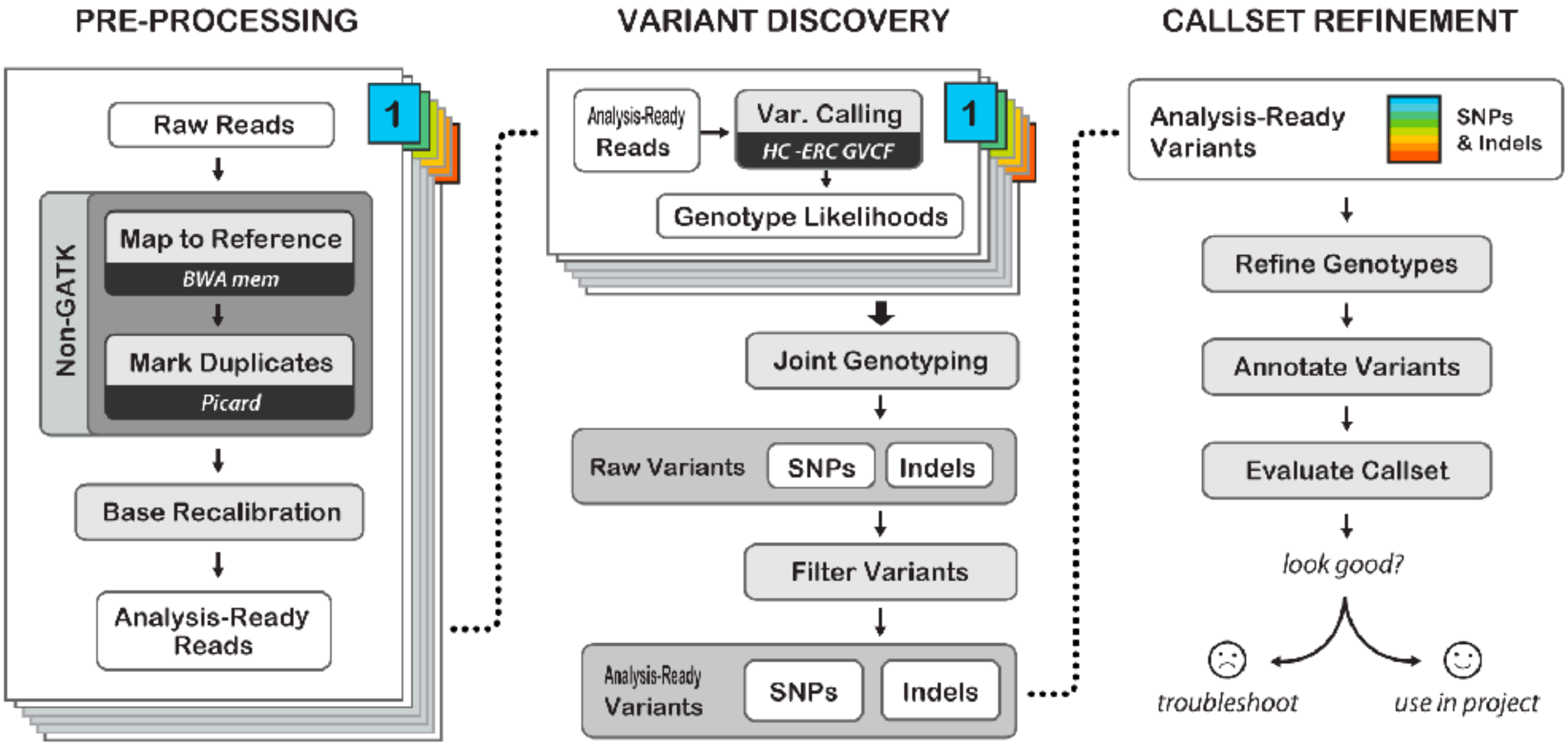
Topic 5: Variant calling using GATK

INTRO TO BIOINFORMATICS - MONASH SBS 2019

Learning Goals

- Define the steps involved in SNP calling and what they are doing.
- Understand the reason for haplotype based SNP calling.
- Understand the recalibration approach to variant filtering.
- Define the N+1 problem in genotyping.

GATK Best Practises for Variant Discovery in DNAseq



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

BAM headers: an essential part of a BAM file

```
@HD    VN:1.0  GO:none SO:coordinate
@SQ    SN:chrM        LN:16571
@SQ    SN:chr1        LN:247249719
@SQ    SN:chr2        LN:242951149
[cut for clarity]
@SQ    SN:chr9        LN:140273252
@SQ    SN:chr10       LN:135374737
@SQ    SN:chr11       LN:134452384
[cut for clarity]
@SQ    SN:chr22    LN:49691432
@SQ    SN:chrX        LN:154913754
@SQ    SN:chrY        LN:57772954
@RG    ID:20FUK.1     PL:illumina    PU:20FUKAAXX100202.1    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.2     PL:illumina    PU:20FUKAAXX100202.2    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.3     PL:illumina    PU:20FUKAAXX100202.3    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.4     PL:illumina    PU:20FUKAAXX100202.4    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.5     PL:illumina    PU:20FUKAAXX100202.5    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.6     PL:illumina    PU:20FUKAAXX100202.6    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.7     PL:illumina    PU:20FUKAAXX100202.7    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.8     PL:illumina    PU:20FUKAAXX100202.8    LB:Solexa-18484 SM:NA12878    CN:BI
@PG    ID:BWA  VN:0.5.7       CL:tk
@PG    ID:GATK TableRecalibration     VN:1.0.2864
20FUKAAXX100202:1:1:12730:189900      163    chrM   1      60     101M   =      282    381
        GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTA…[more bases]
        ?BA@A>BBBBACBBAC@BBCBBCBC@BC@CAC@:BBCBBCACAACBABCBCCAB…[more quals]
        RG:Z:20FUK.1    NM:i:1  SM:i:37 AM:i:37 MD:Z:72G28      MQ:i:60 PG:Z:BWA       UQ:i:33
```

Required: Standard header

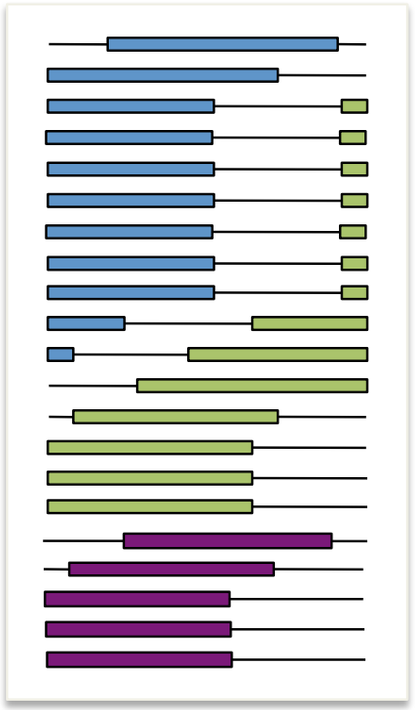
Essential: contigs of aligned reference sequence. Should be in karyotypic order.

Essential: read groups. Carries platform (PL), library (LB), and sample (SM) information. Each read is associated with a read group

Useful: Data processing tools applied to the reads

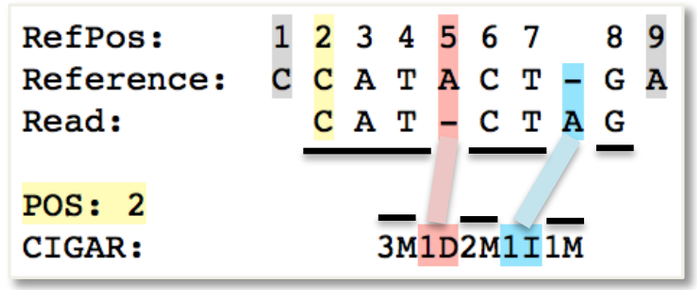
Mapping short reads to a reference is simple in principle

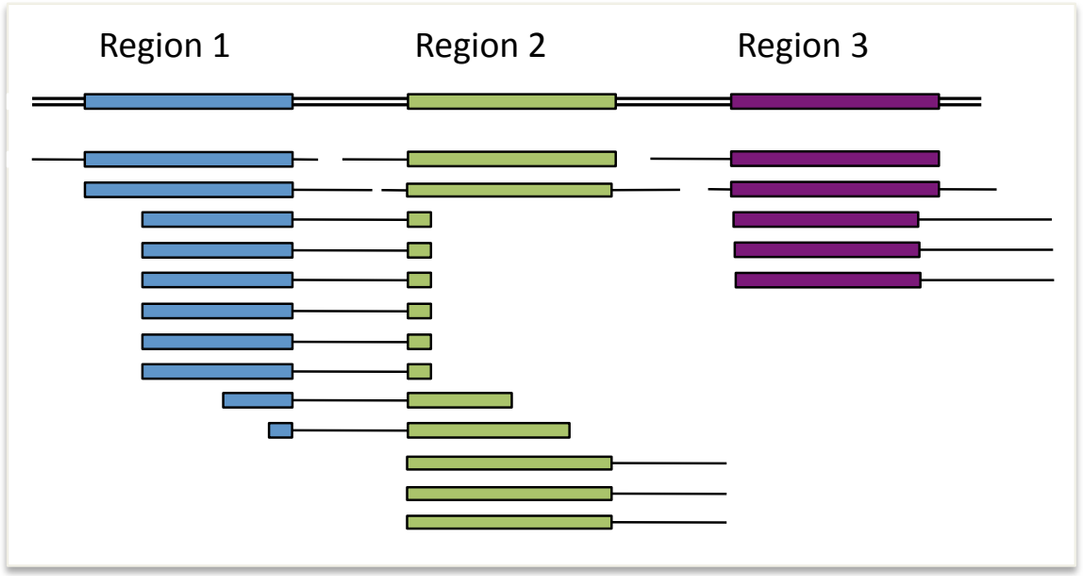
Enormous pile of short reads from NGS



Mapping and alignment algorithms

Identify where the read matches the reference sequence and record match details as CIGAR string



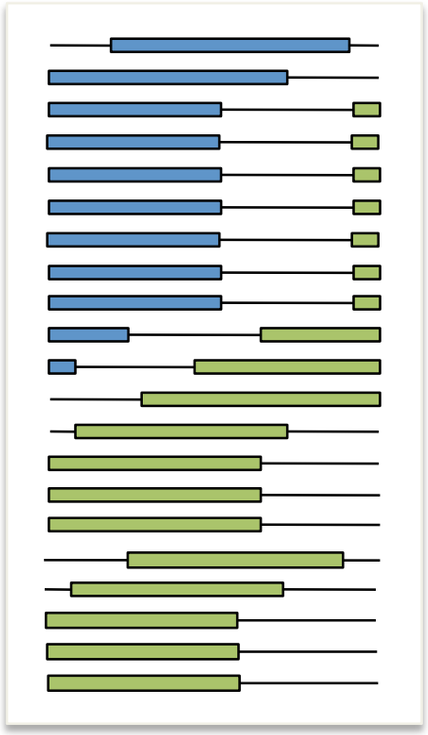


Reference genome

Reads mapped to reference

But mapping is actually very hard because of mismatches (true mutations or sequencing errors), duplicated regions etc.

