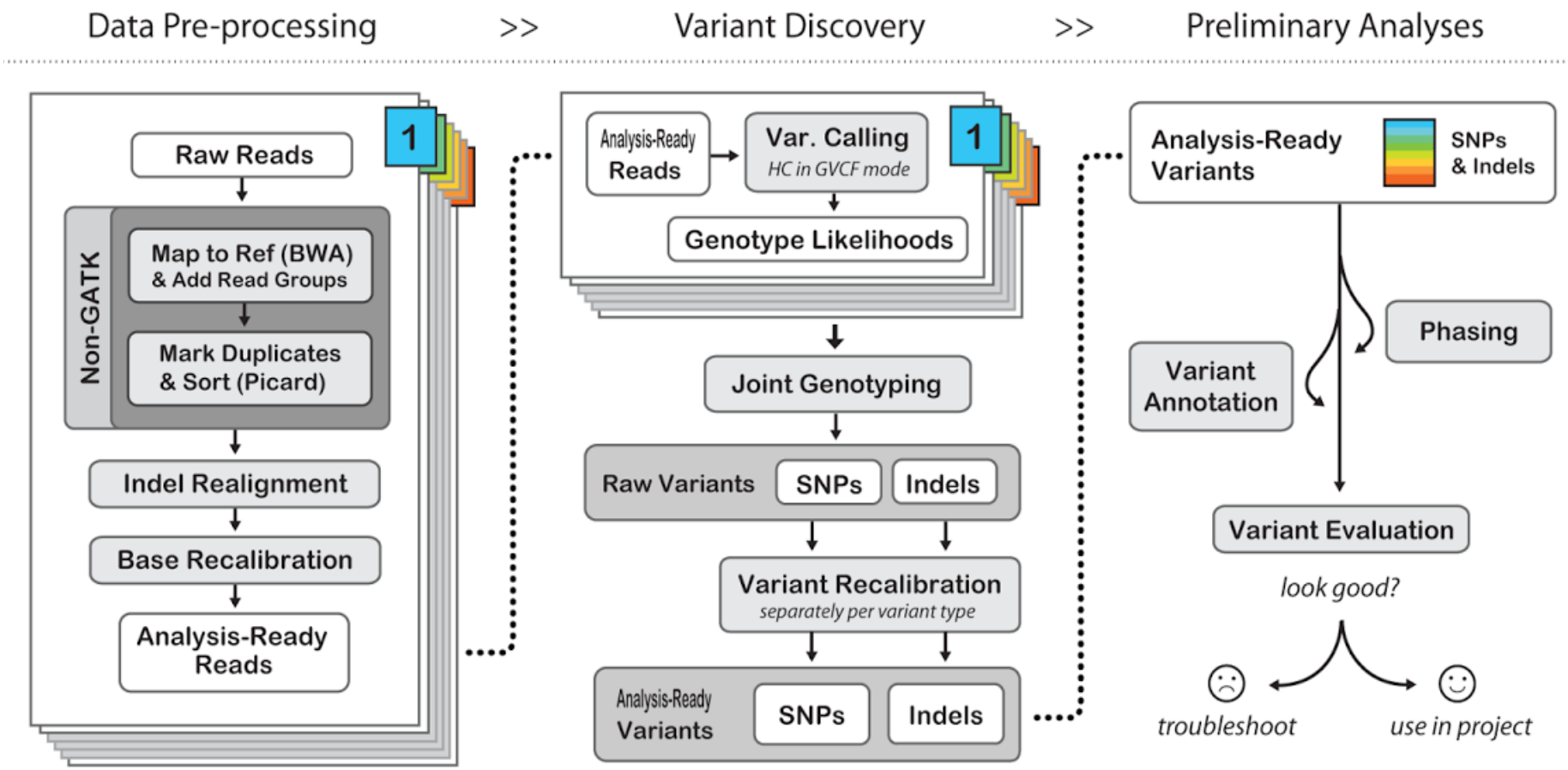
Bioinformatics for Evolutionary Biology

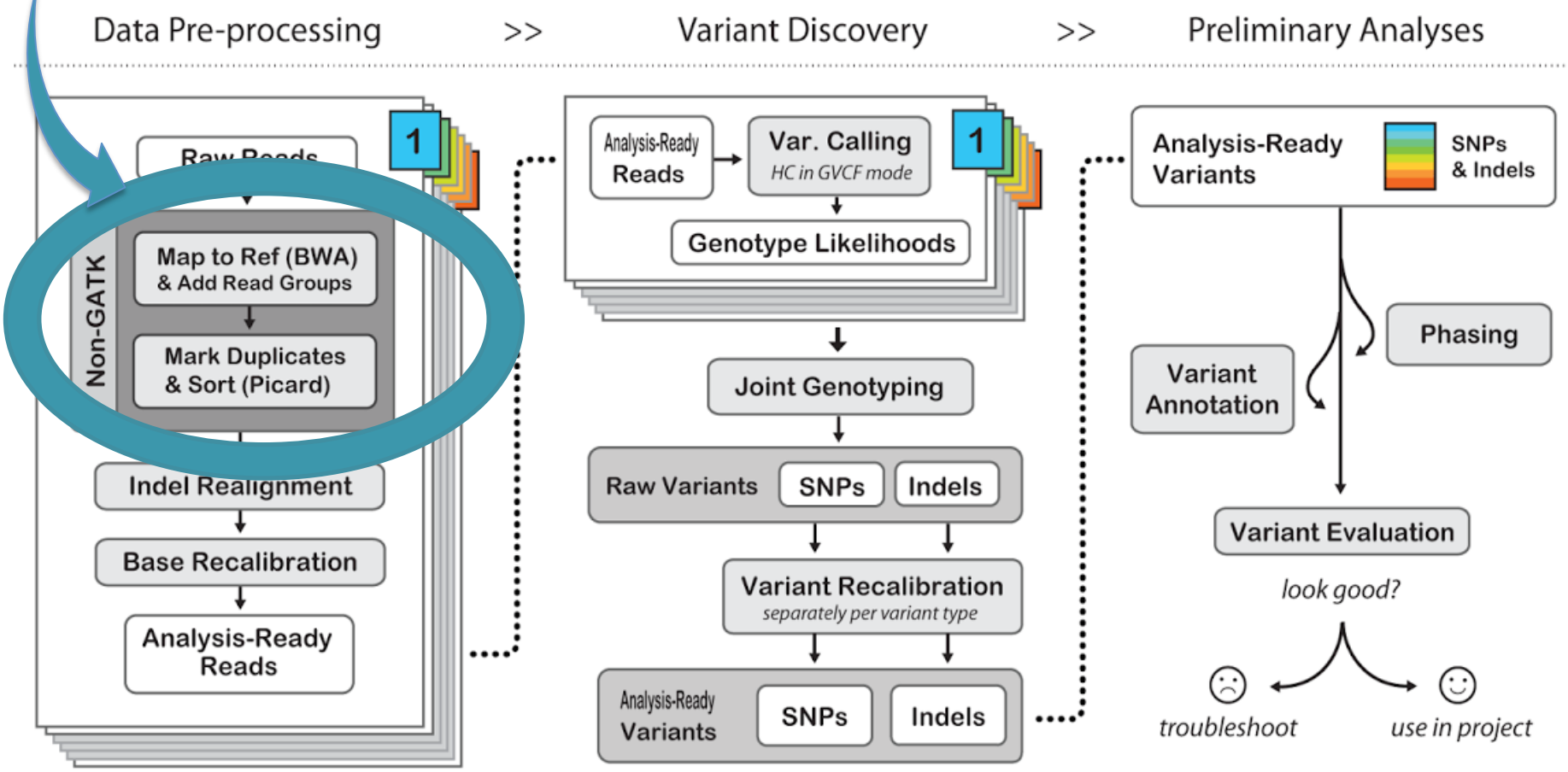
Variant Calling

Best Practices for Variant Discovery in DNAseq

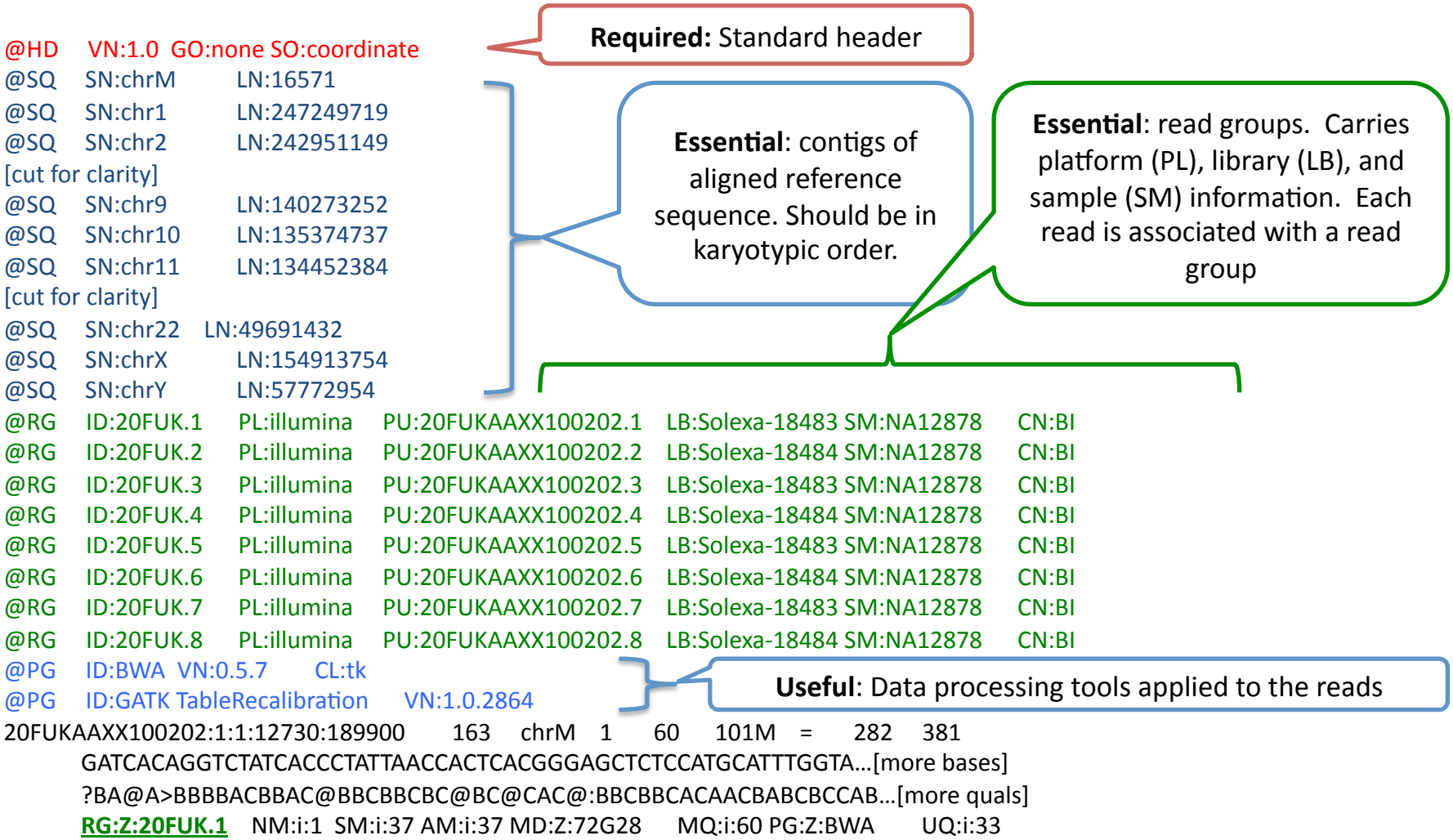


We are here in the Best Practices workflow

Mapping and Marking Duplicates



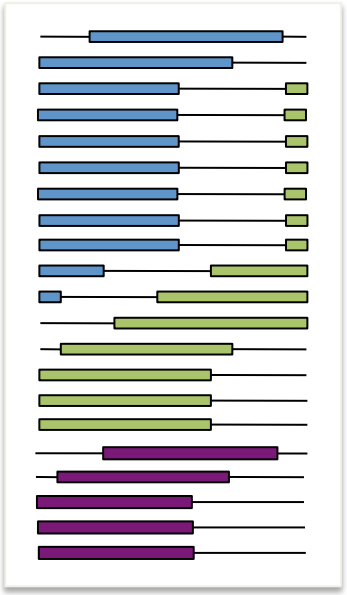
BAM headers: an essential part of a BAM file



Official specification in http://samtools.sourceforge.net/SAM1.pdf

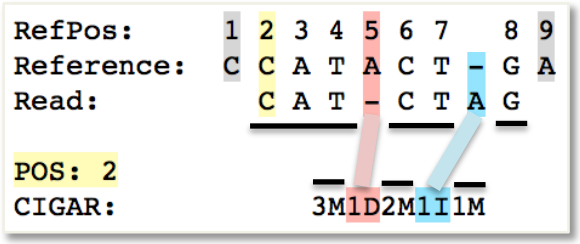
Mapping short reads to a reference is simple in principle

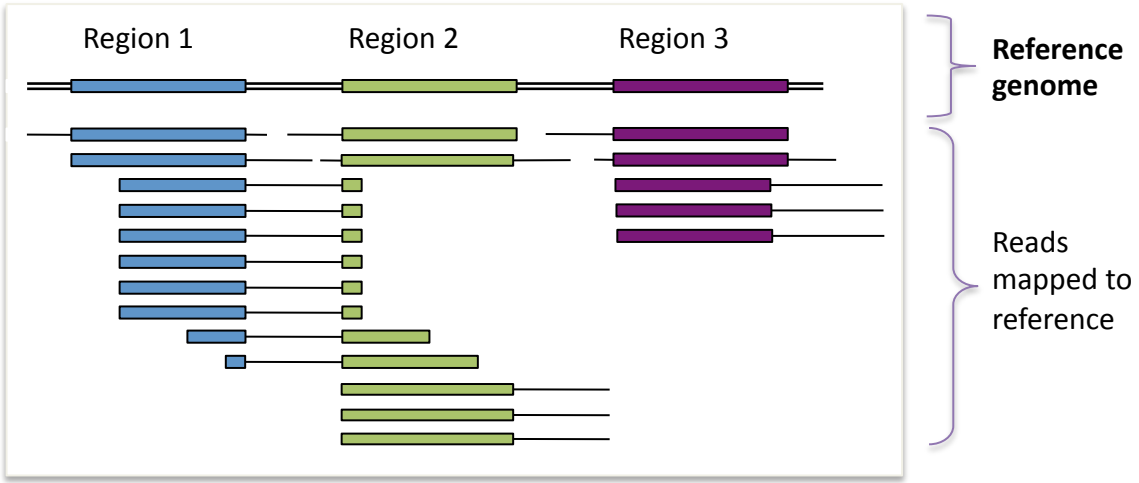
Enormous pile of short reads from NGS



Mapping and alignment algorithms

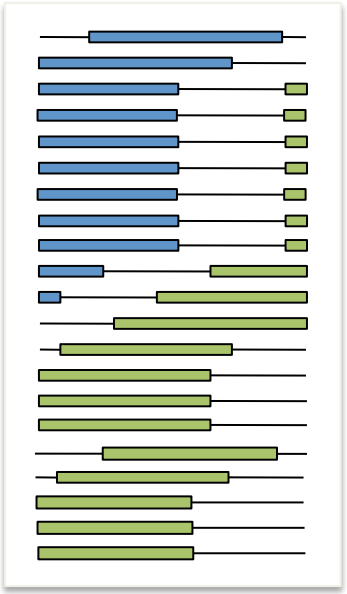
Identify where the read matches the reference sequence and record match details as CIGAR string





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

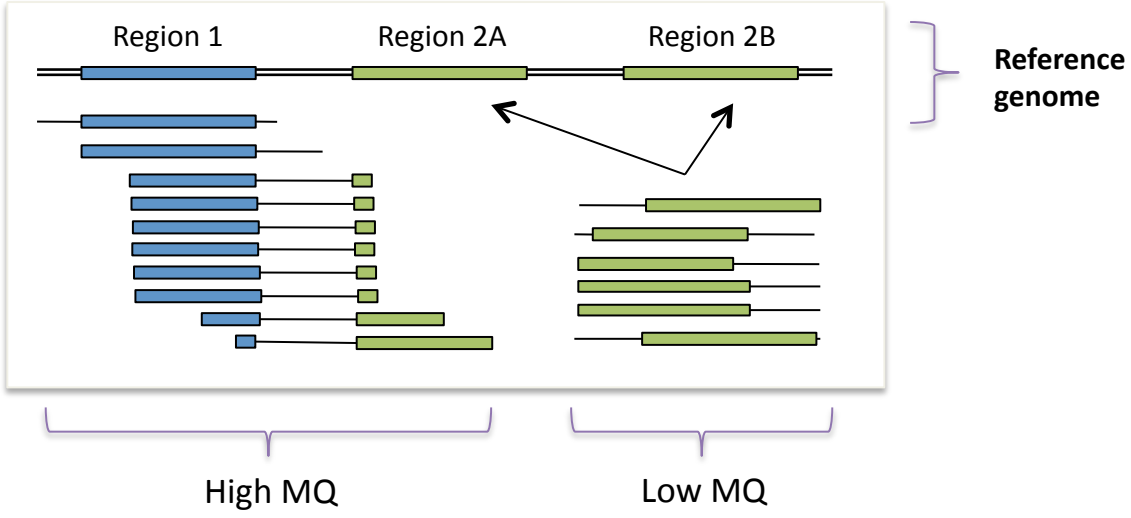
Enormous pile of short reads from NGS



Mapping and alignment algorithms

Mapping algorithms account for this by choosing the most likely placement

➔ **mapping quality (MQ)**

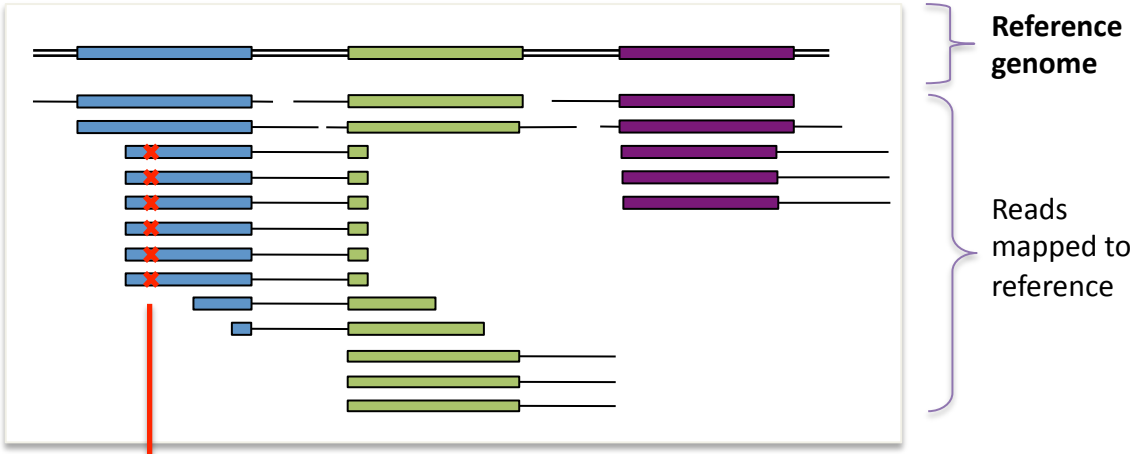


For more information see:

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

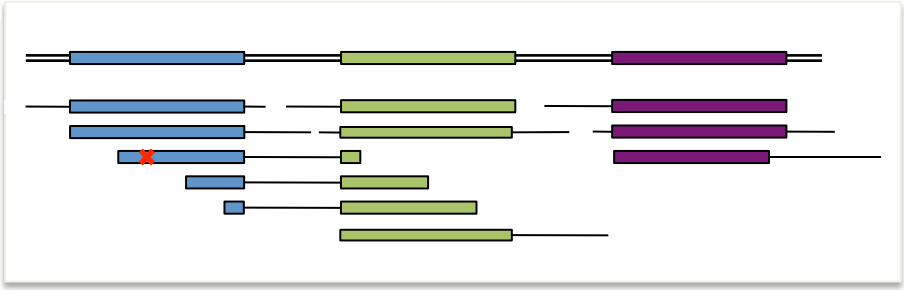
The reason why duplicates are bad

✖ = sequencing error propagated in duplicates



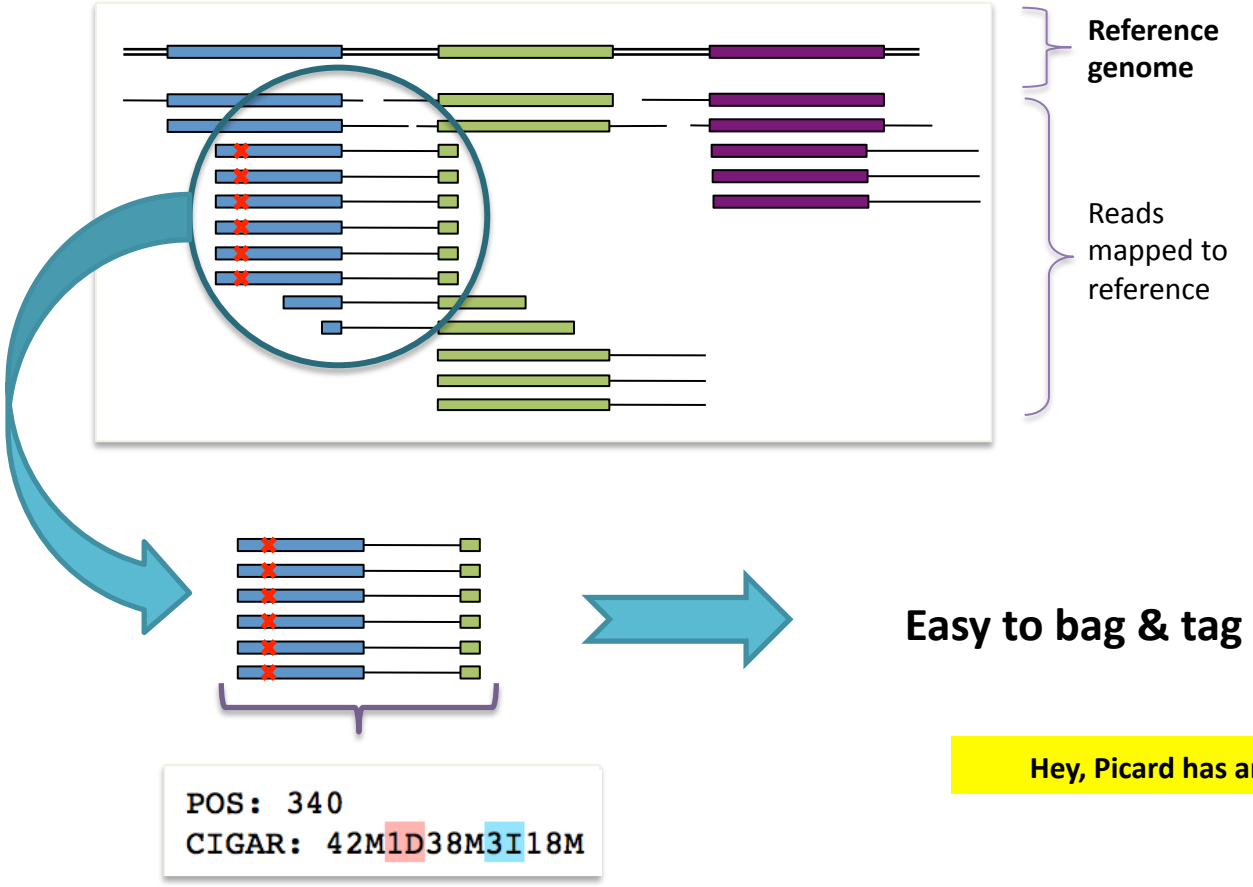
FP variant call (bad)

After marking duplicates, the GATK will only see :



… and thus be more likely to make the right call

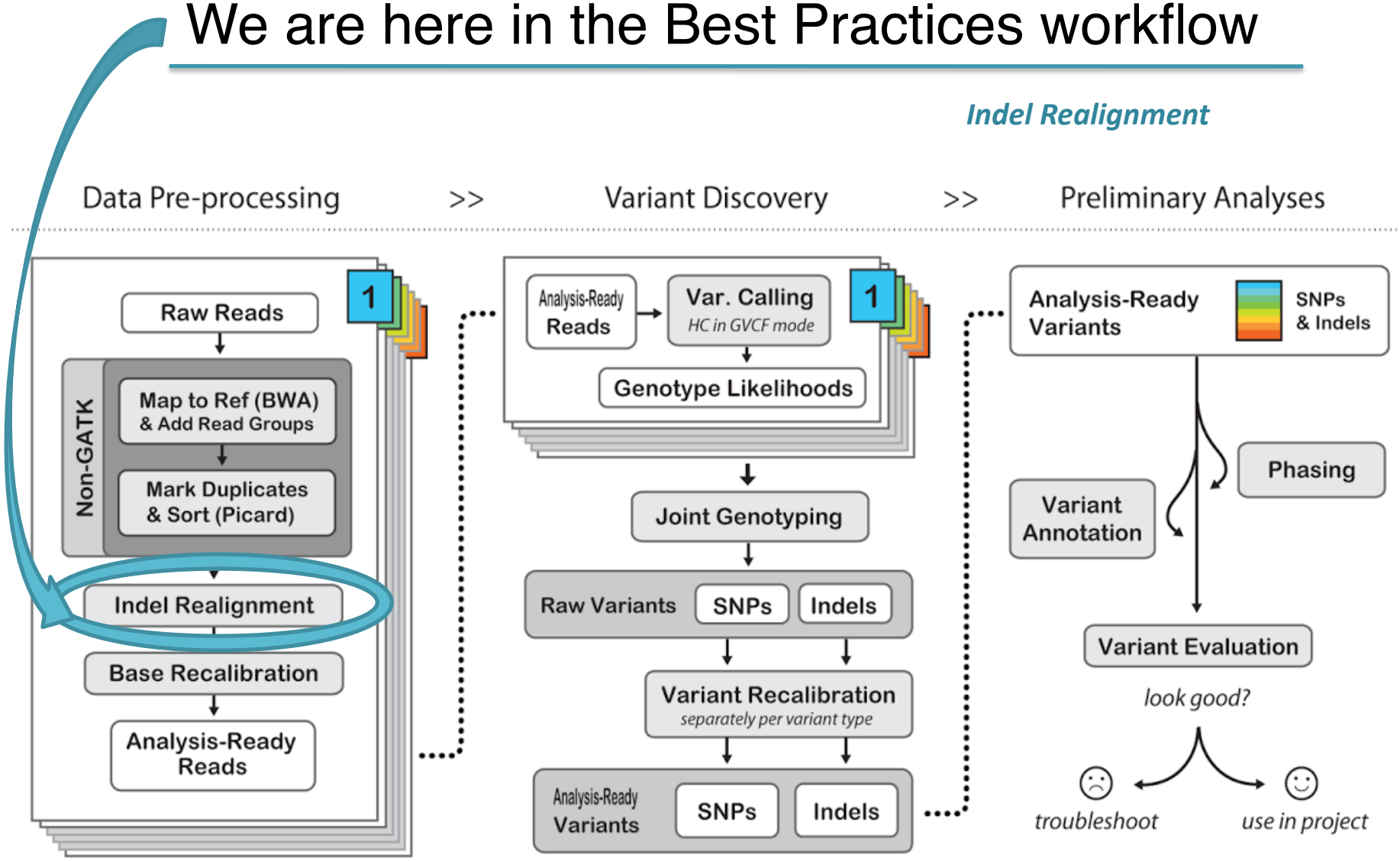
Easy to identify: duplicate reads have the same starting position and same CIGAR string



Easy to bag & tag

Hey, Picard has an app for that!

We are here in the Best Practices workflow

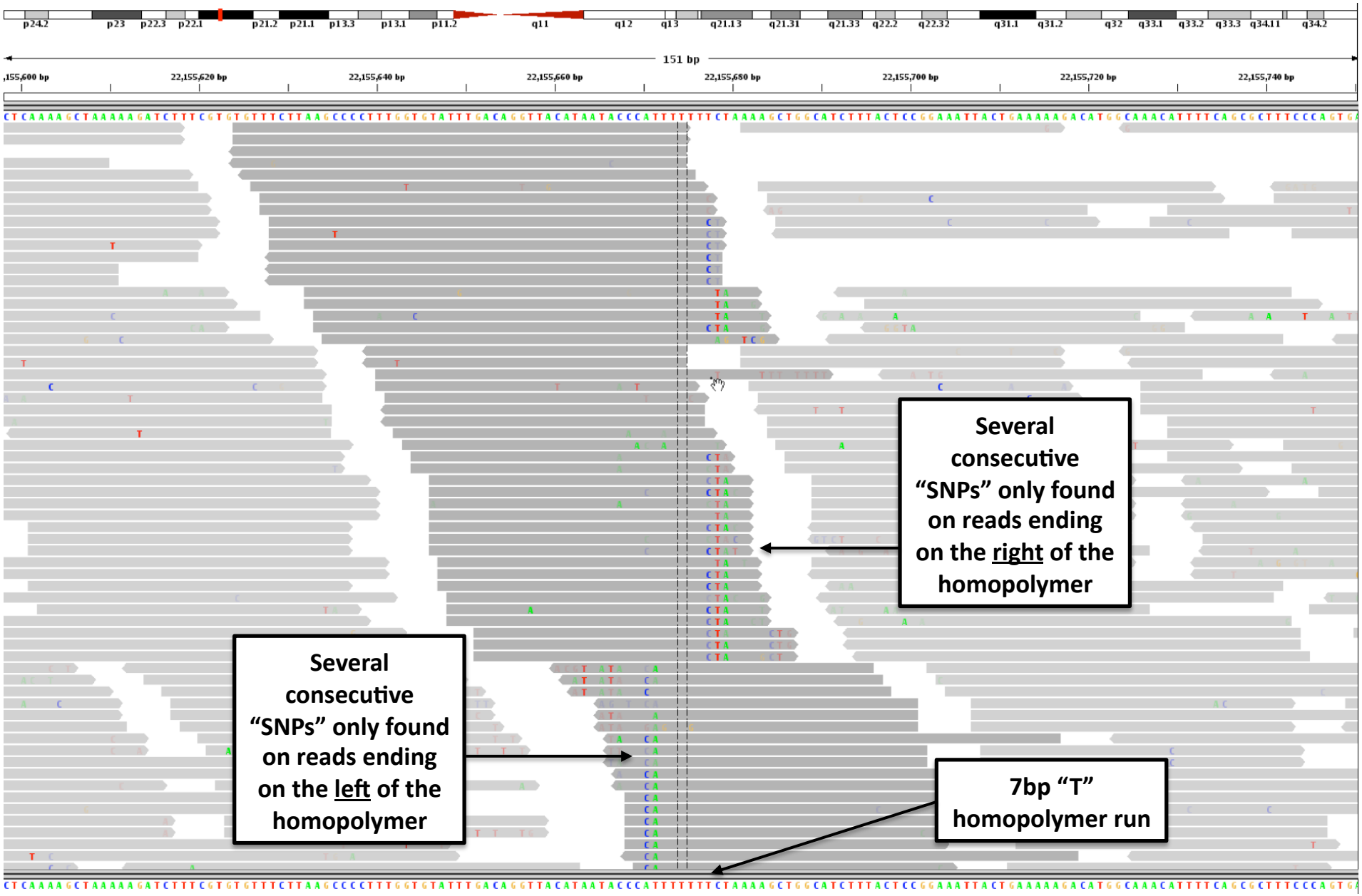


Why realign around indels?