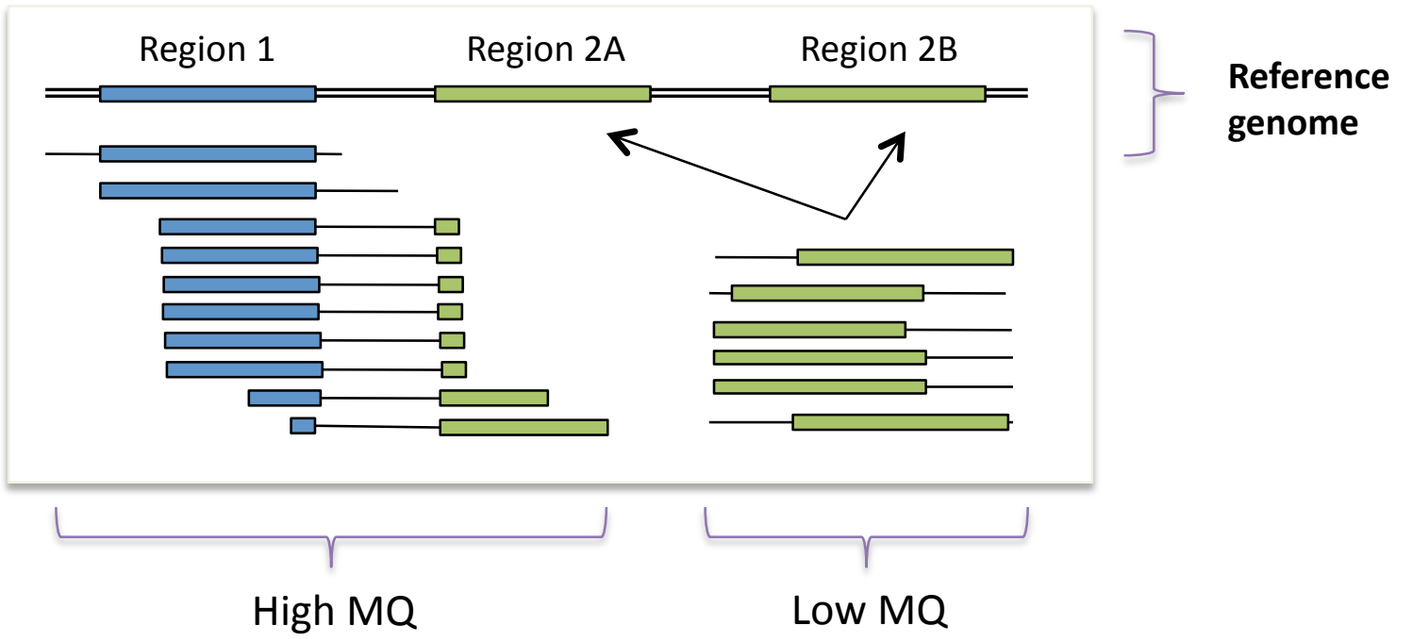
Enormous pile of short reads from NGS



Mapping and alignment algorithms

Mapping algorithms account for this by choosing the most likely placement

→ **mapping quality (MQ)**

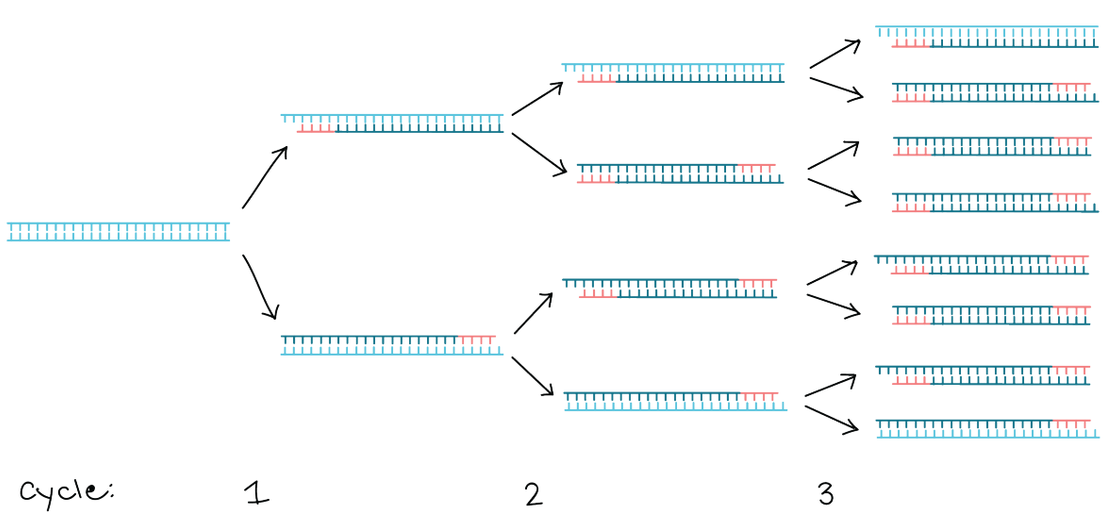


For more information see:

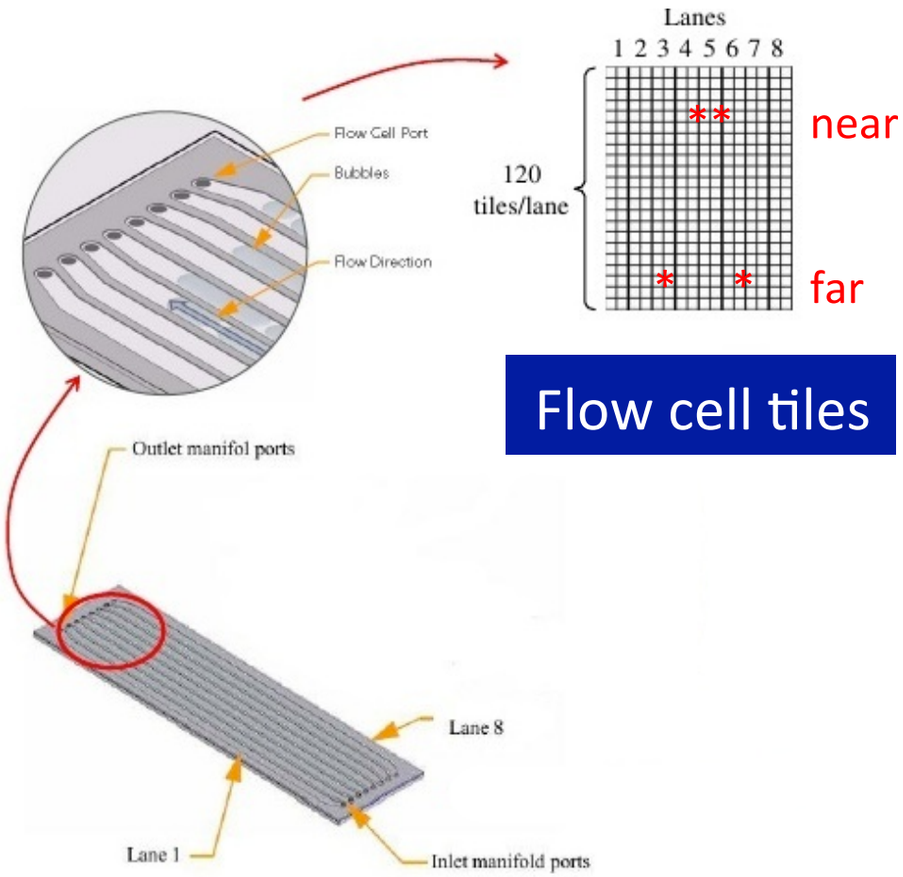
Li and Homer (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*.

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

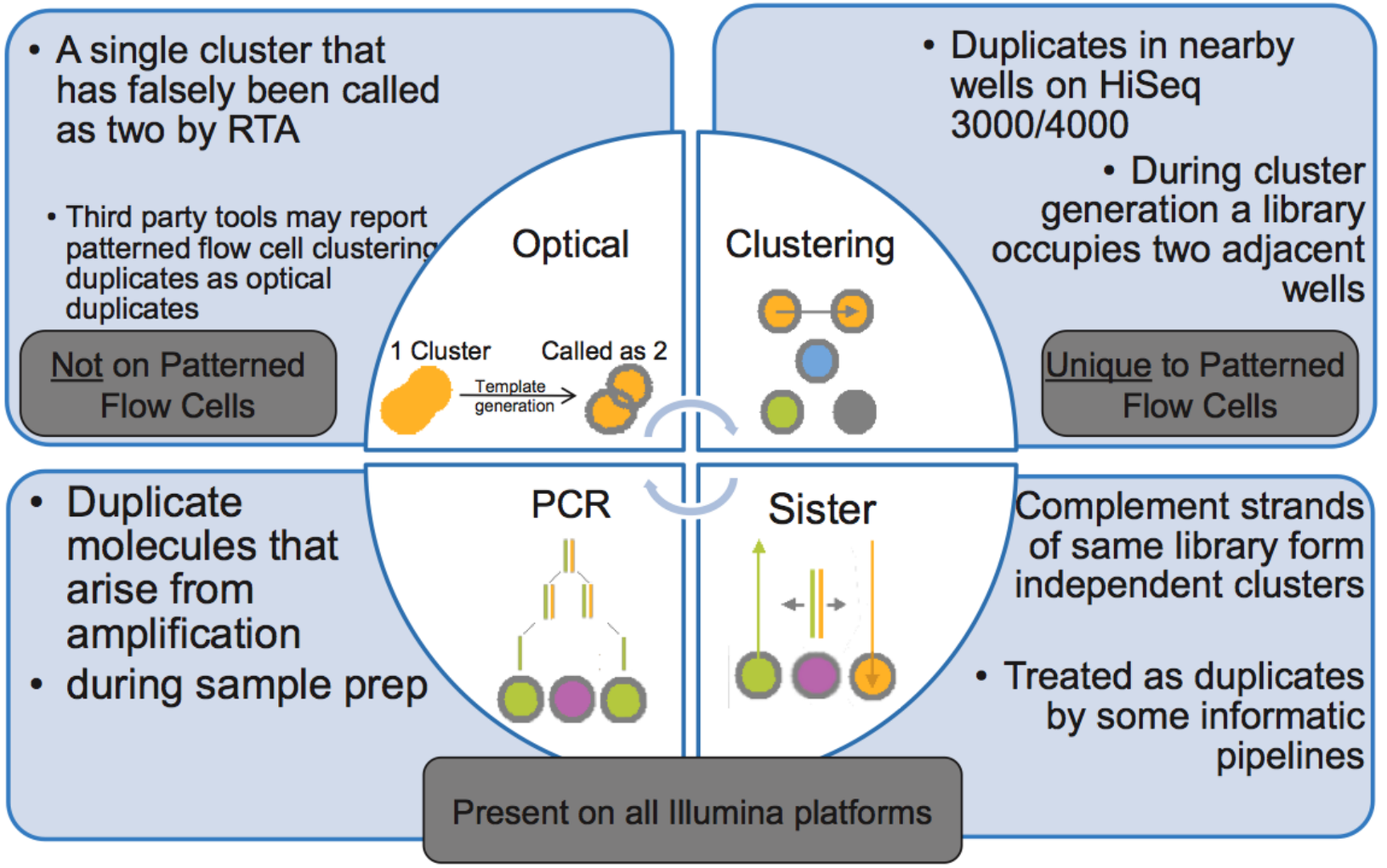


Flow cell tiles

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

Duplicates are flagged the same but can be tagged differently (DT)

0x400 flag DT:SQ



0x400 flag DT:SQ

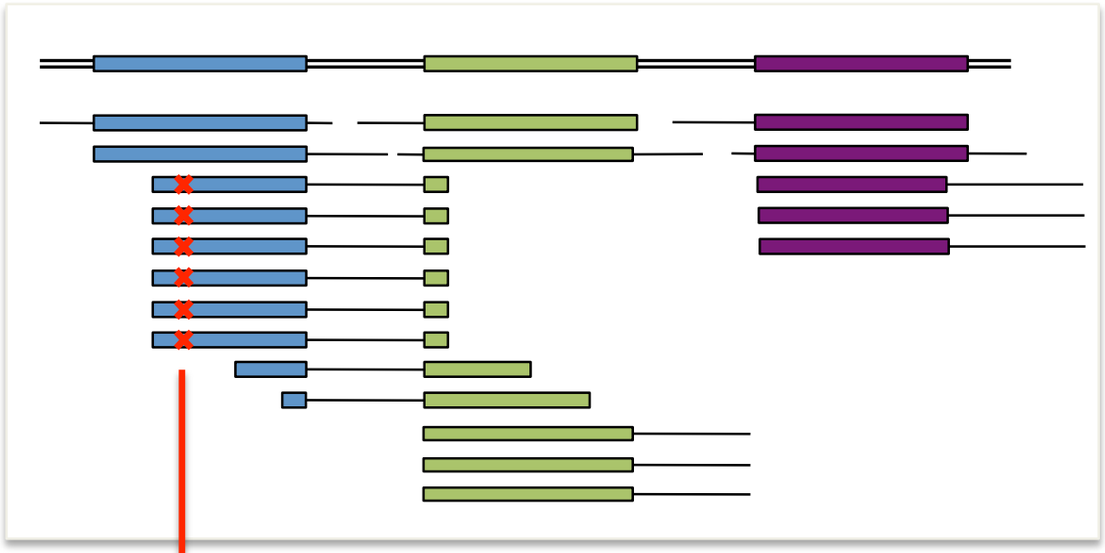
0x400 flag DT:LB

0x400 flag DT:LB

http://core-genomics.blogspot.fi/2016/01/almost-everything-you-wanted-to-know.html