- InDels in reads (especially near the ends) can trick the mappers into mis-aligning with mismatches
- These artifactual mismatches can harm base quality recalibration and variant detection (unless a sophisticated caller like the Haplotype Caller is used)

☑ **Realignment around indels helps improve the accuracy of several of the downstream processing steps.**

An example of a strand-discordant locus



Local realignment uncovers the hidden indel in these reads and eliminates all the potential FP SNPs

Three types of realignment targets

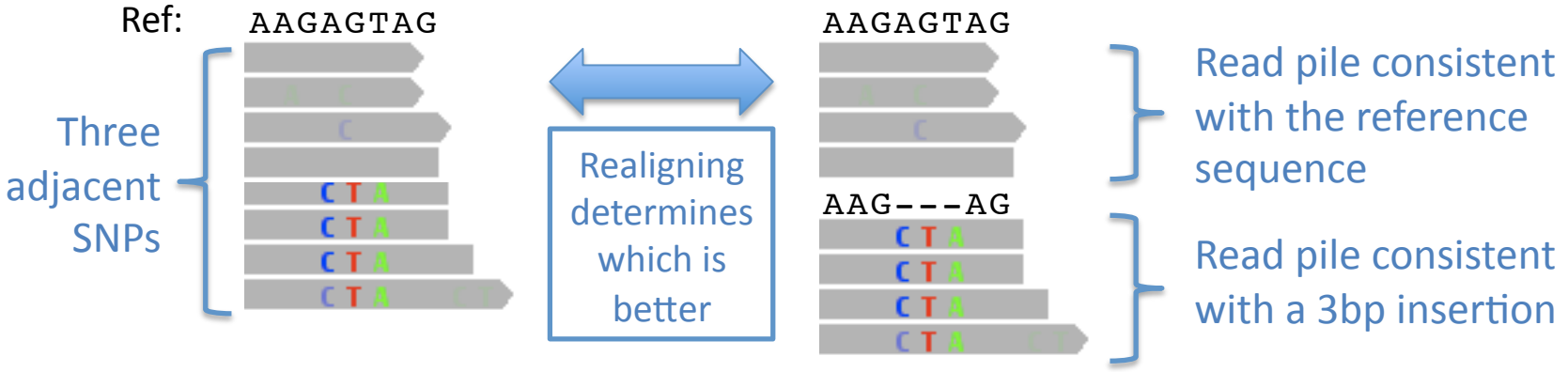
- Known sites (e.g. dbSNP, 1000 Genomes)
- Indels seen in original alignments (in CIGARs)
- Sites where evidence suggests a hidden indel

    - This is done from an entropy calculation

    - Computes activity score and if it is above a threshold, the region is included as realignment interval

Local realignment identifies most parsimonious alignment along all reads at a problematic locus

1. Find the best alternate consensus sequence that, together with the reference, best fits the reads in a pile (maximum of 1 indel)

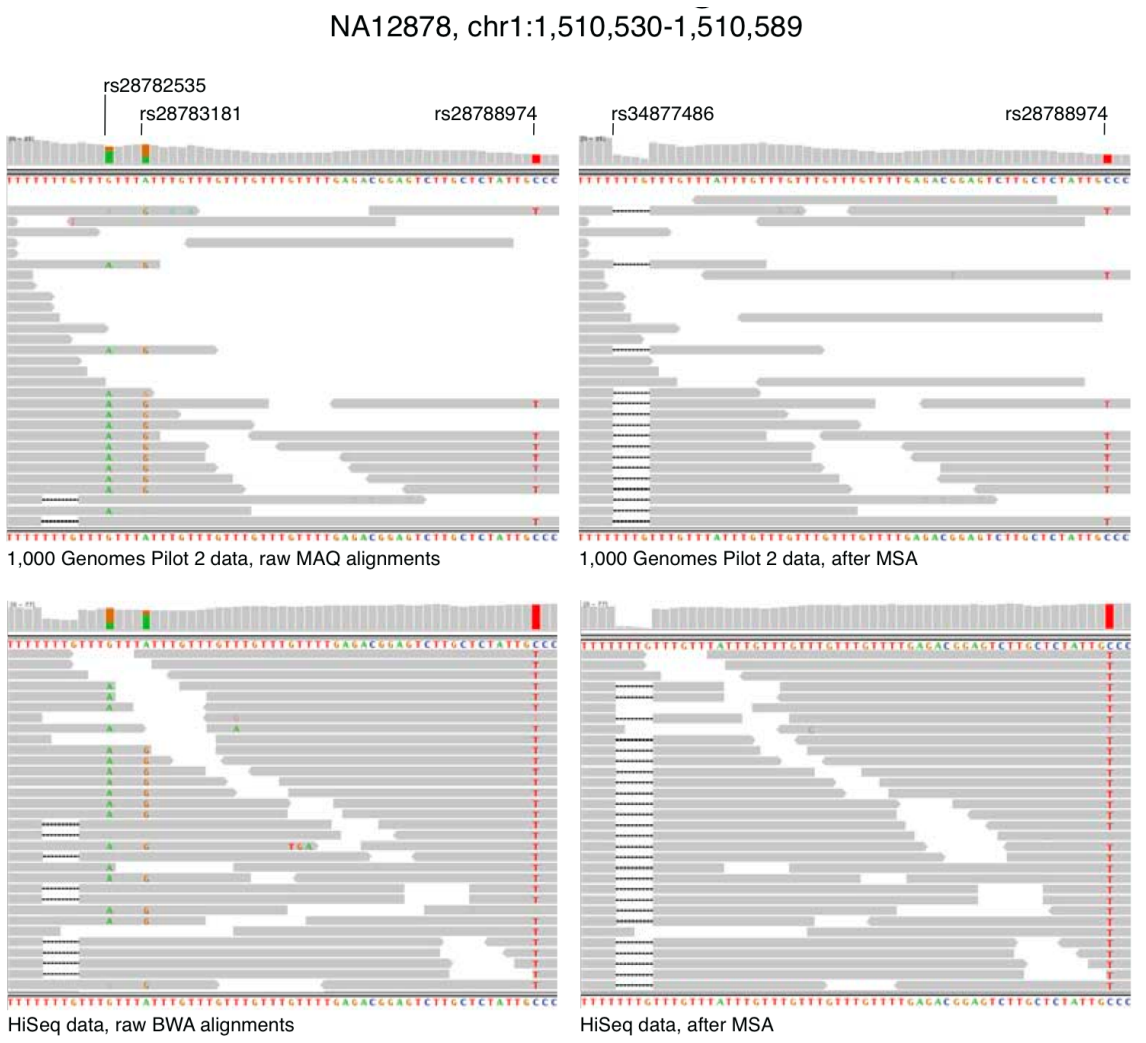


2. The score for an alternate consensus is the total sum of the quality scores of mismatching bases

3. If the score of the best alternate consensus is sufficiently better than the original alignments (using a LOD score), then we accept the proposed realignment of the reads

This is what a realigned BAM looks like

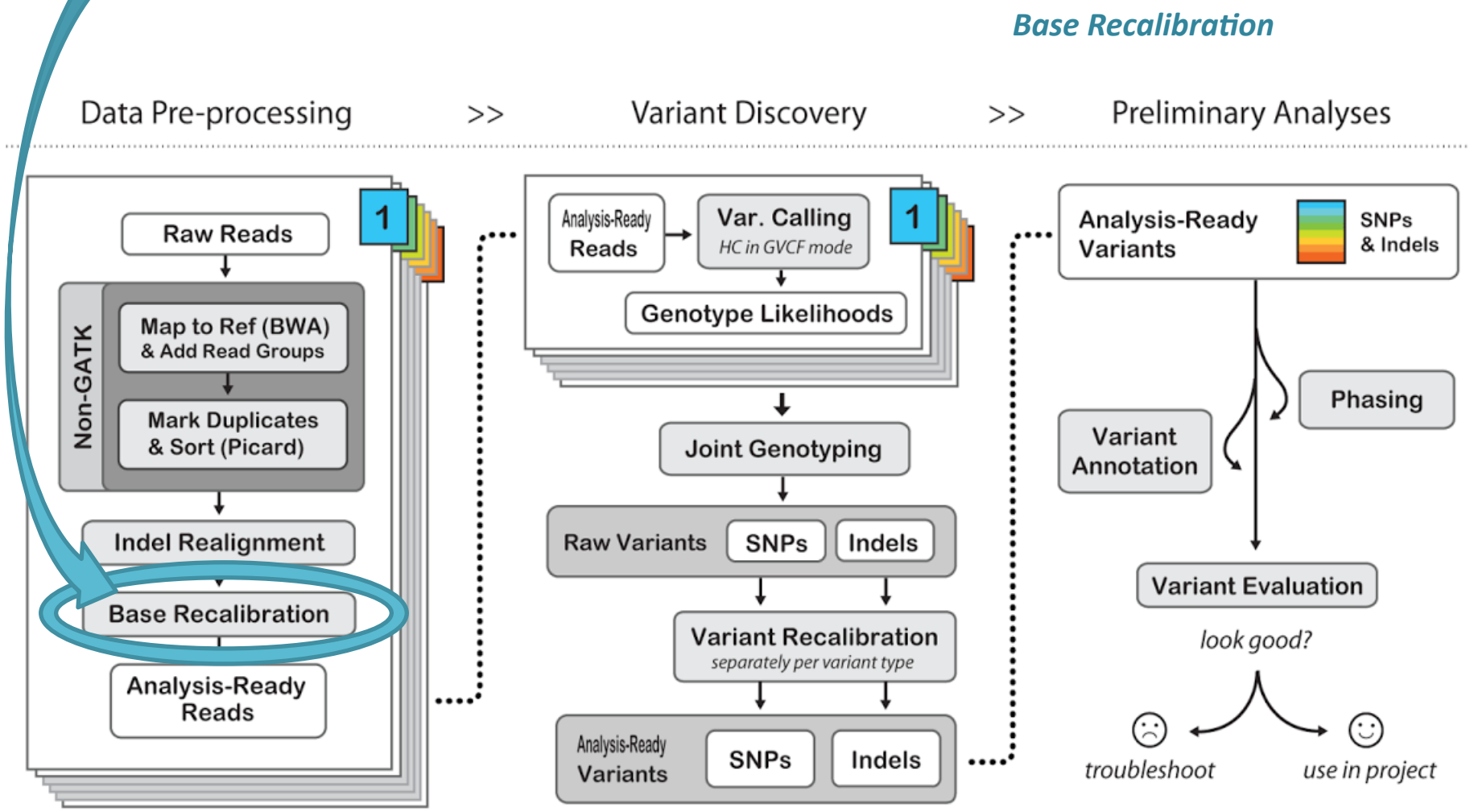
Before



After

DePristo, M., Banks, E., Poplin, R. et. al, A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Gen.

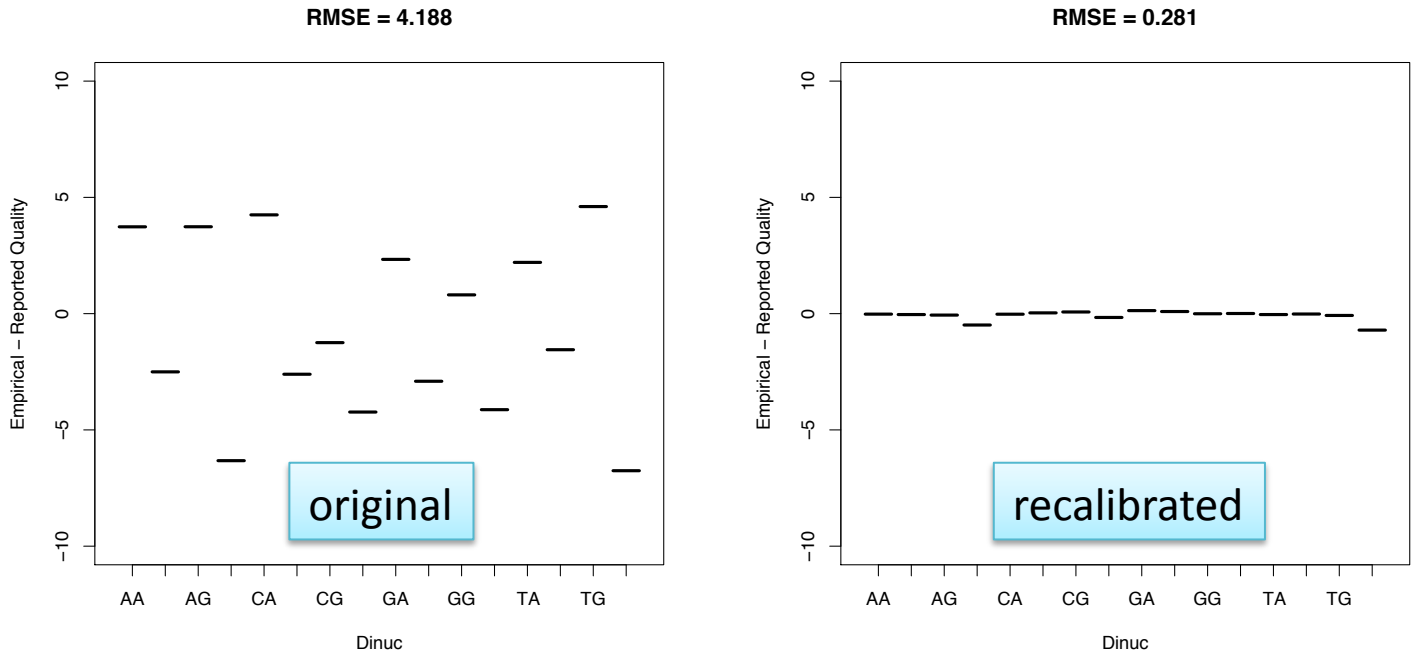
We are here in the Best Practices workflow



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

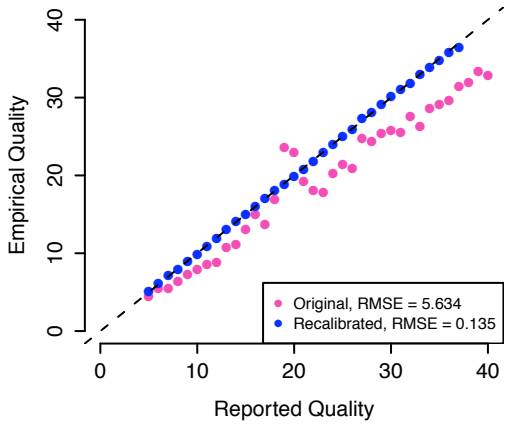
(lane, original quality score, machine cycle, and sequencing context)

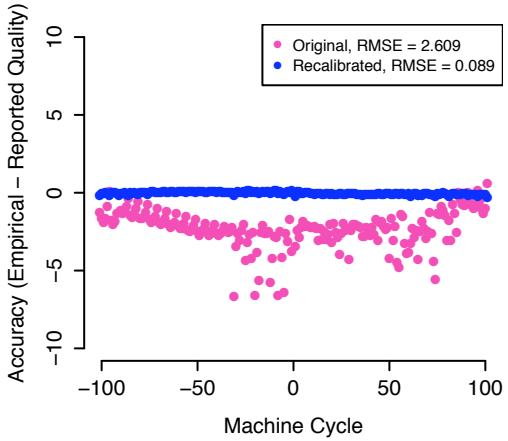
$$\frac{\#\text{ of reference mismatches} + 1}{\#\text{ of observed bases} + 2}$$

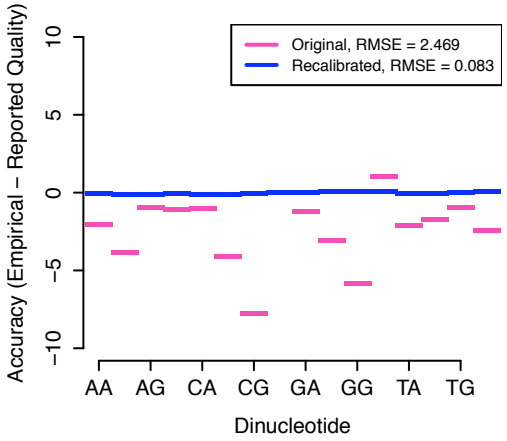
PHRED-scaled quality score

Did the recalibration work properly?