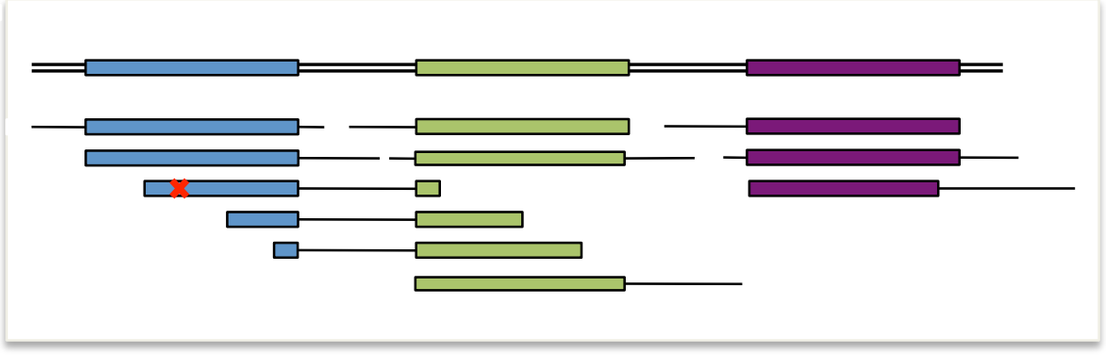
The reason why duplicates are bad

✖ = sequencing error propagated in duplicates



FP variant call (bad)

After marking duplicates, the GATK will only see :

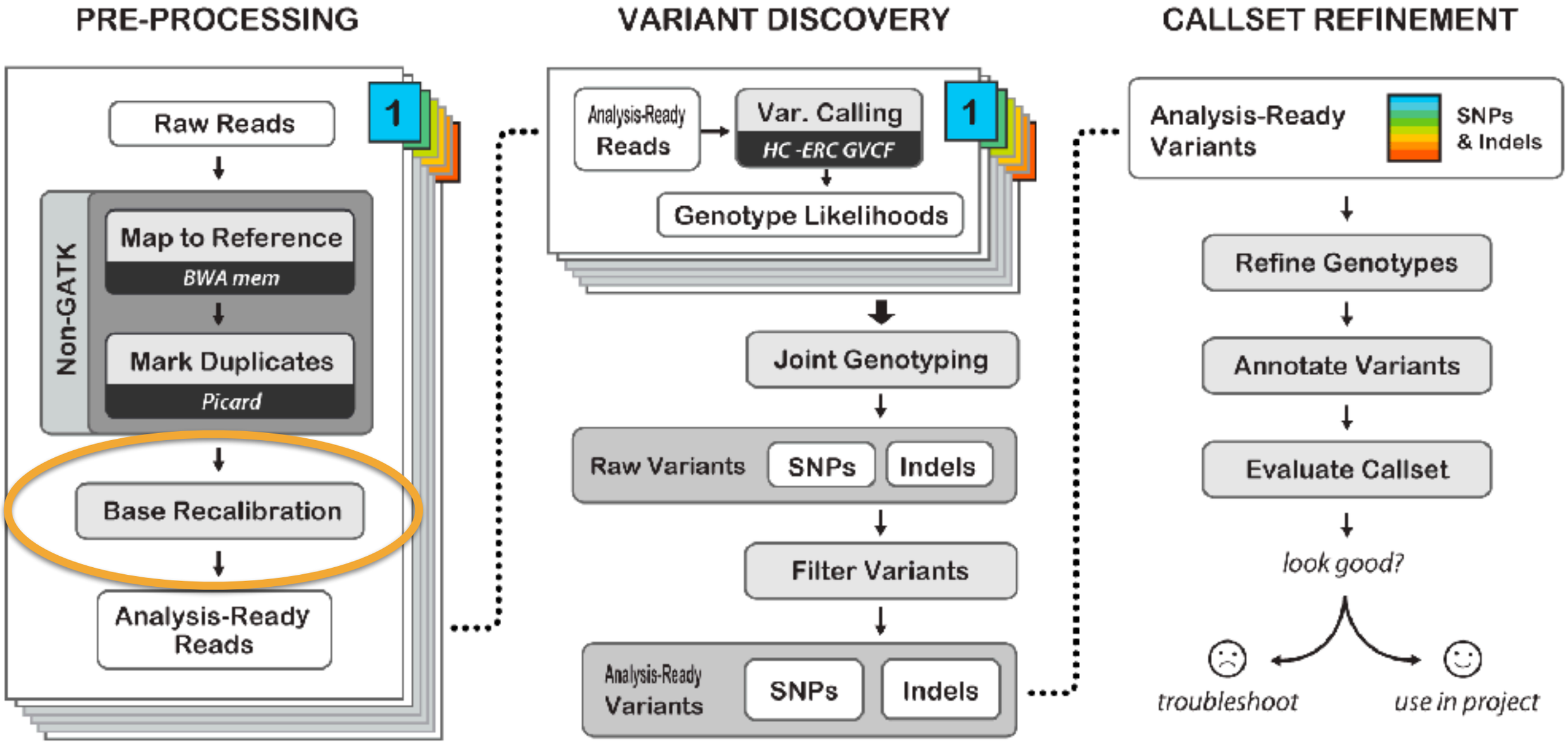


… and thus be more likely to make the right call

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

Why wouldn't we do this for GBS?

Base recalibration

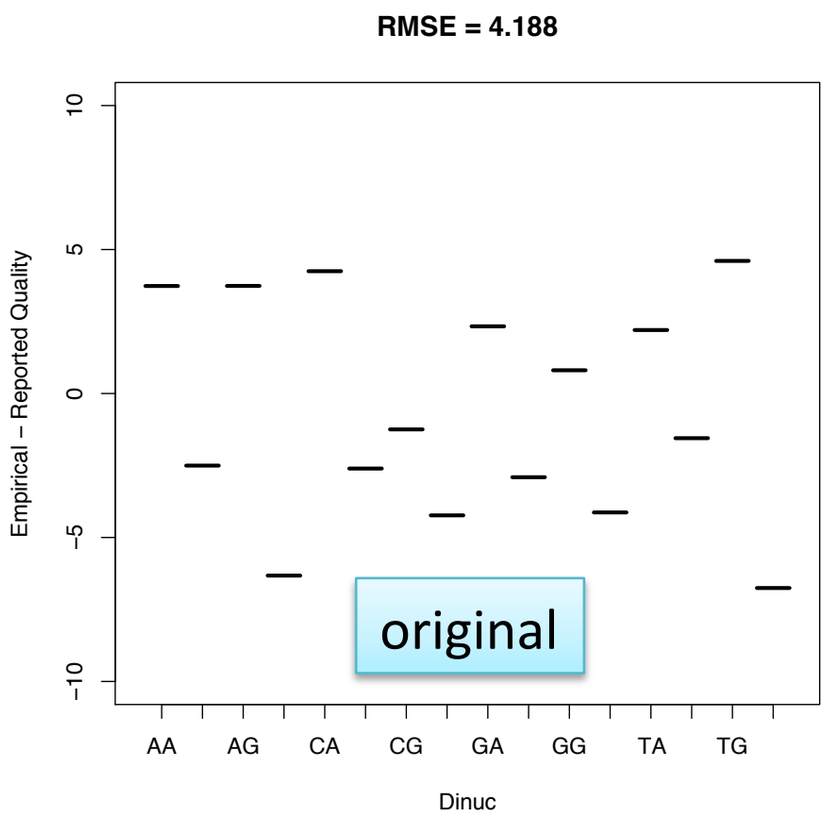


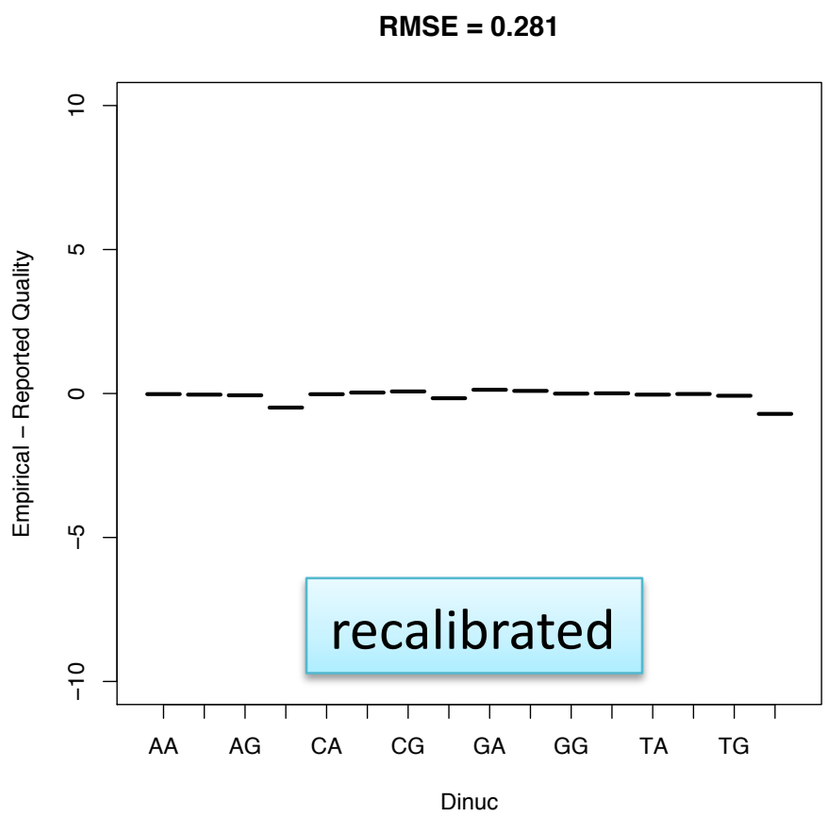
Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

Quality scores issued by sequencers are **inaccurate** and **biased**

- Quality scores are critical for all downstream analysis
- Systematic biases are a major contributor to bad calls

Example of bias: qualities reported depending on nucleotide context





BQSR identifies patterns in how errors correlate with base features

- Empowered by looking at entire lane of data
- Analyze covariation among several features of a base, e.g.:
  - Reported quality score
  - Position within the read (machine cycle)
  - Preceding and current nucleotide (sequencing chemistry effect)
- Based on the patterns identified:

  Apply corrections to recalibrate the quality scores of all reads in the BAM file.

How covariates are analyzed to identify patterns

- Any sequence mismatch = error *except known variants!*

- Keep track of number of observations and number of errors as a function of various error covariates

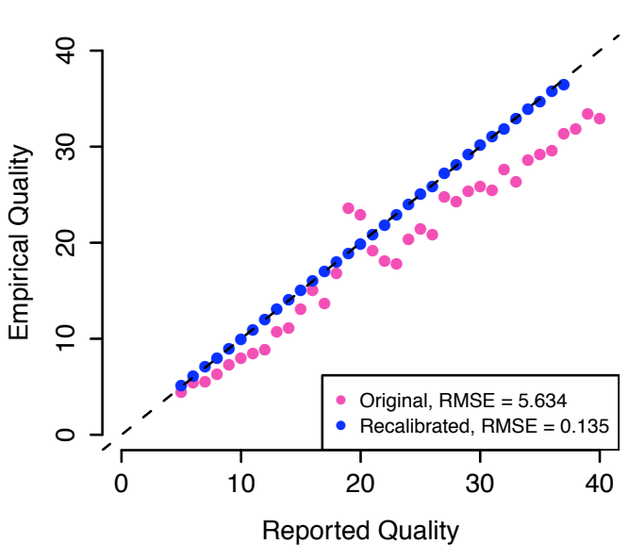
(lane, original quality score, machine cycle, and sequencing context)

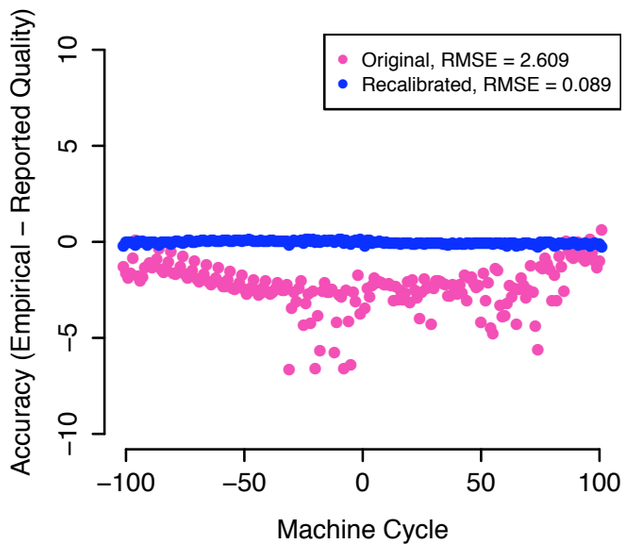
$$\frac{\#\text{ of reference mismatches} + 1}{\#\text{ of observed bases} + 2}$$

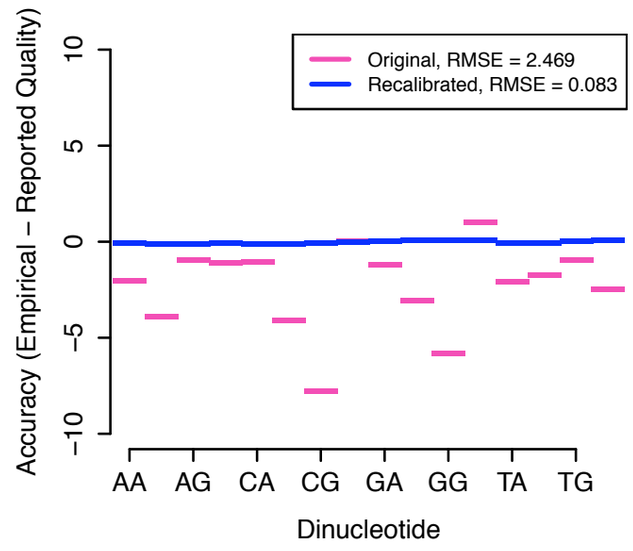
PHRED-scaled quality score

Did the recalibration work properly?

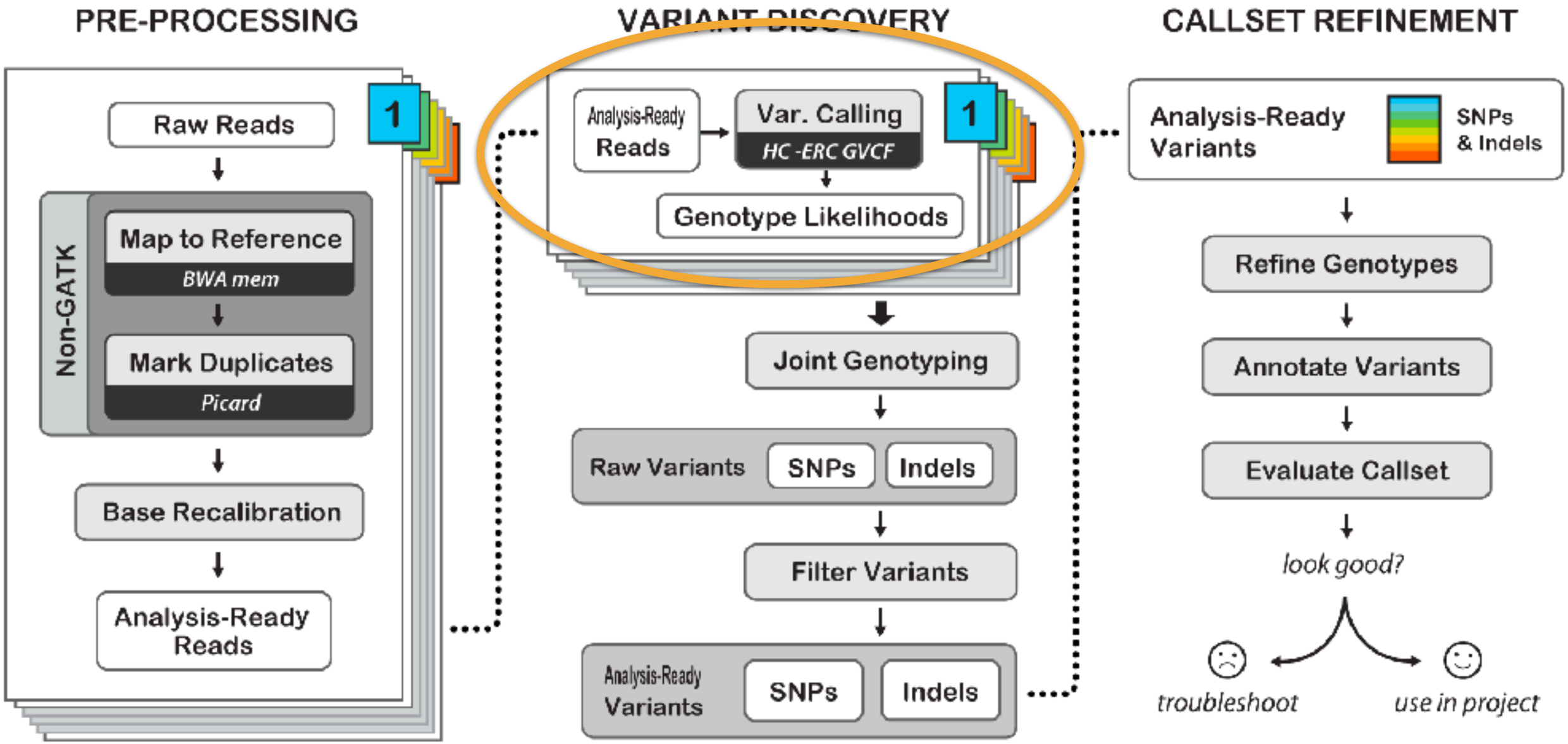
Post-recalibration quality scores should fit the empirically-derived quality scores very well; no obvious systematic biases should remain







Base recalibration



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

Real mutations are hidden in the noise

Summed up in GATK terms

Prior of the genotype

Likelihood of the genotype

Bayesian model

$$\Pr\{G|D\} = \frac{\Pr\{G\}\Pr\{D|G\}}{\sum_i \Pr\{G_i\}\Pr\{D|G_i\}}, \text{ [Bayes' rule]}$$

Diploid assumption

$$\Pr\{D|G\} = \prod_j \left(\frac{\Pr\{D_j|H_1\}}{2} + \frac{\Pr\{D_j|H_2\}}{2} \right) \text{ where } G = H_1H_2$$

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

Variant callers in GATK

- UnifiedGenotyper

  Call SNPs and indels separately by considering each variant locus independently

  - Accepts any ploidy
  - Pooled calling

- HaplotypeCaller

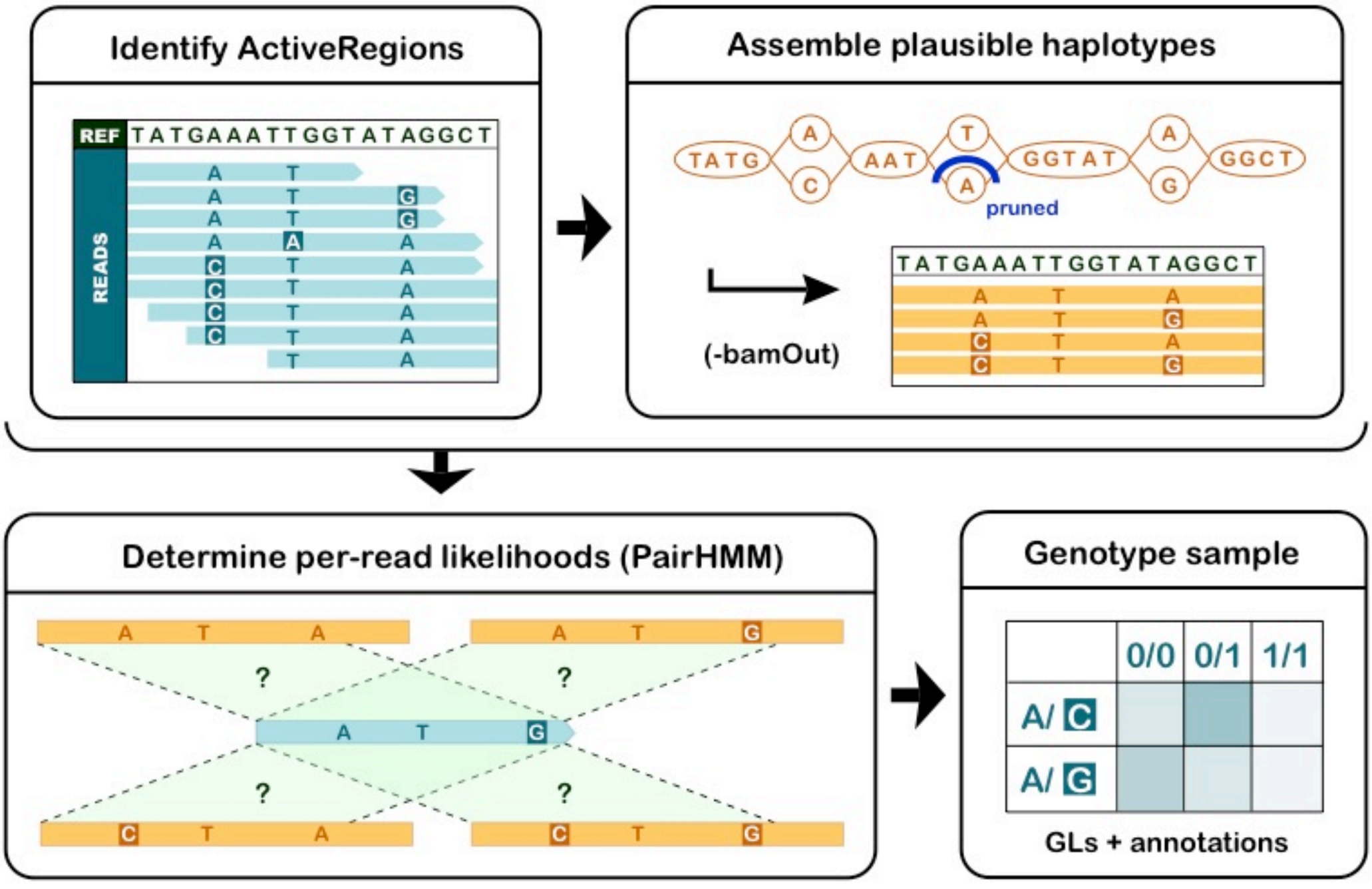
  Call SNPs, indels, and some SVs simultaneously by doing local re-assembly and considering haplotypes