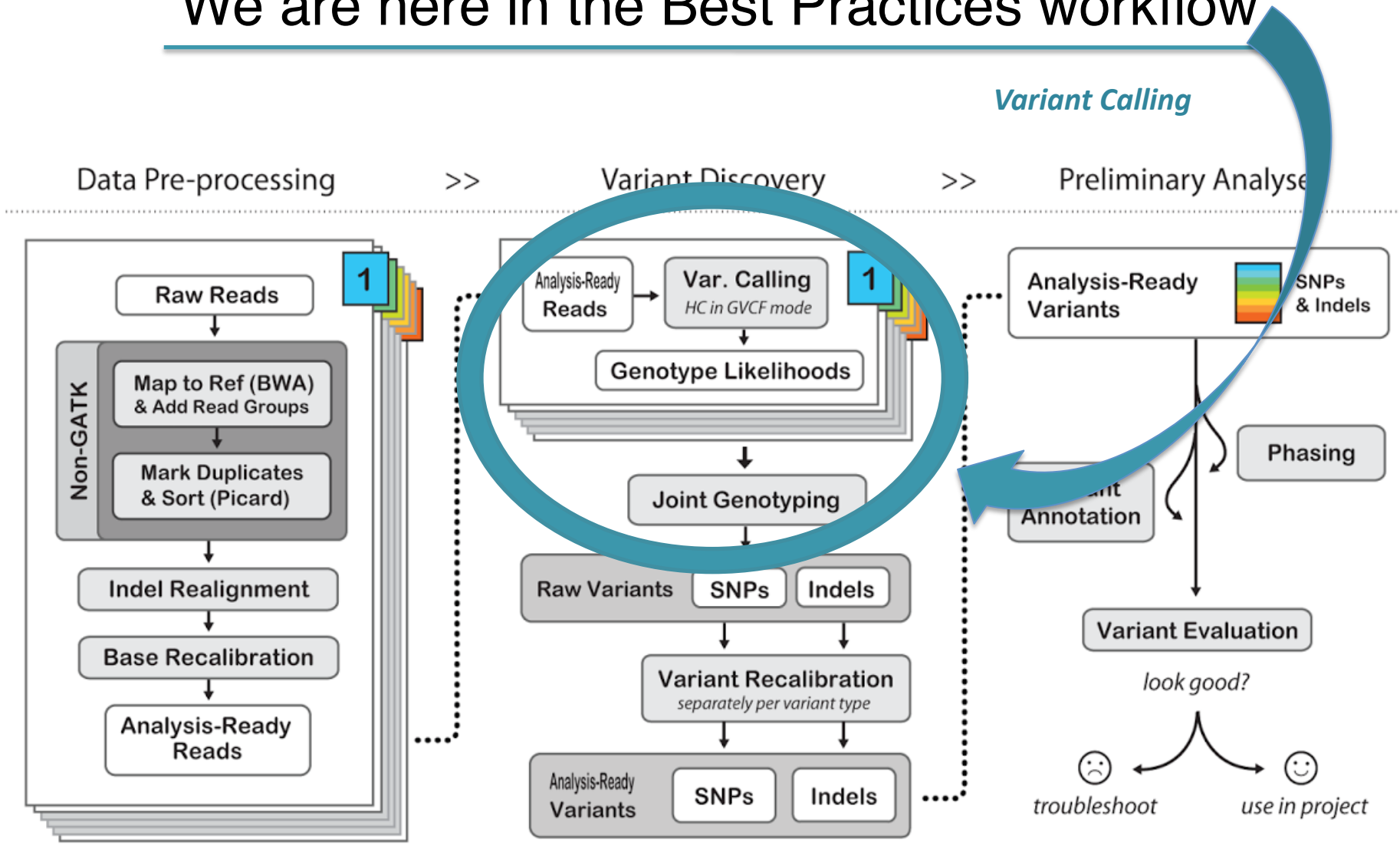
Post-recalibration quality scores should fit the empirically-derived quality scores very well; no obvious systematic biases should remain



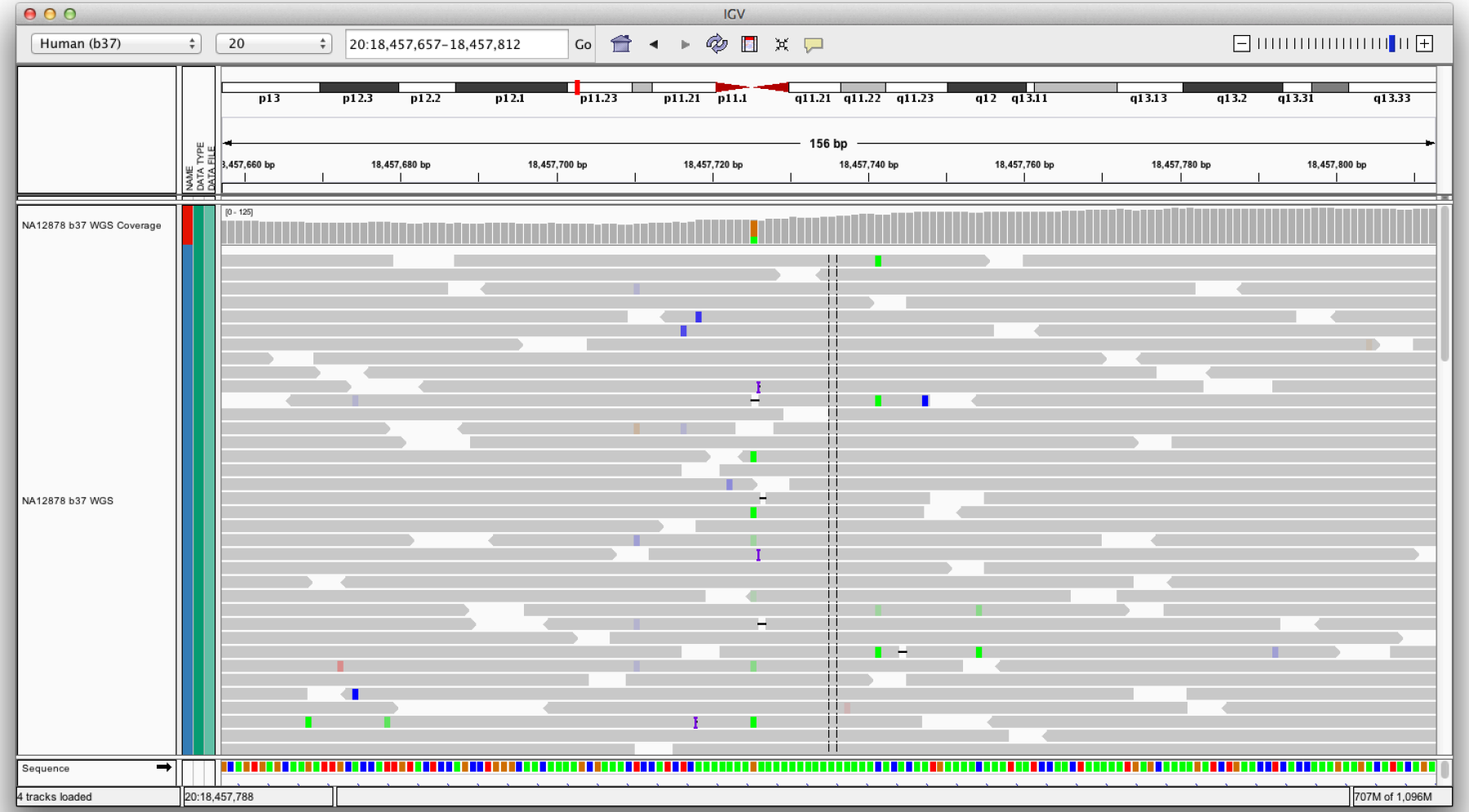




We are here in the Best Practices workflow



Real mutations are hidden in the noise



Summed up in GATK terms

Bayesian model

Prior of the genotype

Likelihood of the genotype

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

Diploid assumption

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

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

Variant callers in GATK

- UnifiedGenotyper

  Call SNPs and indels separately by considering each variant locus independently

  - Accepts any ploidy
  - Pooled calling

- HaplotypeCaller

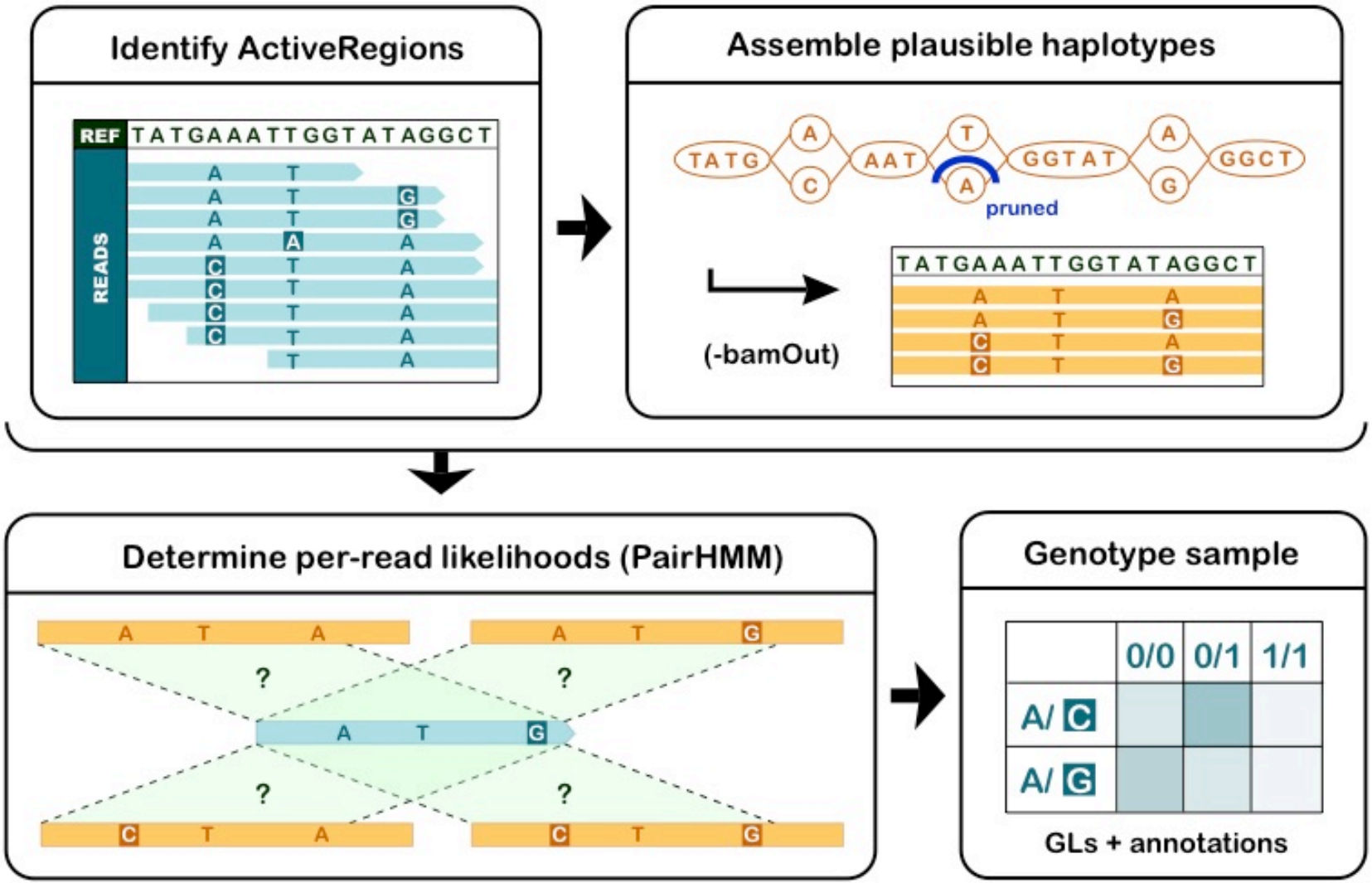
  Call SNPs, indels, and some SVs simultaneously by doing local re-assembly and considering haplotypes

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

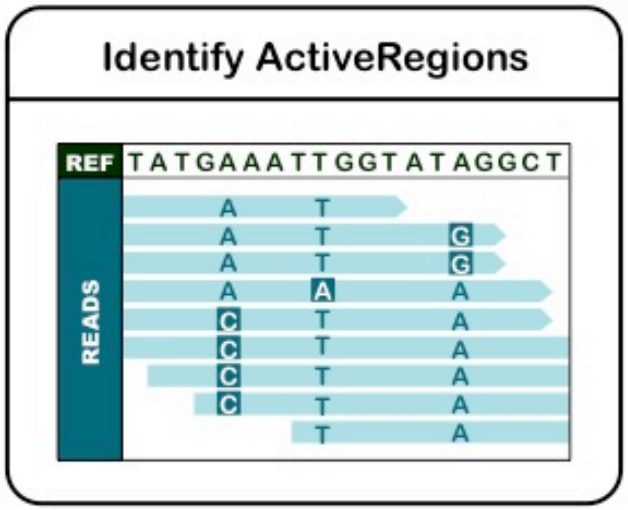
HaplotypeCaller method overview

- 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



Determining ActiveRegions



- Sliding window along the genome reference
- Count mismatches, indels and soft clips