  - More accurate (esp. indels)
  - Reference confidence model
  - Replaces UG

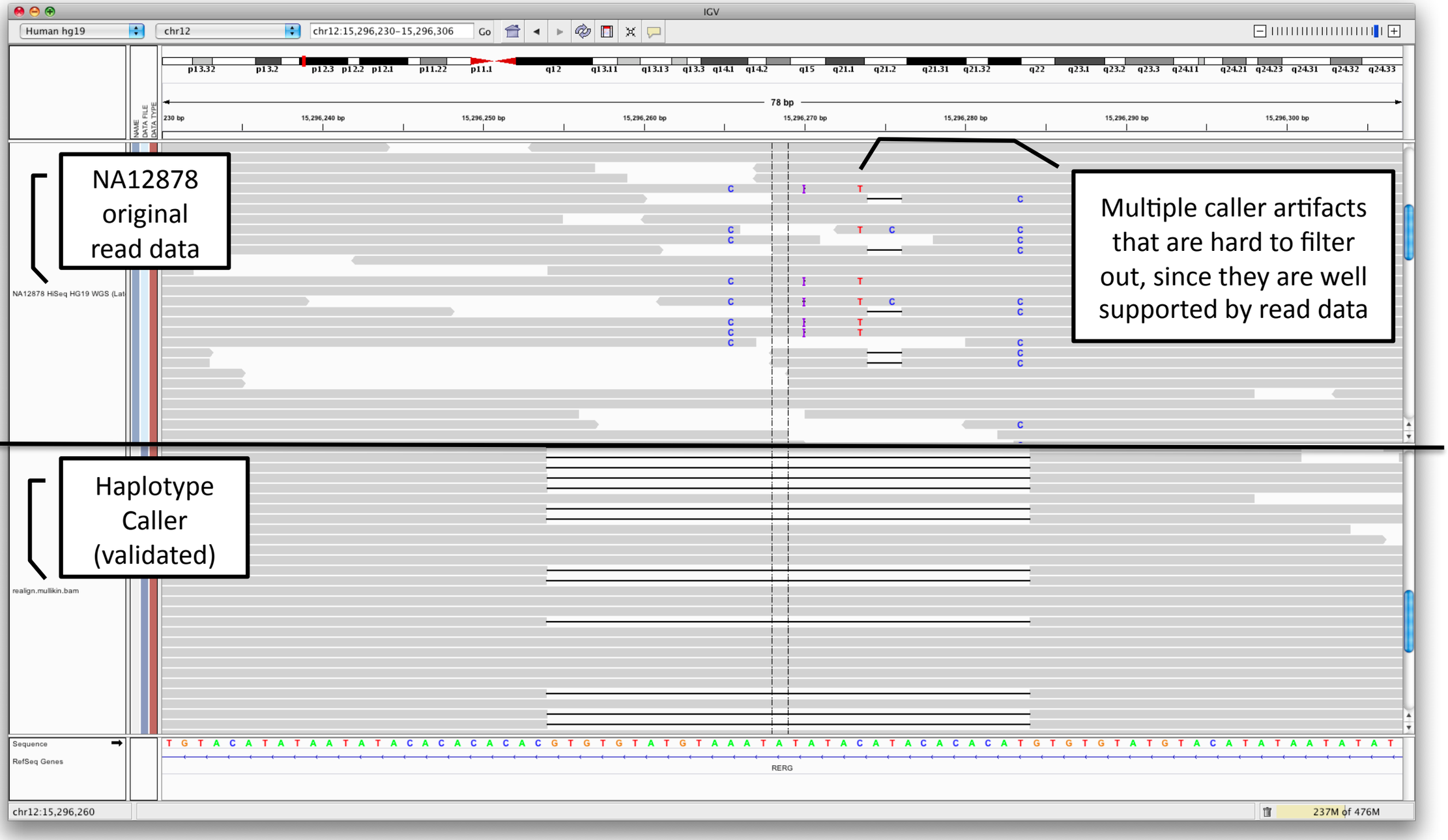
HaplotypeCaller method overview

- 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

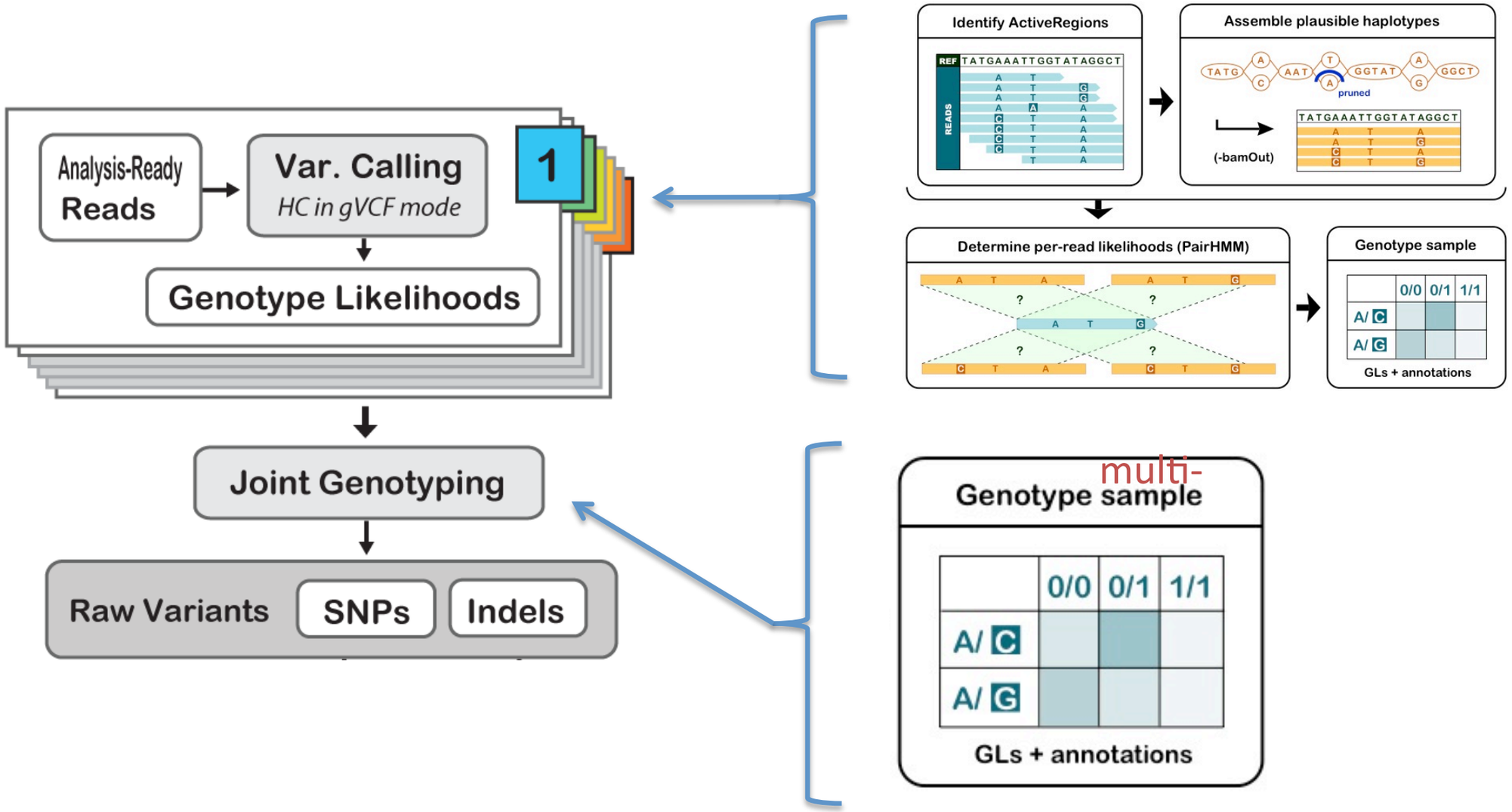


Artifactual SNPs and small indels caused by large indel can be recovered by assembly

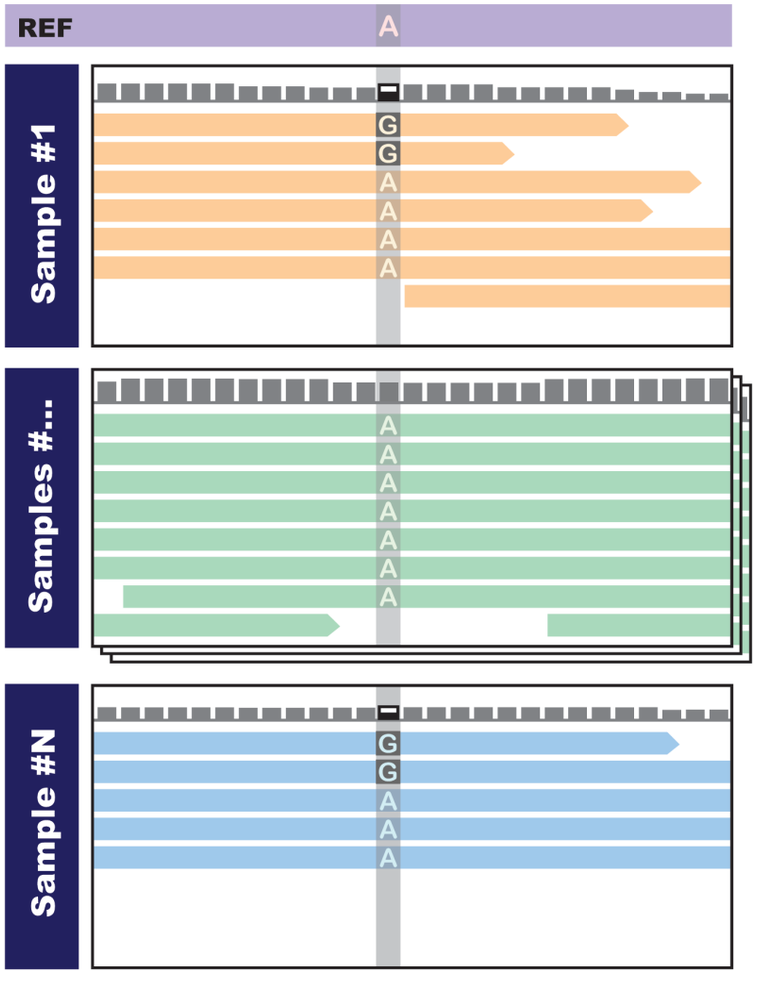


Joint Genotyping

Add a joint analysis step to take advantage of cohort / pop genetics data



Joint discovery empowers discovery at difficult sites

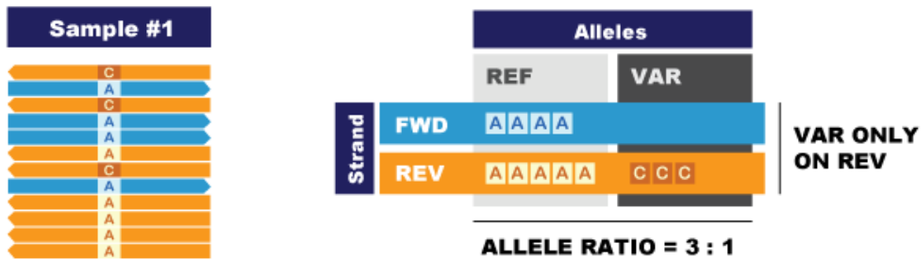


- If we analyze Sample #1 or Sample #N alone we are not confident that the variant is real

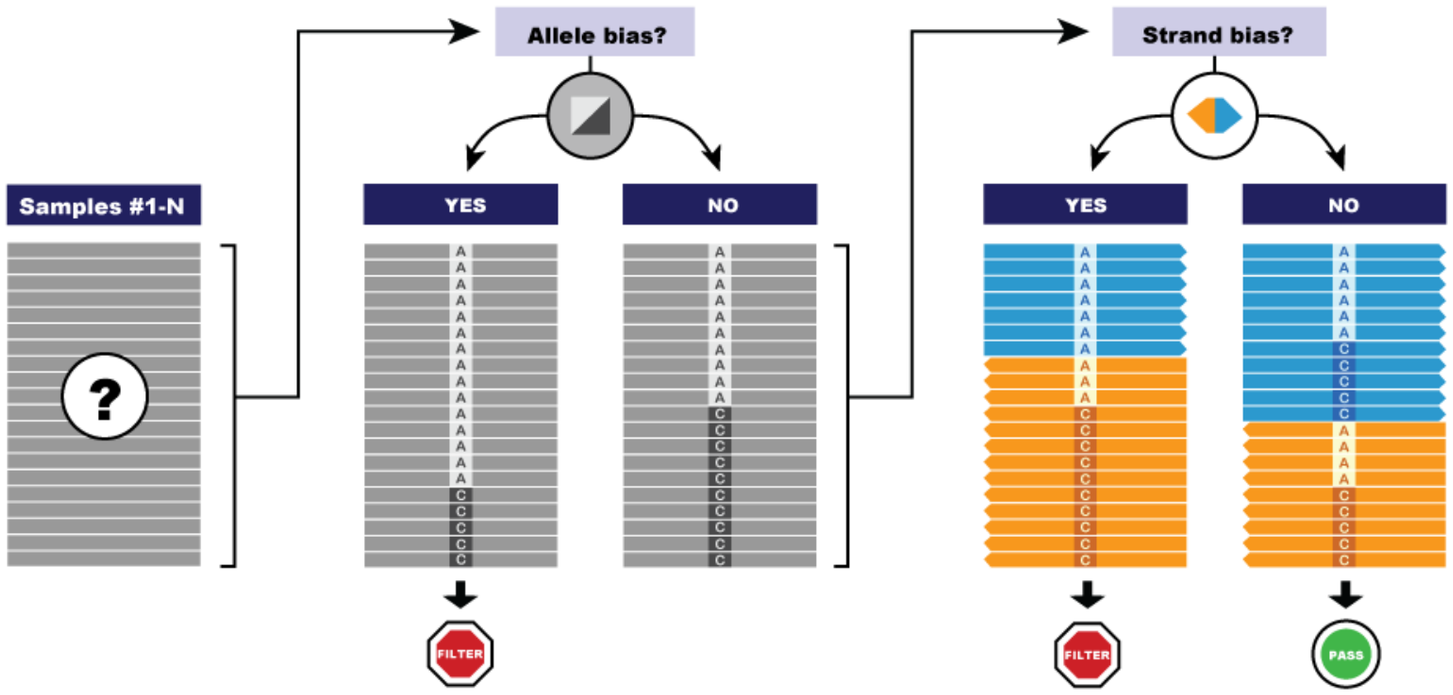
- If we see both samples then we are more confident that there is real variation at this site in the cohort

Joint discovery helps resolve bias issues

A. Single sample showing strand and allelic biases

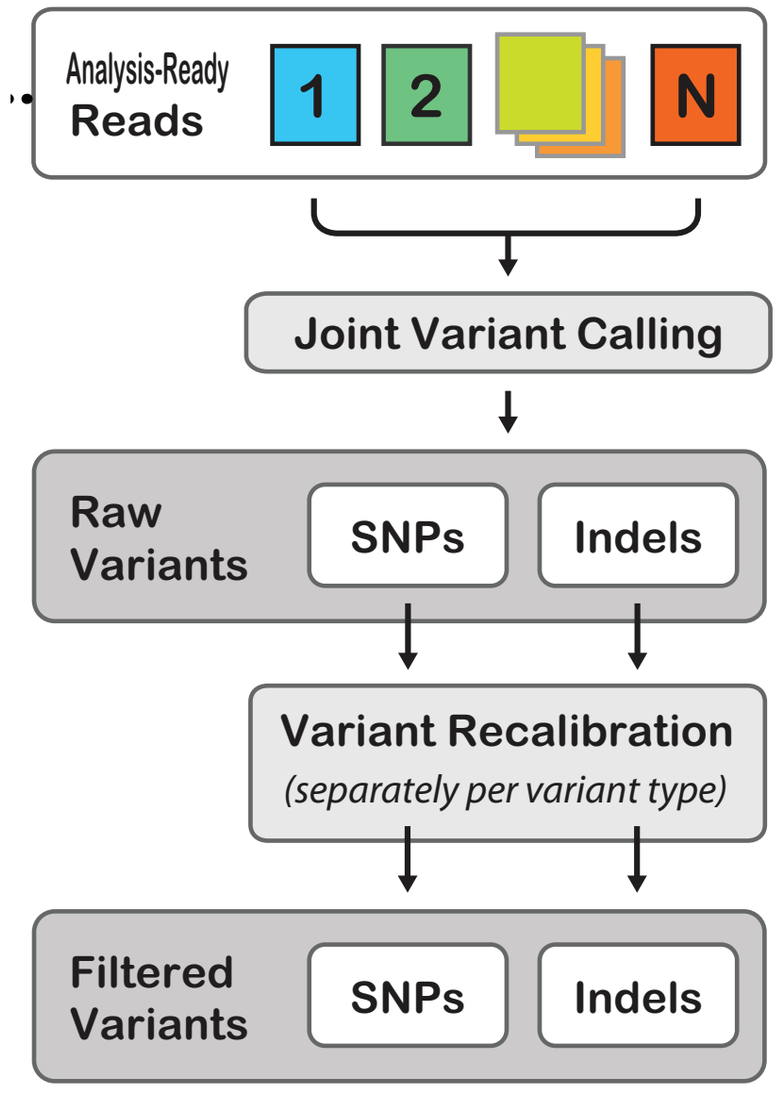


B. Decision process using evidence from multiple samples to filter out sites showing systematic biases



Classic approach to multi-sample variant discovery

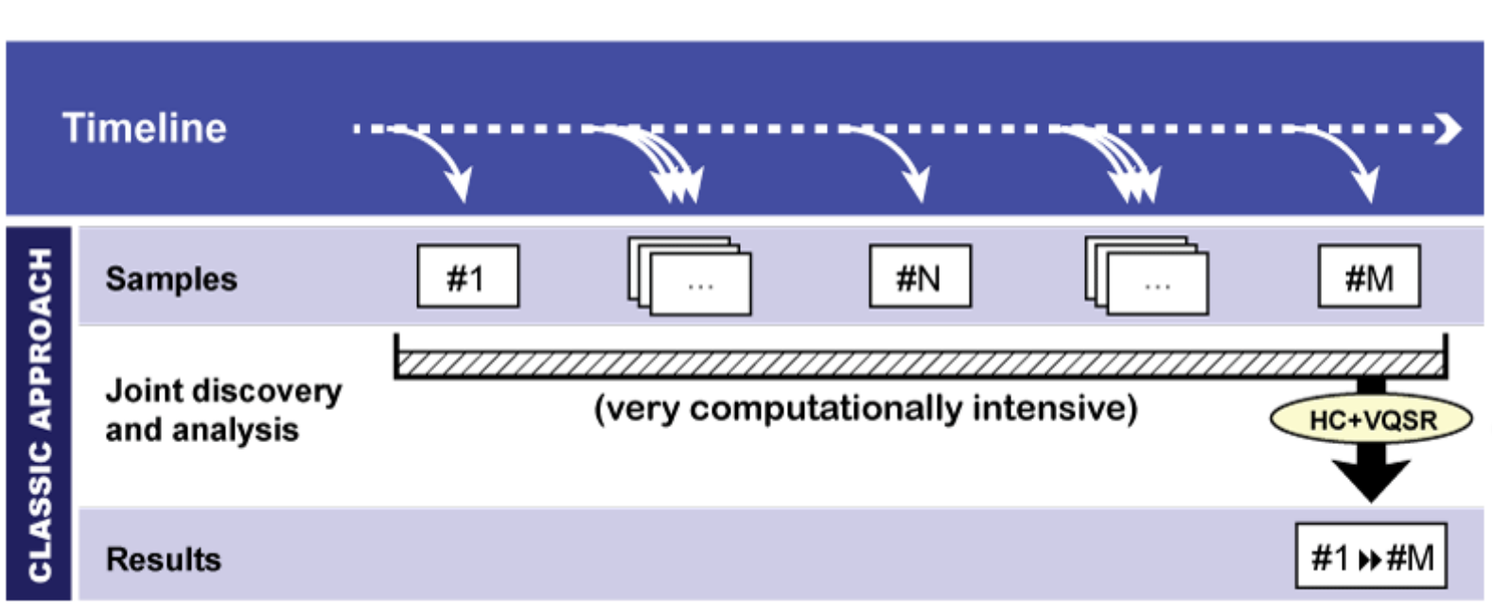
- Call variants jointly on all sample data
  - **Scales badly** -> limitations in amount of data that can be processed
  - Slow with **UnifiedGenotyper** (per-locus calculations)
  - *Impossibly* slow with **HaplotypeCaller** (so much extra work!)

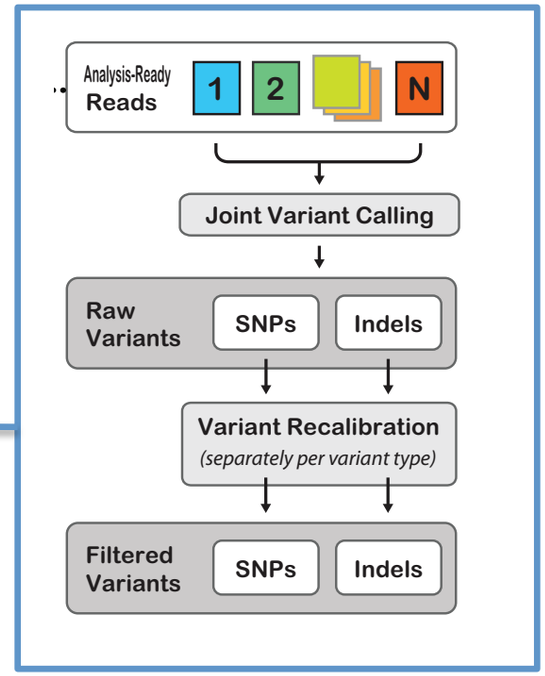


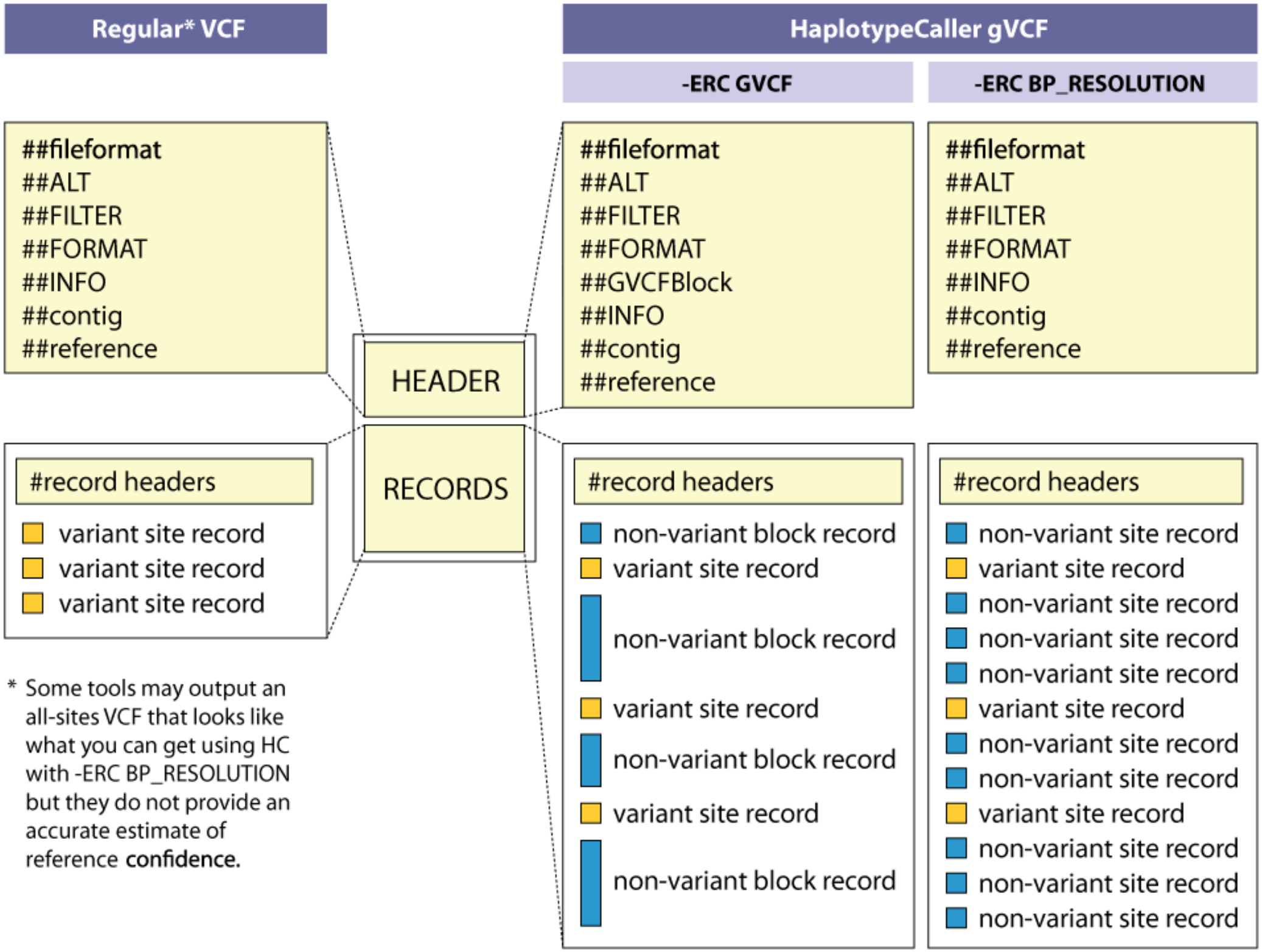
But we want to use HaplotypeCaller because it is so much better!

Problems with the “all together” approach

- Computing costs
- The “N+1 problem”

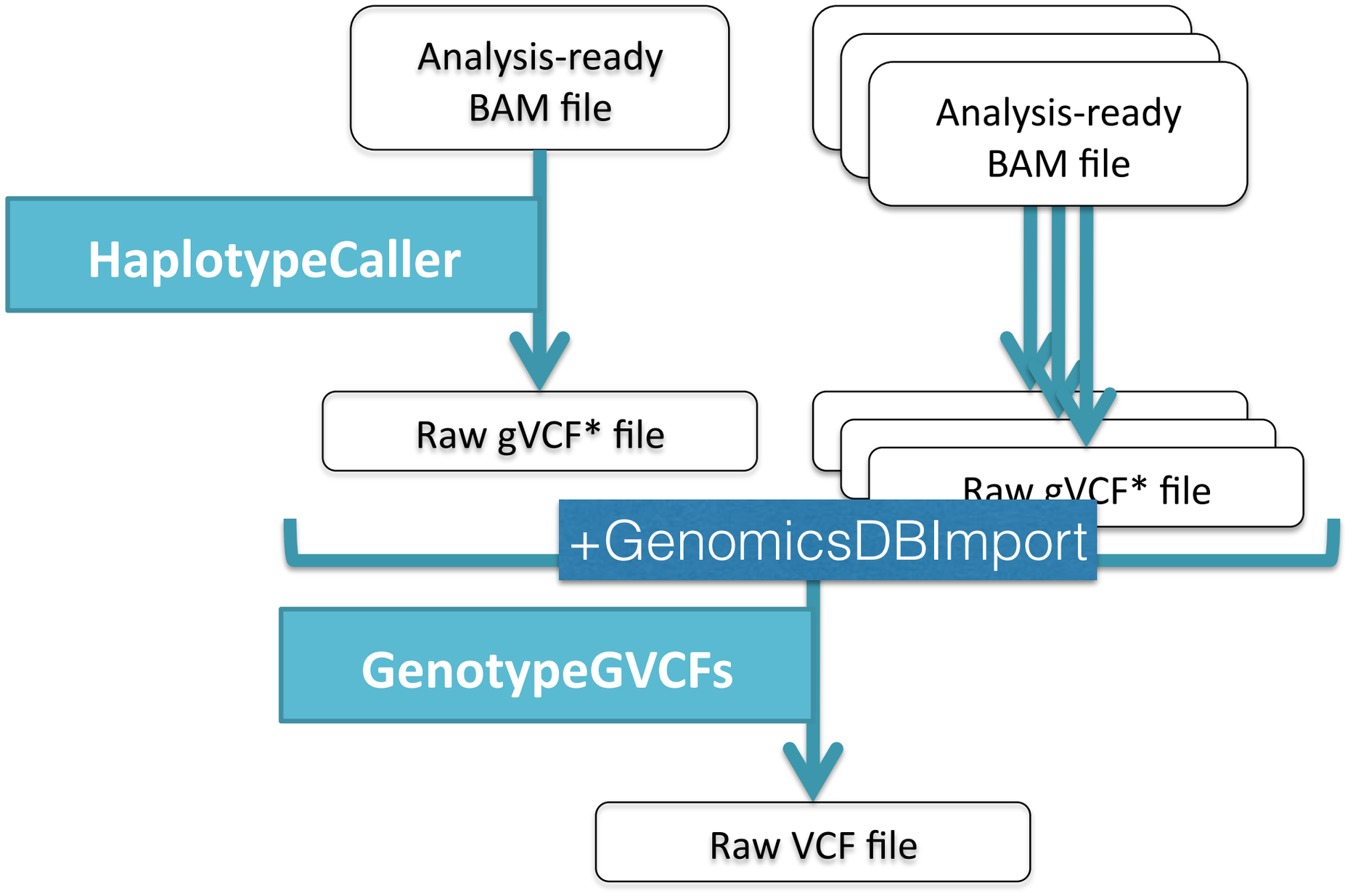






* Some tools may output an all-sites VCF that looks like what you can get using HC with -ERC BP_RESOLUTION but they do not provide an accurate estimate of reference **confidence.**

Variant calling + joint genotyping workflow



Variant annotations provide key information to identify and remove artifacts!