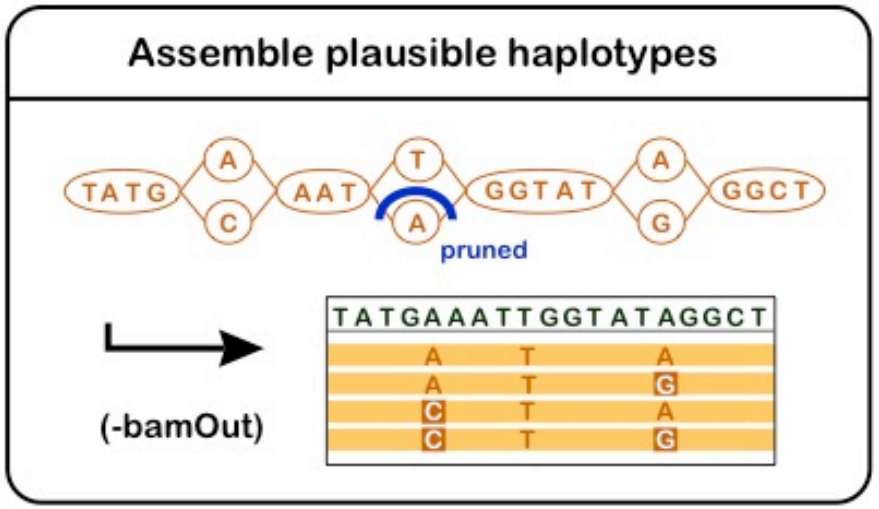
➢ **Measure of entropy**

Over threshold: trigger "ActiveRegion" where HC will proceed

Can specify a perticular interval using the –forceActive parameter

Assemble plausible haplotypes

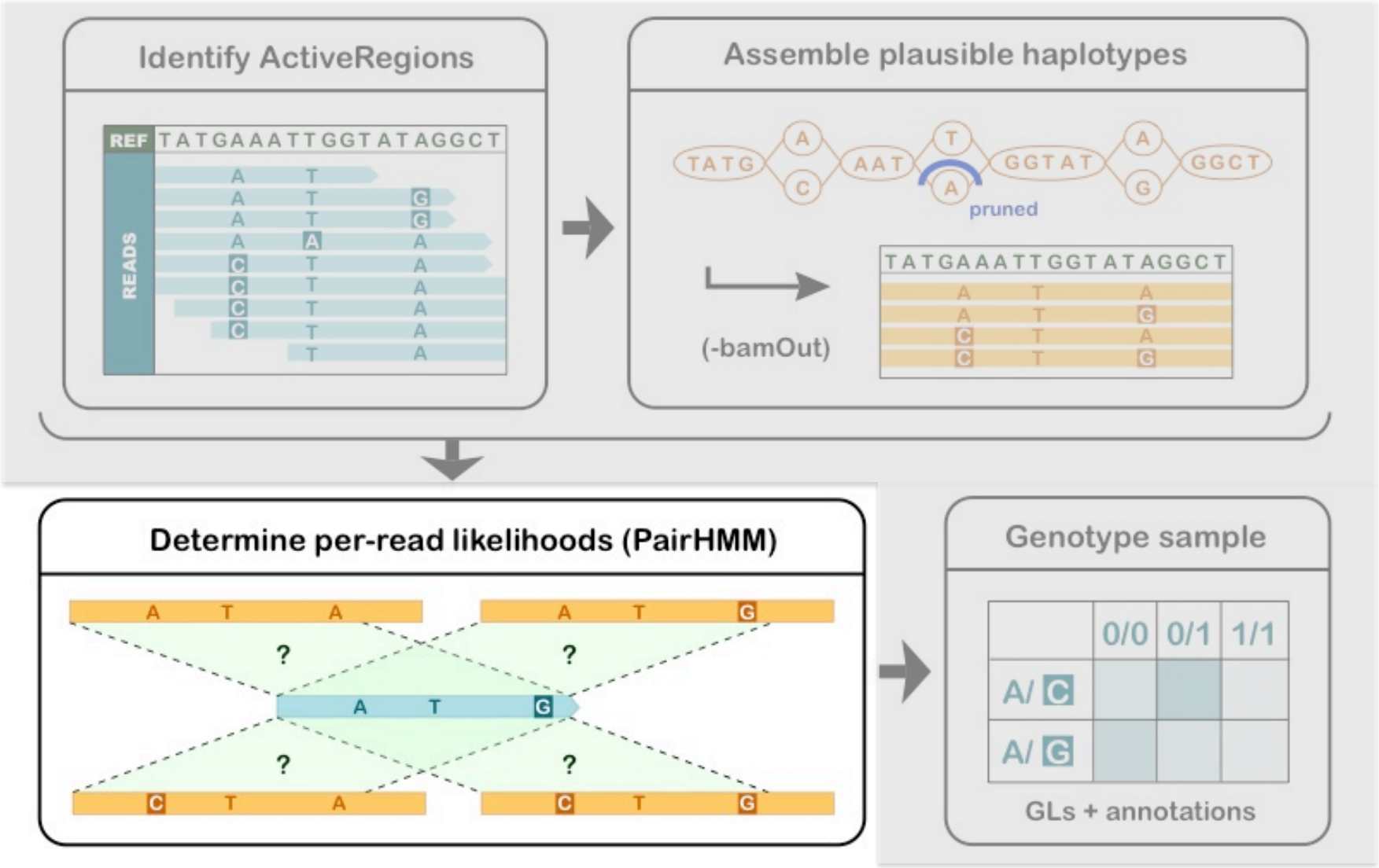
- Local re-assembly
- Traverse graph to collect most likely haplotypes
- Align haplotypes to ref using Smith-Waterman



Likely haplotypes + candidate variant sites

Can have HC output the reassembled reads and selected halpotypes using the –bamOut parameter

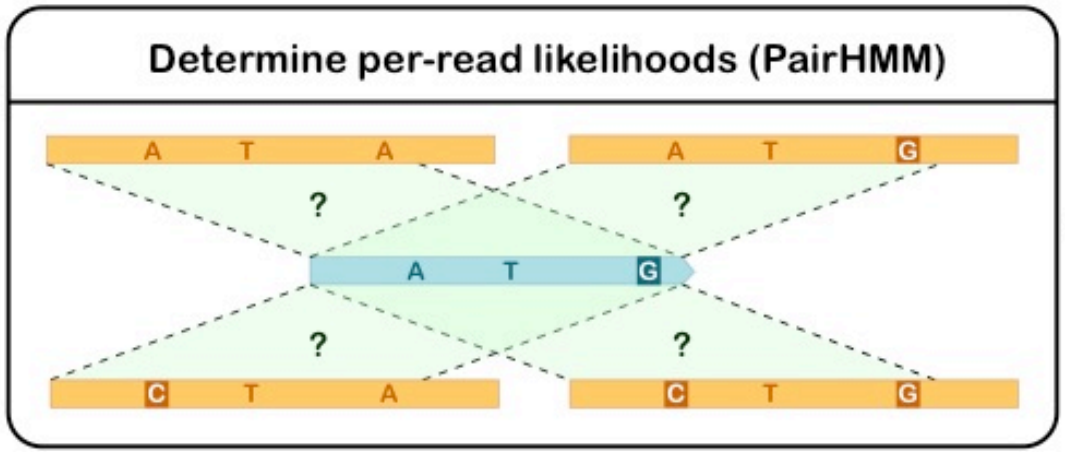
HC method illustrated

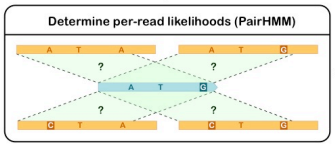


Score haplotypes using PairHMM

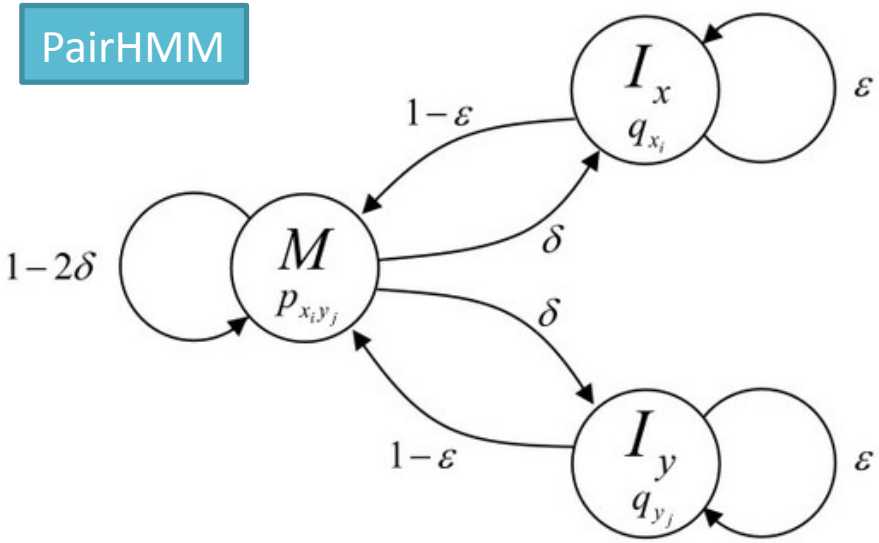
- Calculate data likelihoods given the haplotype
  - PairHMM aligns each read to each haplotype

Likelihood of the read being observed if a given haplotype is considered true





PairHMM



Empirical gap penalties
= derived from data by BQSR

Base mismatch penalties
= base quality scores

-> likelihoods of the reads vs the haplotypes
-> store in matrix

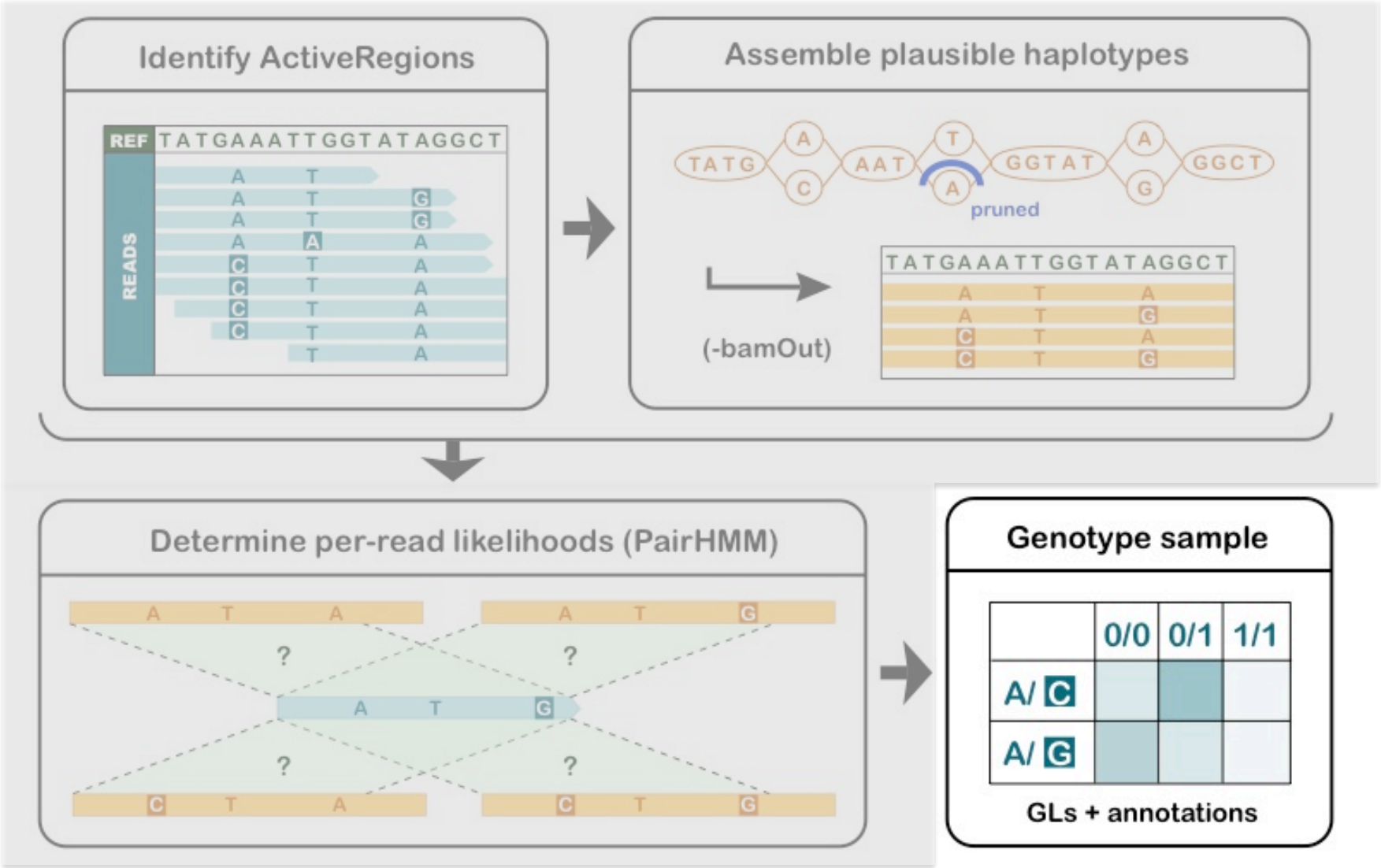
Haplotypes

Reads

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & & & A_{2n} \\ \vdots & & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

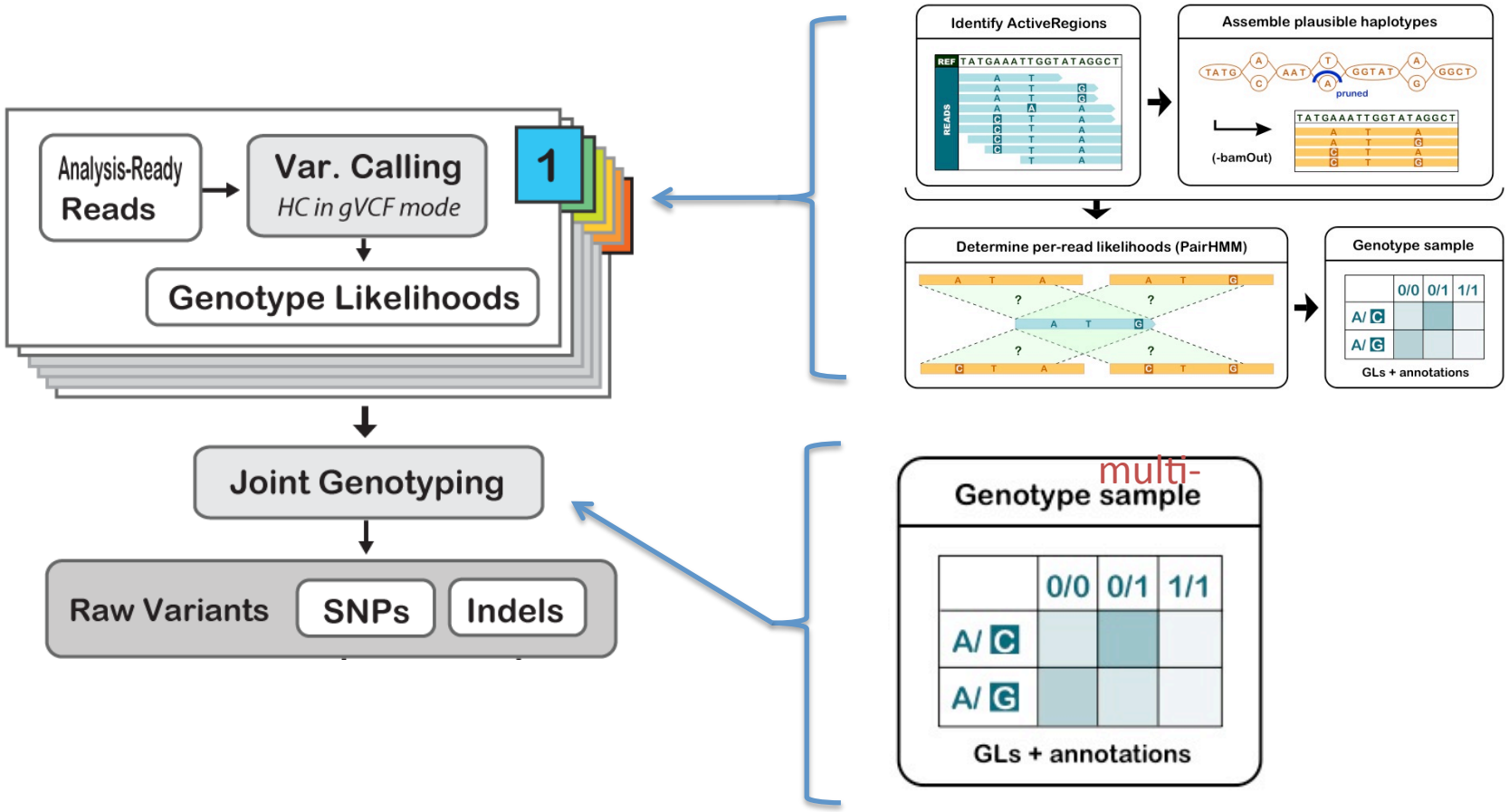
A_{ij} = probability of read vs haplotype

HC method illustrated

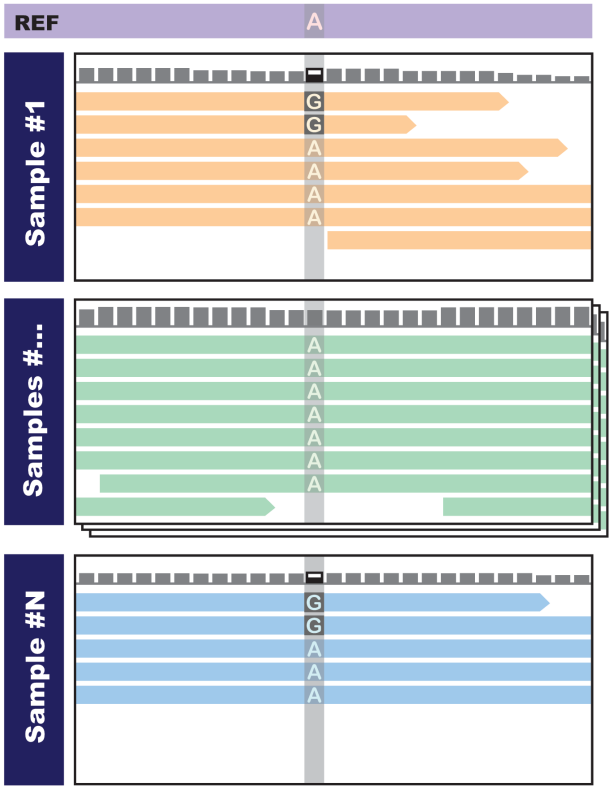


Joint Genotyping

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



Joint discovery empowers discovery at difficult sites

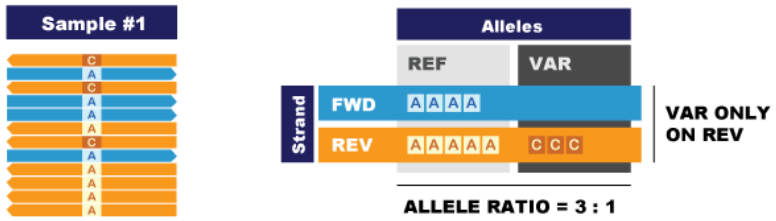


- If we analyze Sample #1 or Sample #N alone we are not confident that the variant is real