VCF record for an A/G SNP at 22:49582364

```
22 49582364    .    A    G    198.96    .
   AC=3;
   AF=0.50;
   AN=6;
   DP=87;
   MLEAC=3;
   MLEAF=0.50;
   MQ=71.31;
   MQ0=22;
   QD=2.29;
   SB=-31.76
   GT:DP:GQ    0/1:12:99.00    0/1:11:89.43    0/1:28:37.78
```

INFO field

AC	No. chromosomes carrying alt allele	MLEAF	Max likelihood AF
AN	Total no. of chromosomes	MQ	RMS MAPQ of all reads
AF	Allele frequency	MQ0	No. of MAPQ 0 reads at locus
DP	Depth of coverage	QD	QUAL score over depth
MLEAC	Max likelihood AC	SB	Estimated strand bias score

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

Variant Filtering

- By default, GATK is very permissive. It will output false positive sites!
- Two ways of filtering:
  - Hard filters
  - Variant recalibration

Hard Filters

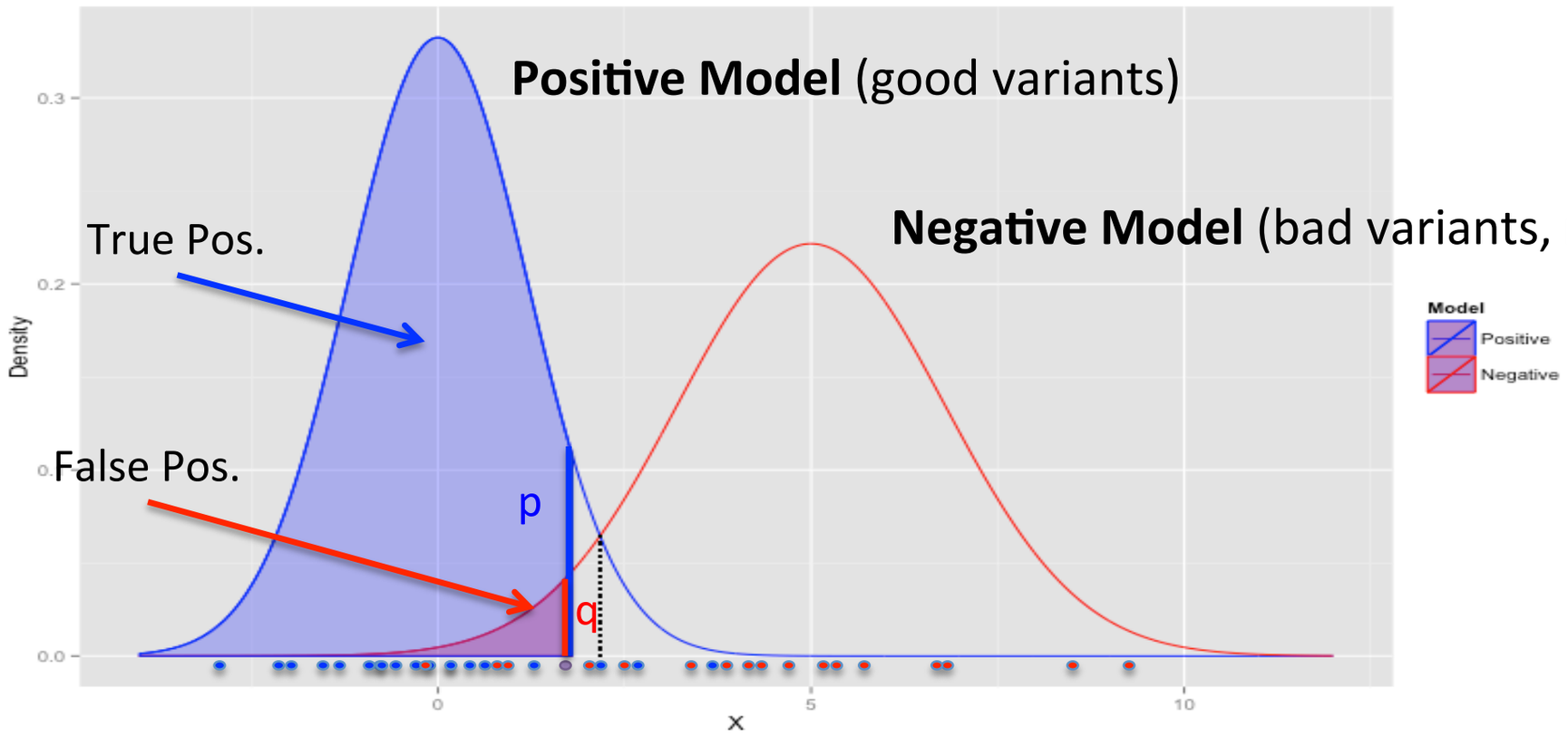
- User defined thresholds for each site. Hard to know where to make cut offs?
  - Mapping quality high enough
  - Depth above a minimum but not too high
  - Minor allele frequency above a minimum
  - Heterozygosity not too high

How variant recalibration works

Train on high-confidence known sites to determine the probability that other sites are true or false

- Assume annotations tend to form **Gaussian clusters**
- Build a "Gaussian mixture model" from annotations of **known variants** in our dataset
- Score **all variants** by where their annotations lie relative to these clusters
- Filter base on **sensitivity to truth set**

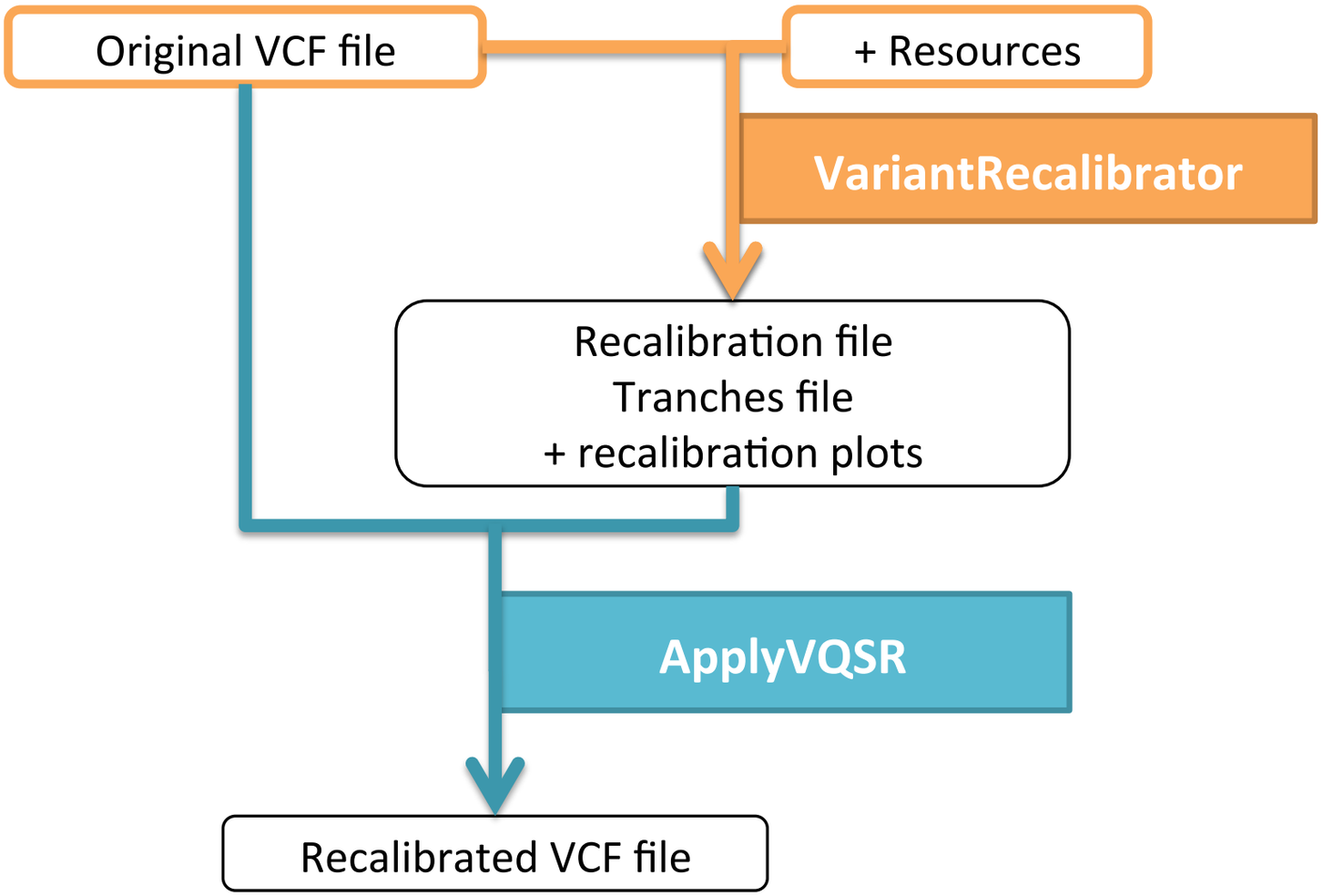
Actually two models: positive and negative



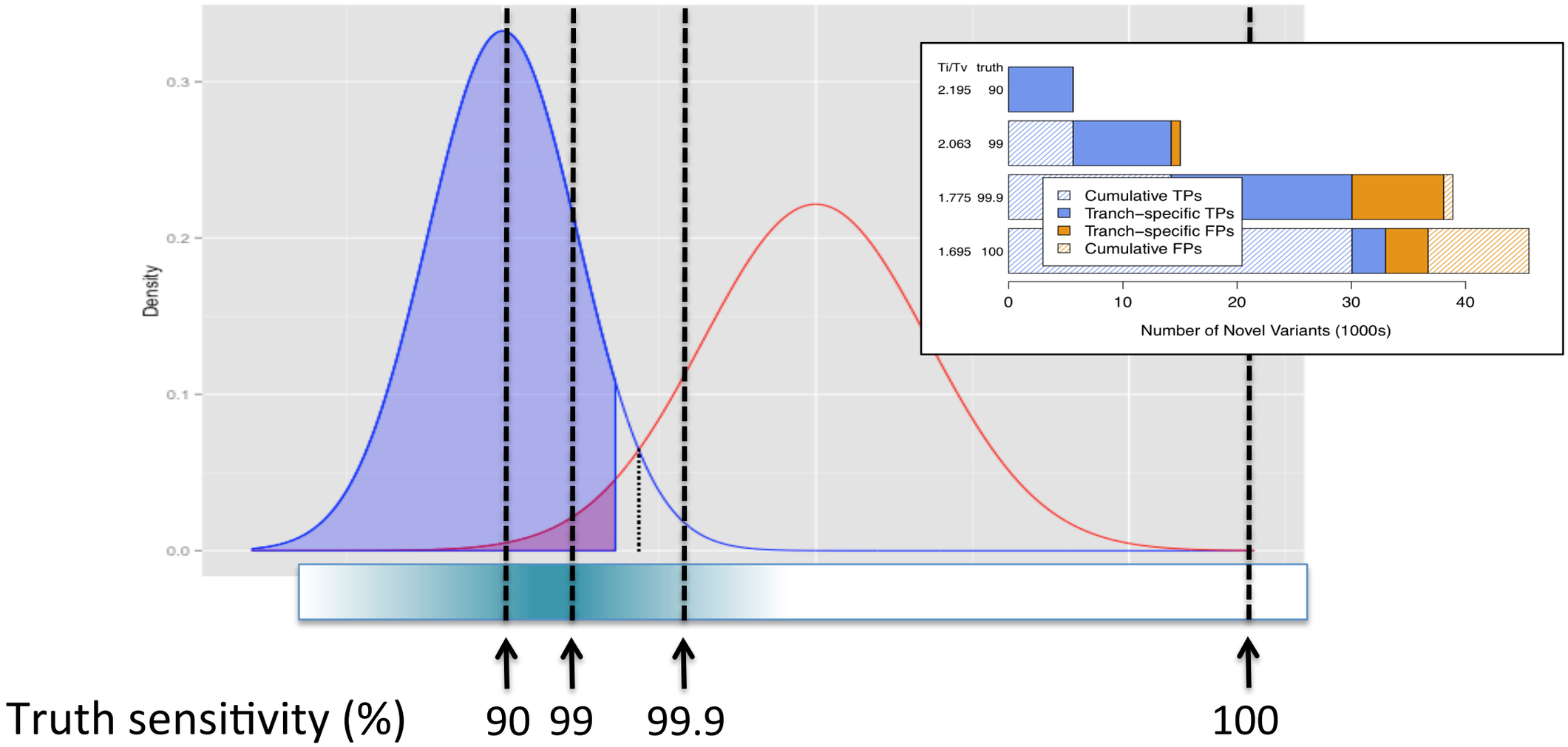
VQSLOD(x) = Log(p(x)/q(x))

Done for each annotation X
then integrated into single overall VQSLOD

Step 1: VariantRecalibrator



Tranches : slices of sensitivity threshold values



Lower tranche = More stringent filtering

Where to get truth set?