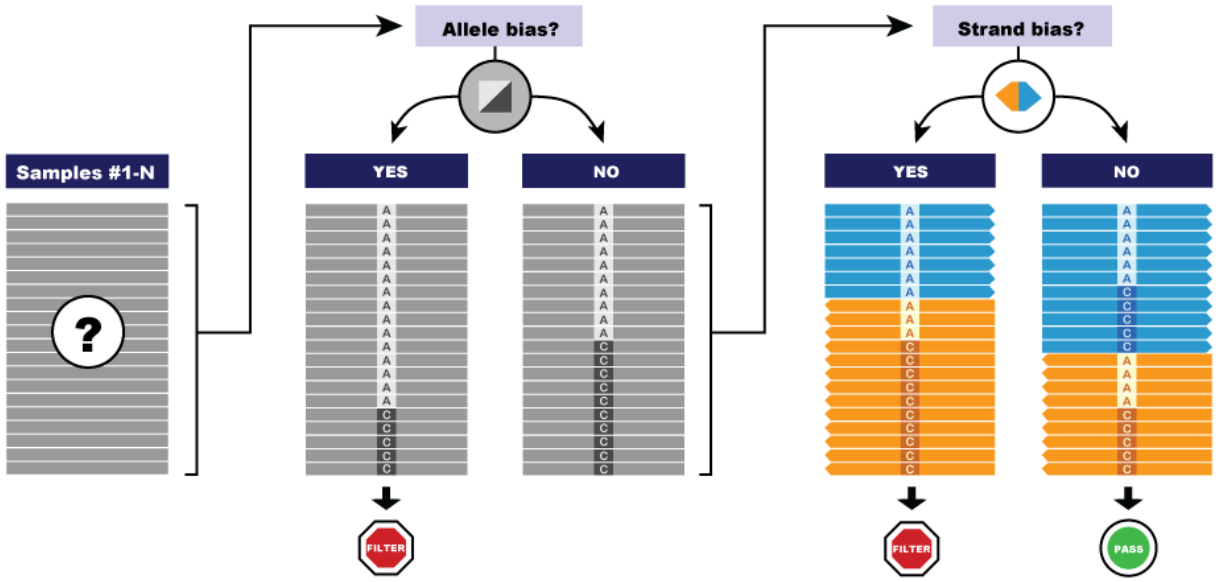
- If we see both samples then we are more confident that there is real variation at this site in the cohort

Joint discovery helps resolve bias issues

A. Single sample showing strand and allelic biases

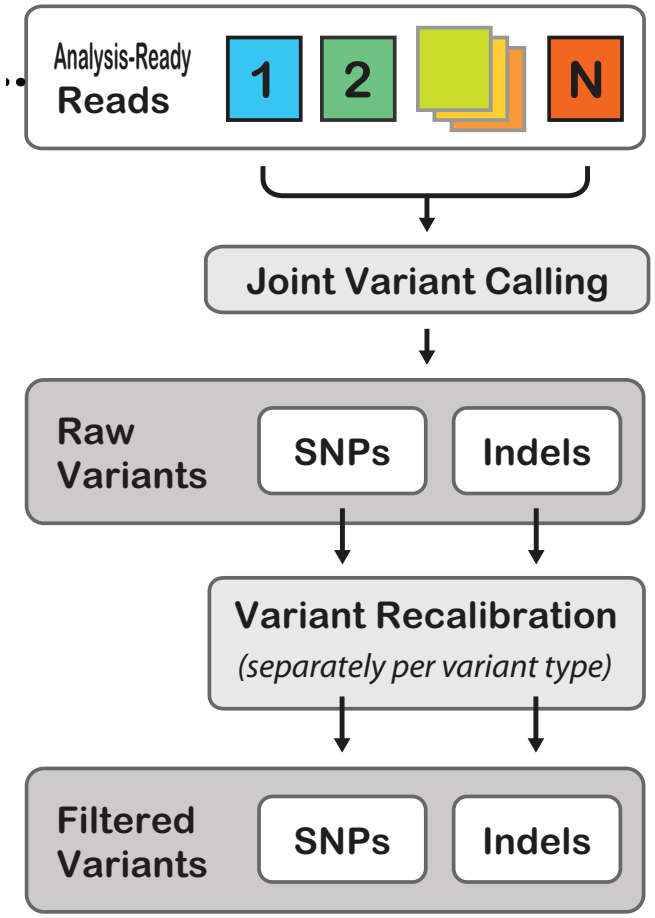


B. Decision process using evidence from multiple samples to filter out sites showing systematic biases



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!)

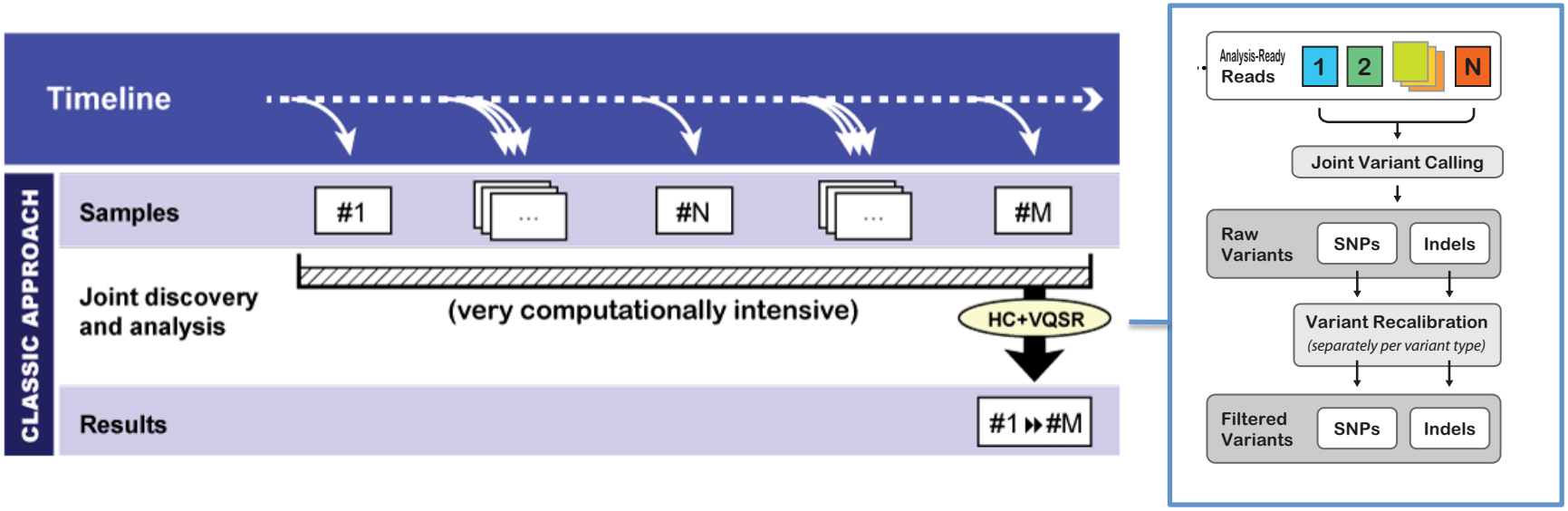


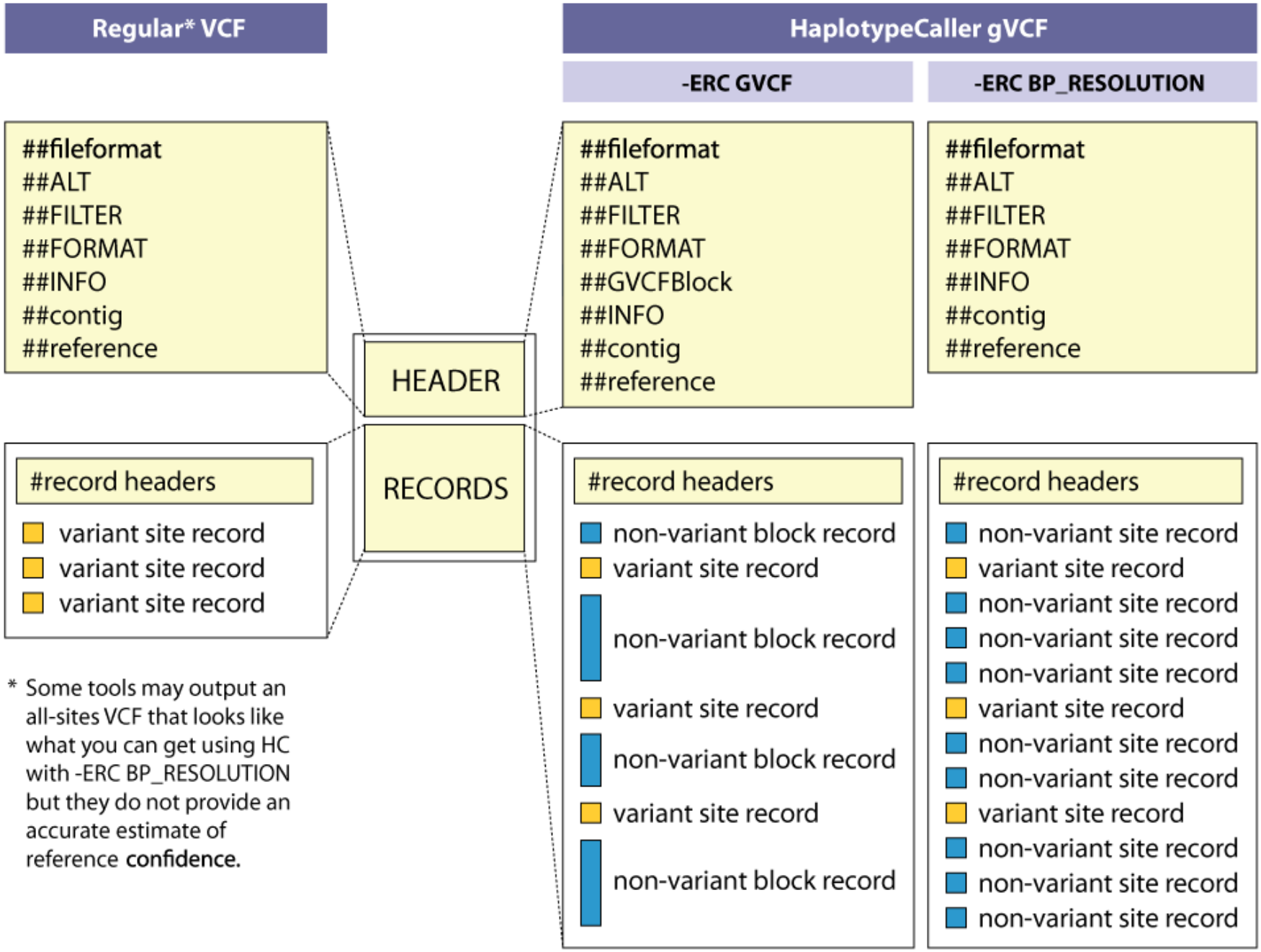
But we want to use HaplotypeCaller because it is so much better!

Problems with the “all together” approach

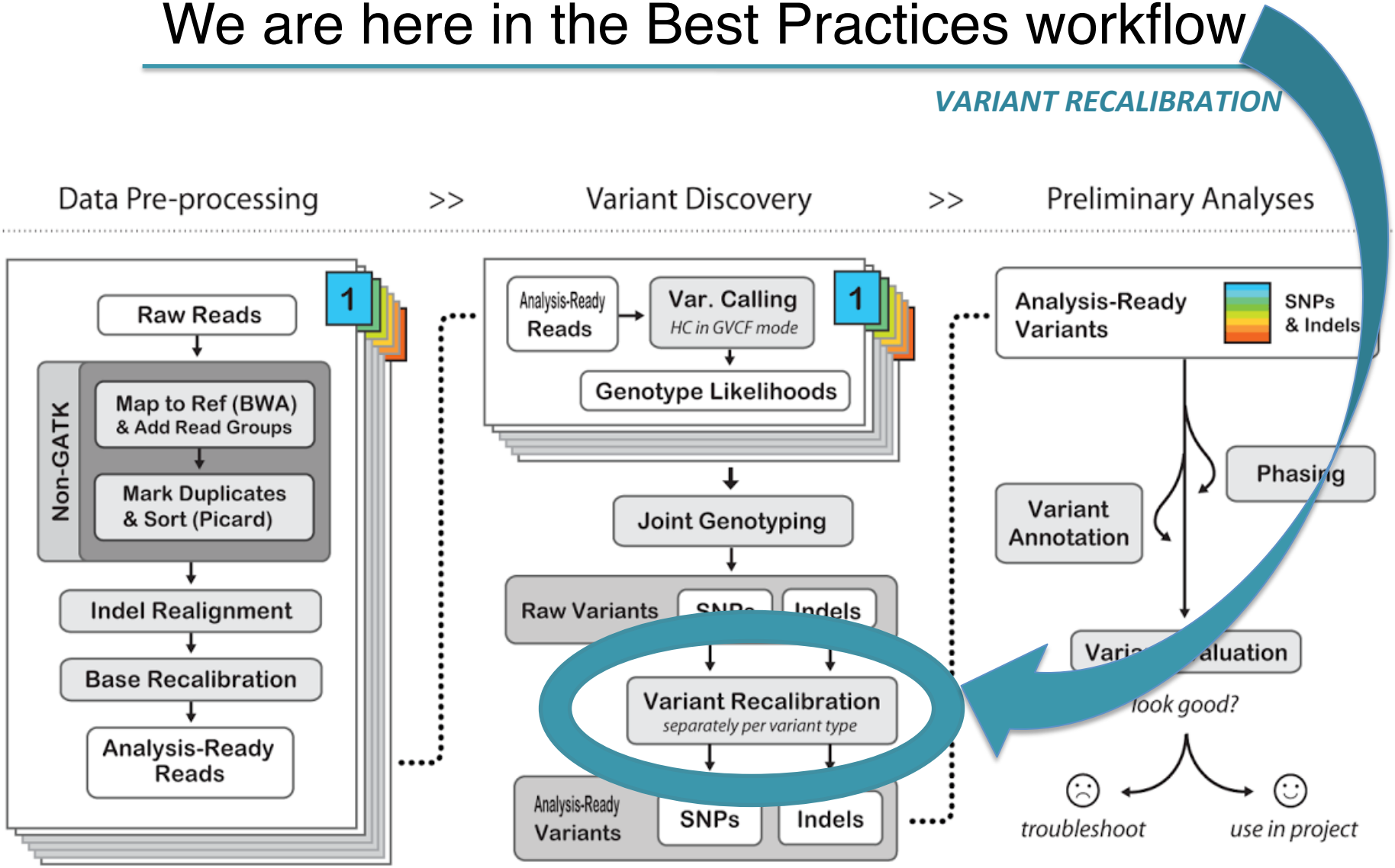
– Computing costs

– The “N+1 problem”




Regular* VCF
HaplotypeCaller gVCF
-ERC GVCF
-ERC BP_RESOLUTION
##fileformat
##ALT
##FILTER
##FORMAT
##INFO
##contig
##reference
HEADER
RECORDS
#record headers
variant site record
variant site record
variant site record
#record headers
non-variant block record
variant site record
non-variant block record
variant site record
non-variant block record
variant site record
non-variant block record
#record headers
non-variant site record
variant site record
non-variant site record
non-variant site record
non-variant site record
variant site record
non-variant site record
non-variant site record
variant site record
non-variant site record
non-variant site record
non-variant site record
* 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.

We are here in the Best Practices workflow



From annotations to mixture models

- Each variant has a diverse set of statistics associated with them called variant annotations
- Real variants tend to cluster together via these statistics
- The clusters tend to be Gaussianly distributed
- So a Gaussian mixture model can be fit to the data and new potential variants can be evaluated against this model

Variant annotations provide key information to identify and remove artifacts!

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

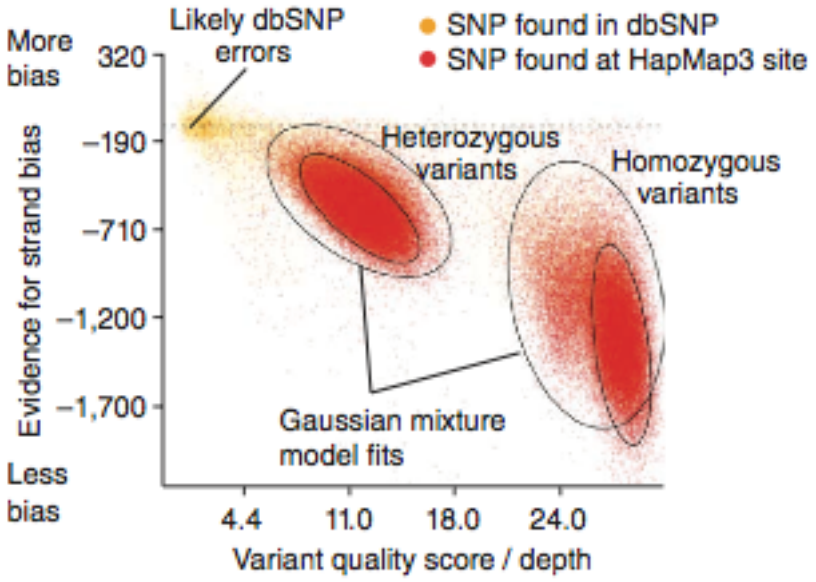
```
22 49582364    .    A    G    198.96    .
   AC=3;
   AF=0.50;
   AN=6;
   DP=87;
   MLEAC=3;
   MLEAF=0.50;
   MQ=71.31;
   MQ0=22;
   QD=2.29;
   SB=-31.76
   GT:DP:GQ    0/1:12:99.00    0/1:11:89.43    0/1:28:37.78
```

INFO field

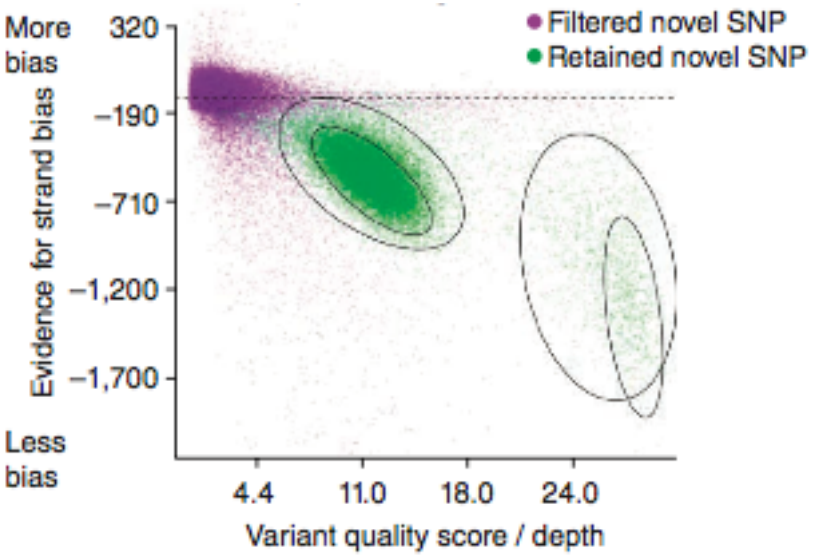
AC	No. chromosomes carrying alt allele	MLEAF	Max likelihood AF
AN	Total no. of chromosomes	MQ	RMS MAPQ of all reads
AF	Allele frequency	MQ0	No. of MAPQ 0 reads at locus
DP	Depth of coverage	QD	QUAL score over depth
MLEAC	Max likelihood AC	SB	Estimated strand bias score

Training on high-confidence known sites to determine the probability that other sites are true

Model Training Using HapMap



Evaluating Novel Variants

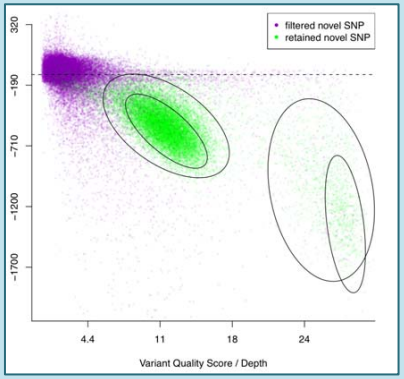


Modified from DePristo et al. Nature Genetics. 2011

Variant Recalibration steps & tools

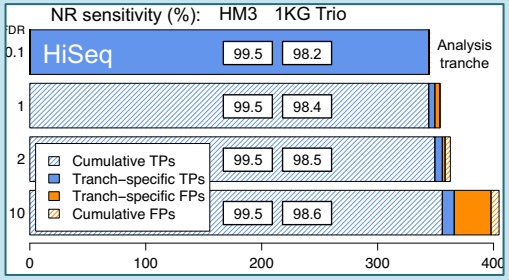
- Build the Gaussian mixture model

➔ **VariantRecalibrator**

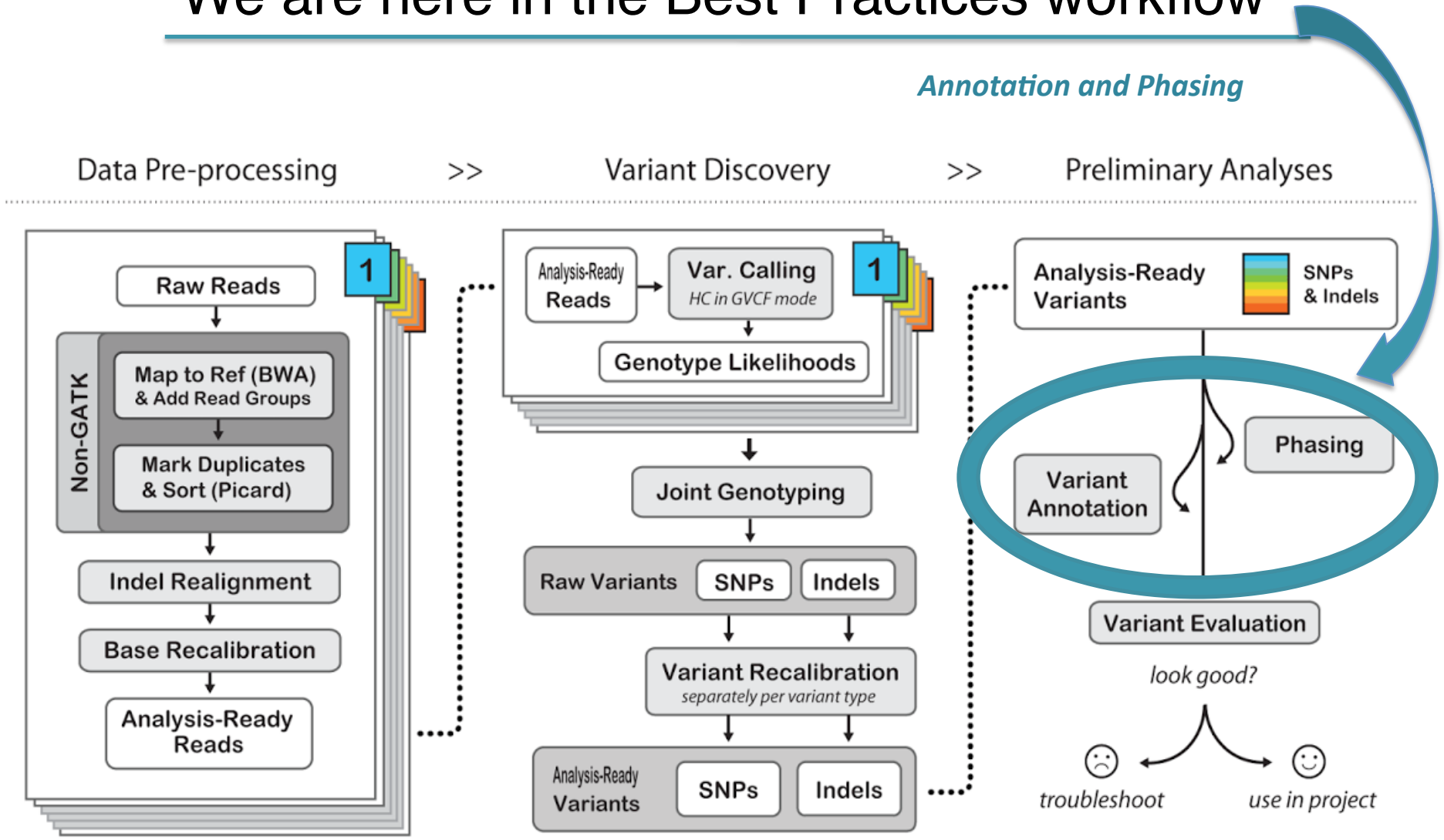


- Apply filters and write new annotated VCF

➔ **ApplyRecalibration**



We are here in the Best Practices workflow



Variant evaluation requires information about
the **nature and context** of variants

- Examples:
  - Are there **repeats** around the variant?
  - What is the **sequence quality** like in the variant context?
  - How **frequent** is the variant in related individuals?

☑ **Variant annotation in general helps refine our estimate of how likely a variant is to be true**

Variant Annotation steps & tools

- VariantAnnotator

  Built-in context annotations, extensible to any* question

- Non-GATK (snpEff, Oncotator)

  Add functional annotations to a set of variants

* In practice, some annotations are hard to add with GATK

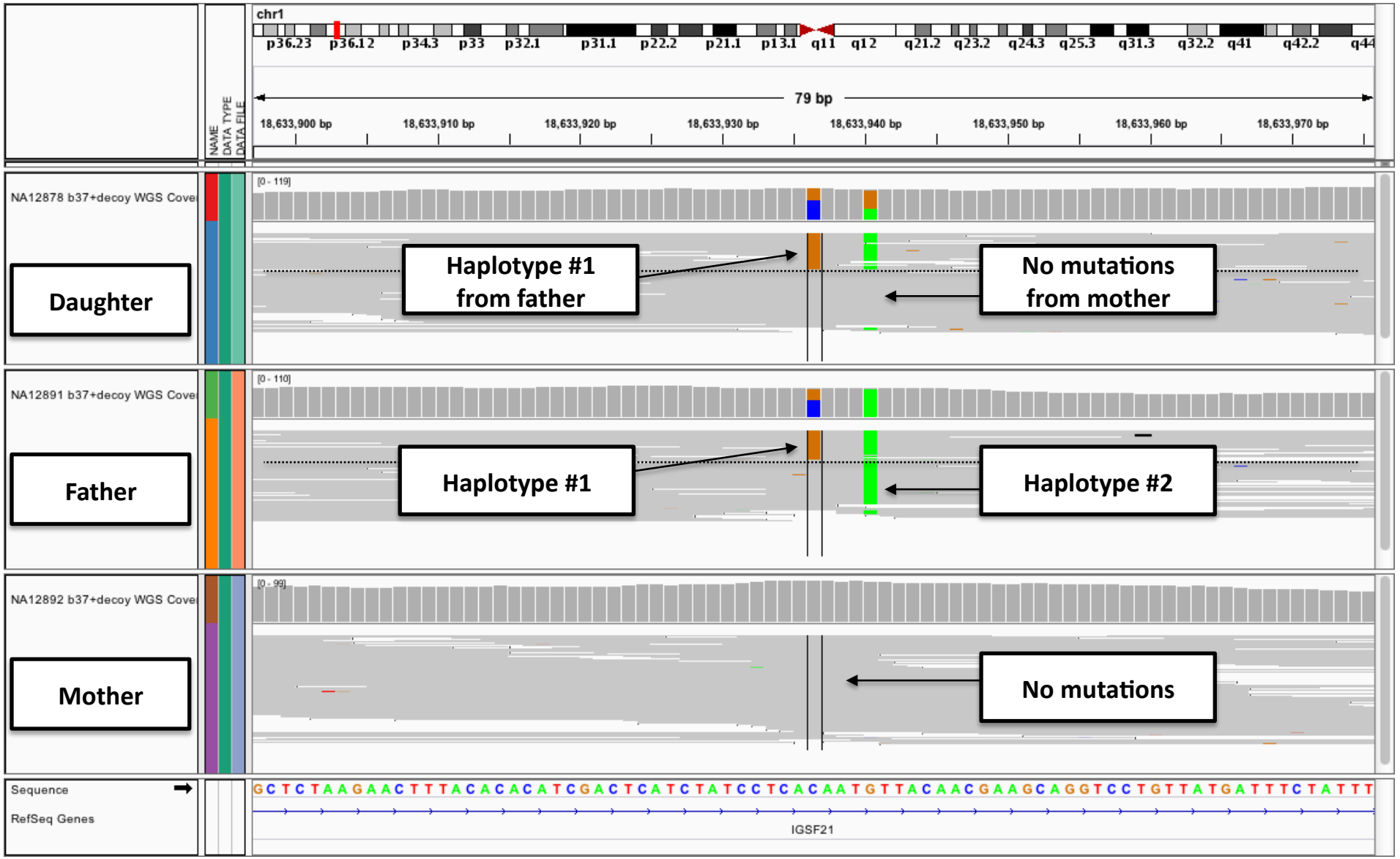
Many downstream genetic analyses need accurate genotypes and/or phasing information

- E.g. Mendelian disease caused by Loss Of Function event
  - Homozygous mutation causing disease (both copies affected)
  - Compound heterozygote (het mutations on different copies)

- Critical in population genetics studies to determine haplotype structure

☑ **Refining and phasing genotypes empowers downstream medical and population genetics analyses that require accurate determination of haplotype structure.**

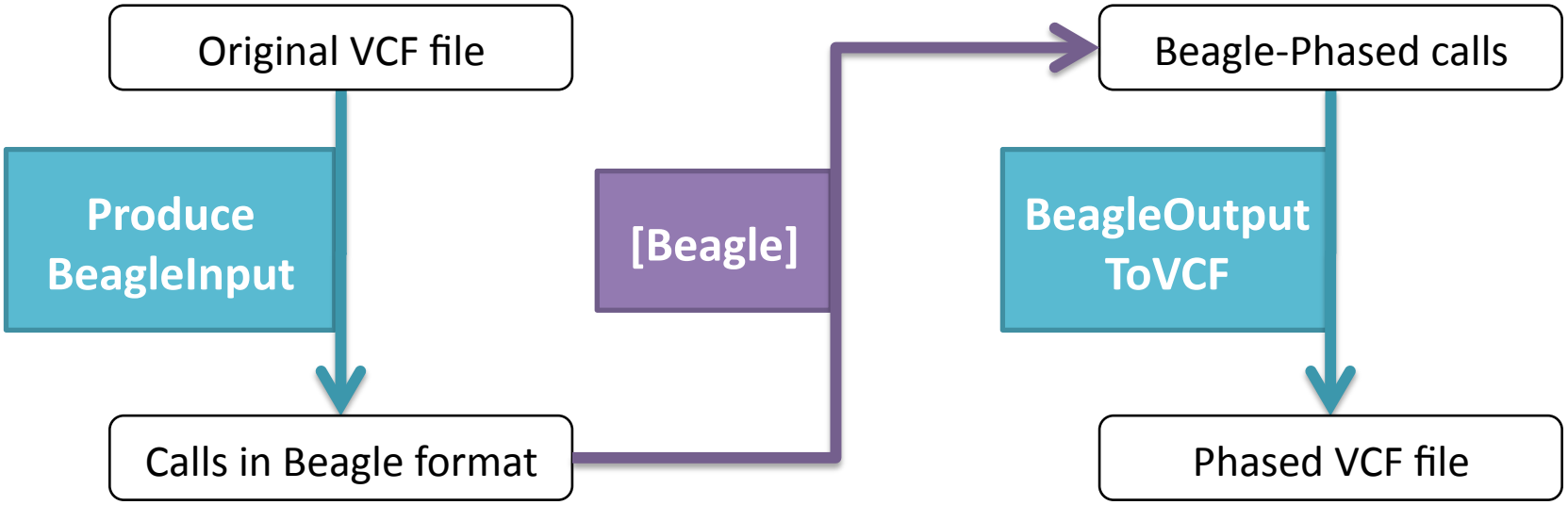
Example site showing Mendelian inheritance in a trio