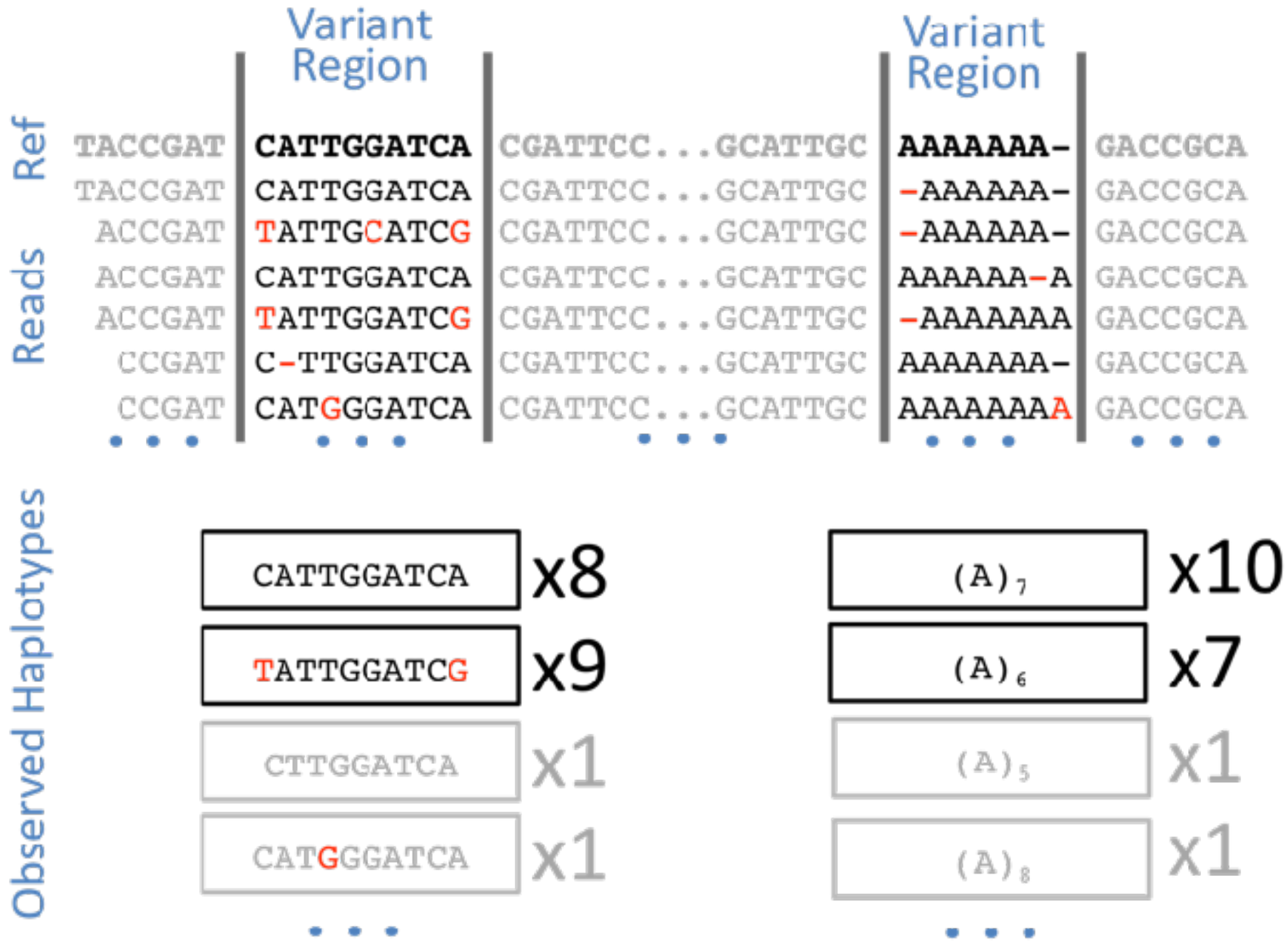
- Reference sets (e.g. 1000 genomes)
- Take your dataset, call SNPs, hard filter heavily, then use those as a truth set for recalibrating the unfiltered dataset.

Alternative Filtering

- Call SNPs using multiple programs and look for variants called in multiple programs, or combine the information in each.

- e.g. BAYSIC

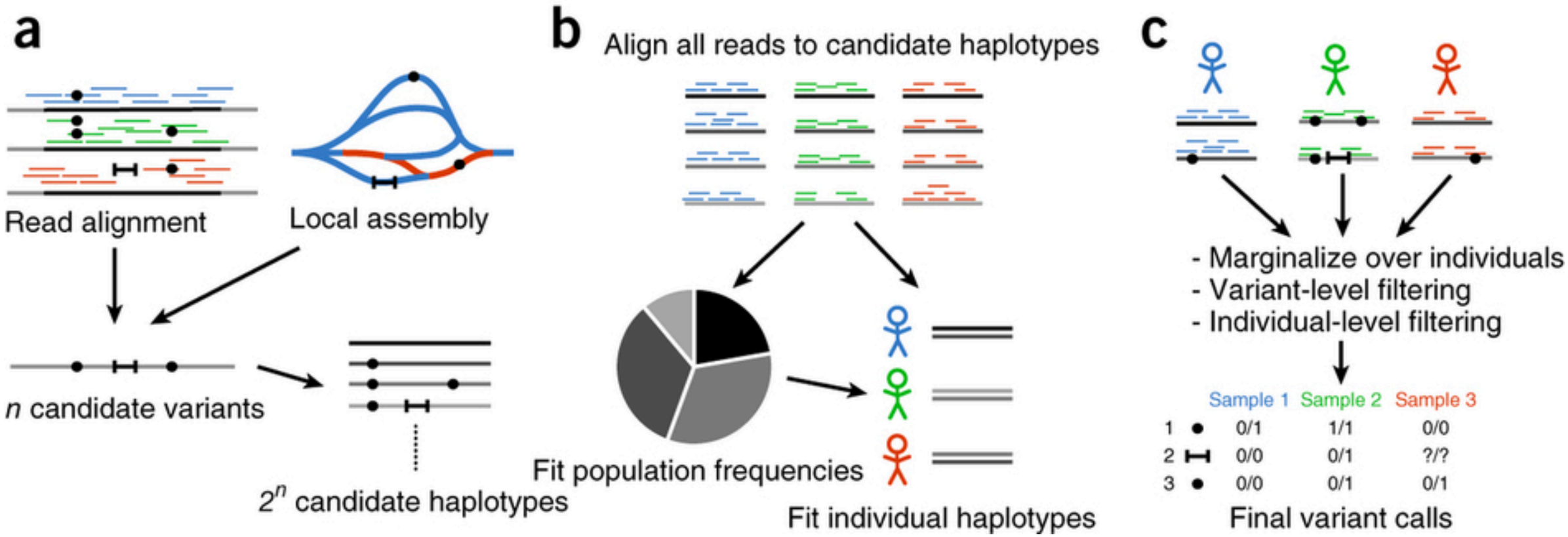
FreeBayes


Variant Region
Variant Region
Ref
Reads
TACCGAT CATTGGATCA CGATTCC...GCATTGC AAAAAAA- GACCGCA
TACCGAT CATTGGATCA CGATTCC...GCATTGC -AAAAAA- GACCGCA
ACCGAT TATTGCATCG CGATTCC...GCATTGC -AAAAAA- GACCGCA
ACCGAT CATTGGATCA CGATTCC...GCATTGC AAAAAA-A GACCGCA
ACCGAT TATTGGATCG CGATTCC...GCATTGC -AAAAAAA GACCGCA
CCGAT C-TTGGATCA CGATTCC...GCATTGC AAAAAAA- GACCGCA
CCGAT CATGGGATCA CGATTCC...GCATTGC AAAAAAAA GACCGCA
Observed Haplotypes
CATTGGATCA x8
TATTGGATCG x9
CTTGGATCA x1
CATGGGATCA x1
(A)7 x10
(A)6 x7
(A)5 x1
(A)8 x1

FreeBayes

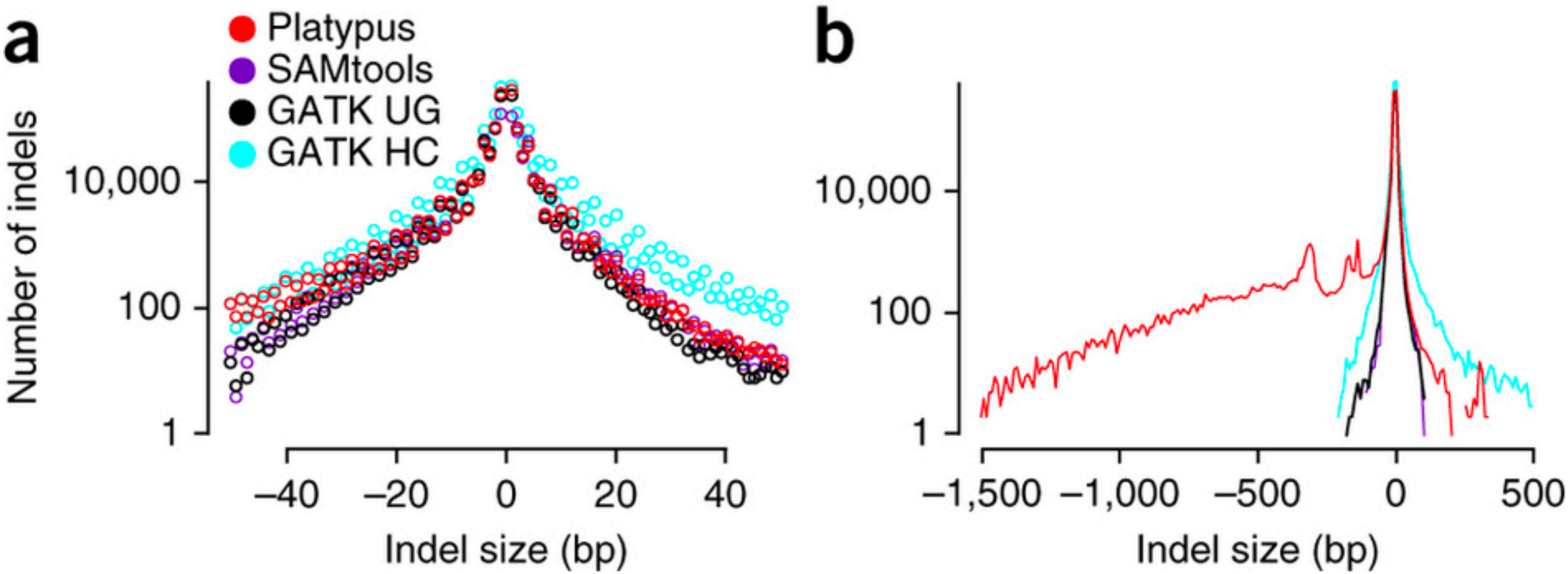
- Free and open source.
- Uses literal sequences of bases and haplotypes to call SNPs so is less affected by local alignment issues.
- Does not have “gvcf” n+1 method, although complicated work around exists.
- Generally faster than GATK, although RAM intensive.

Platypus



Includes local assembly, better for large indels

Platypus



Includes local assembly, better for large indels

ANGSD

- Calls SNPs based on reads per site, no realignment.
- Outputs genotype likelihoods.
- Links with algorithms that use likelihood.
- Questionable with high diversity systems.