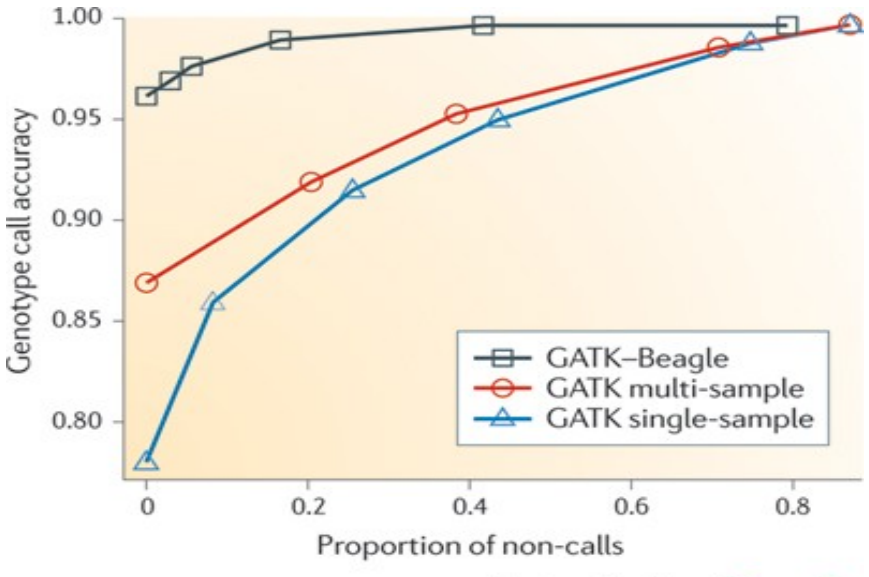
TOOL TIPS

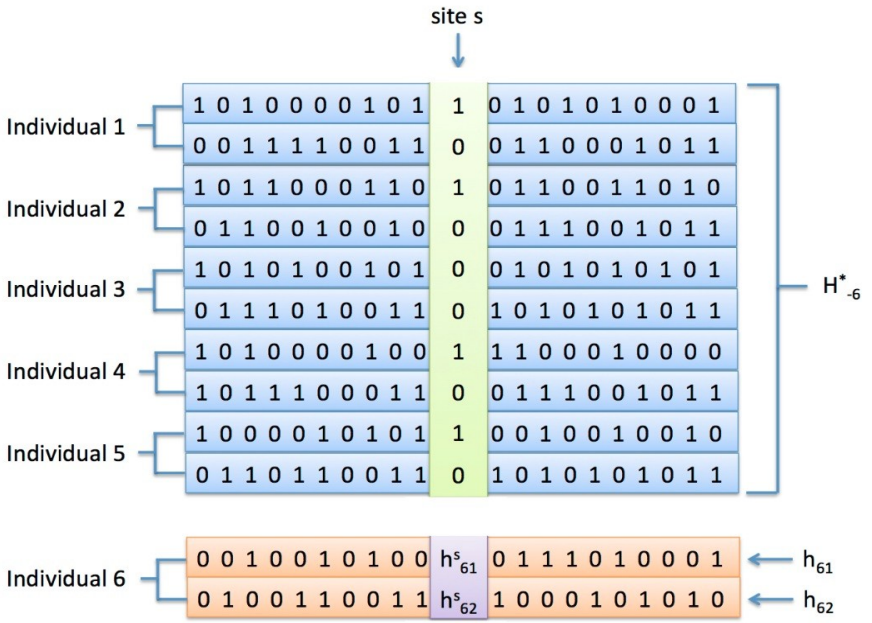
Imputation Software

- Phases variants in any number of samples by determining shared haplotypes given genotype likelihoods
- Latest version of Beagle can use VCF files as input and output but GATK provides conversion utilities for older Beagle files:



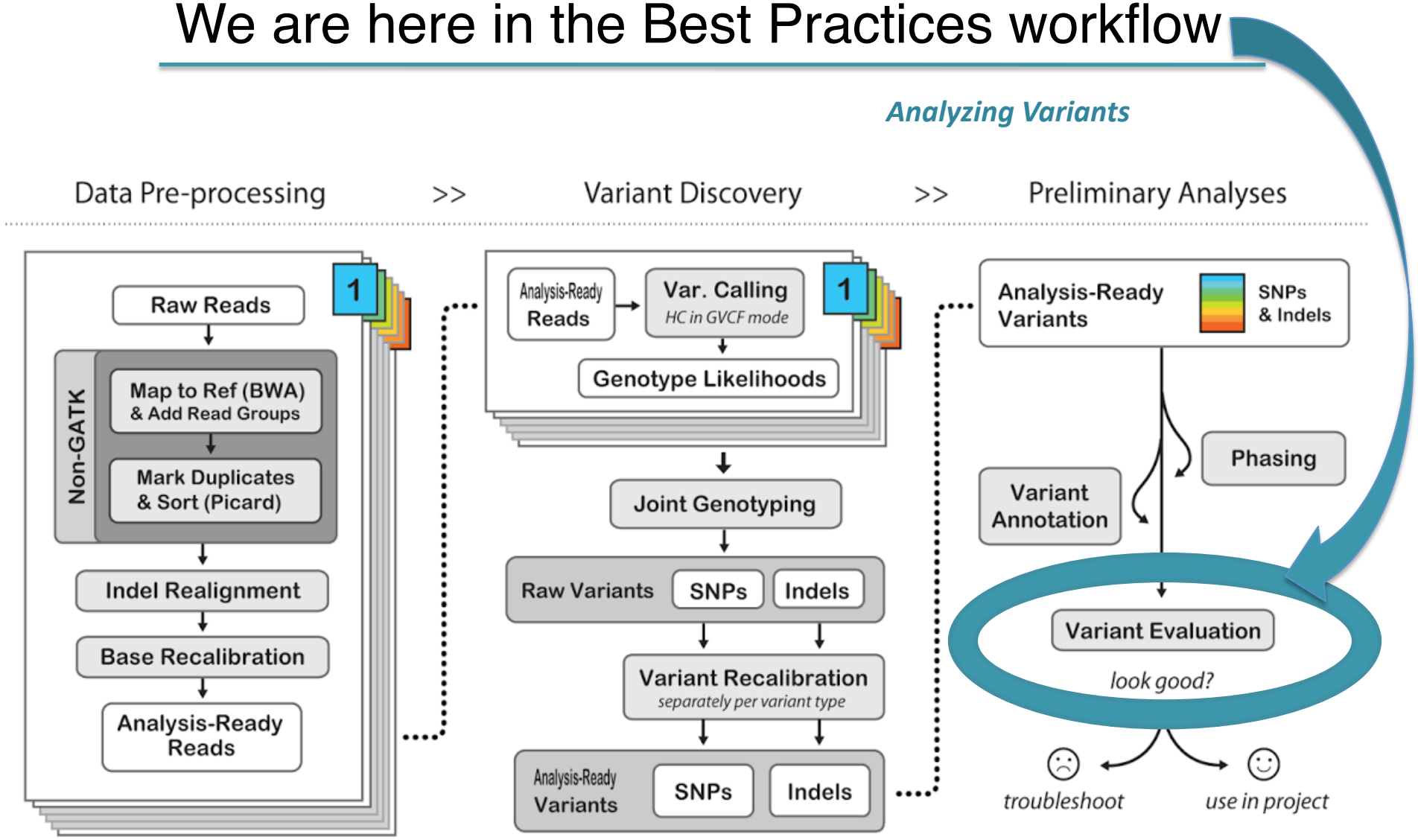
Imputation and phasing





Nielsen et al. 2011

We are here in the Best Practices 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

Official specification in
www.1000genomes.org/wiki/Analysis/Variant Call Format/vcf-variant-call-format-version-41

This is what a phased VCF looks like

Original VCF

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	MOTHER	FATHER	CHILD
1	10109	.	A	T	99	PASS	.	GT:PL	0/0:0,50,200	0/0:0,40,200	0/1:30,0,200
1	10147	.	C	A	99	PASS	.	GT:PL	**0/1**:0,30,200	**0/0**:0,50,200	**0/1**:200,40,0
1	10150	.	C	T	99	PASS	.	GT:PL	0/1:0,40,200	0/1:30,0,200	1/1:200,50,0

Phased VCF

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	MOTHER	FATHER	CHILD
1	10109	.	A	T	99	PASS	.	GT:PL:TP	0|0:0,50,200:10	0|0:0,40,200:10	0|0:30,0,200:10
1	10147	.	C	A	99	PASS	.	GT:PL:TP	**1|0**:0,30,200:10	**0|0**:0,50,200:10	**1|0**:200,40,0:10
1	10150	.	C	T	99	PASS	.	GT:PL:TP	1|0:0,40,200:10	1|0:30,0,200:10	1|1:200,50,0:10

The convention is:
Allele From Mother | Allele From Father

Need to be able to handle and evaluate variants

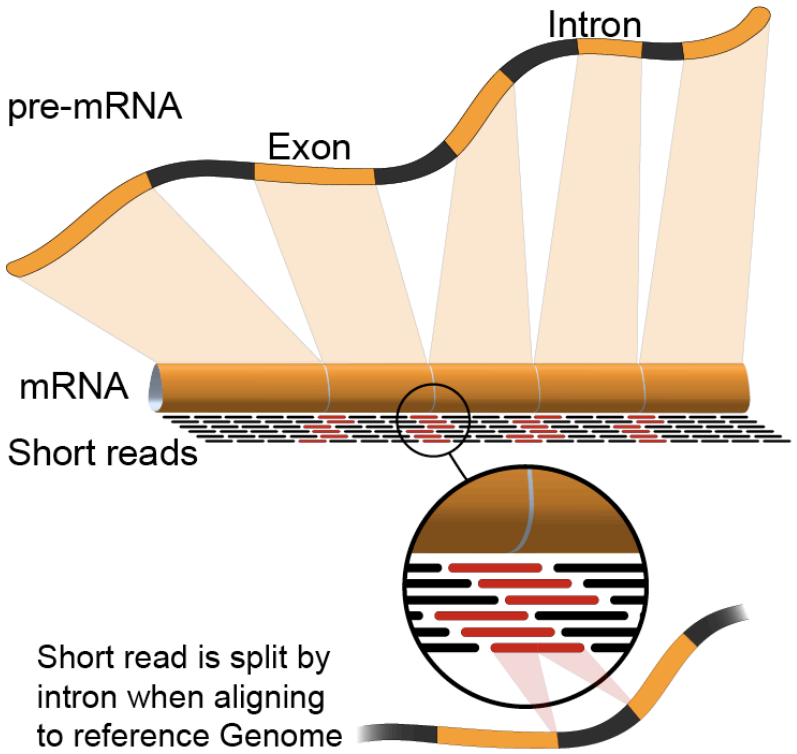
- To answer biological questions given direct observations of genetic mutations
- Quality of variant detection and genotypes directly impacts scientific (or clinical) results
- Ability to query and evaluate different cuts of a callset while integrating external data is paramount
- This module: **real examples** using GATK

Calling variants on RNAseq

Best Practices pipeline for RNAseq

A key challenge for variant calling

- Correctly handling splice junctions



Aligned to reference:

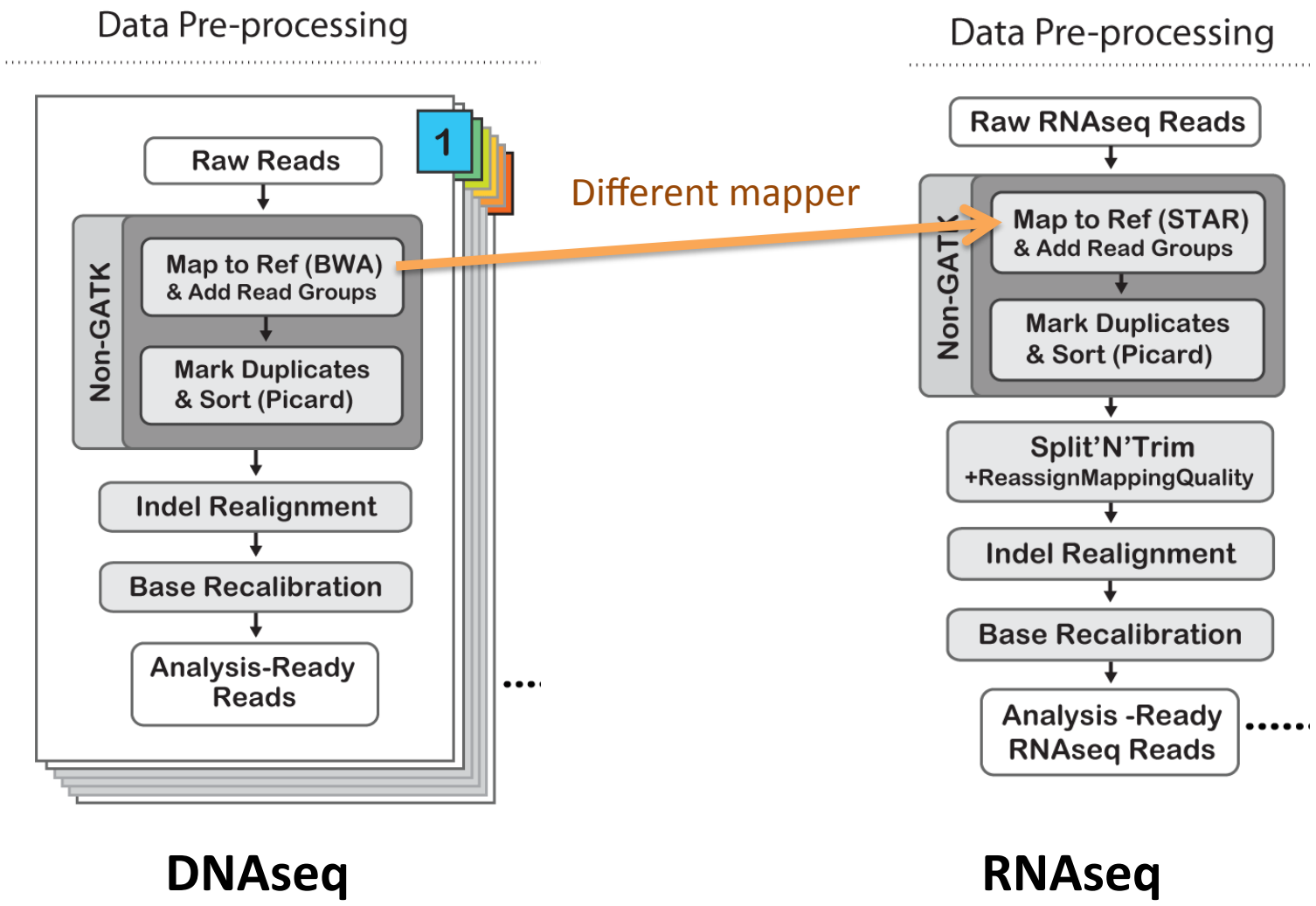
REF

GATTCNNNNNNNAATTATT

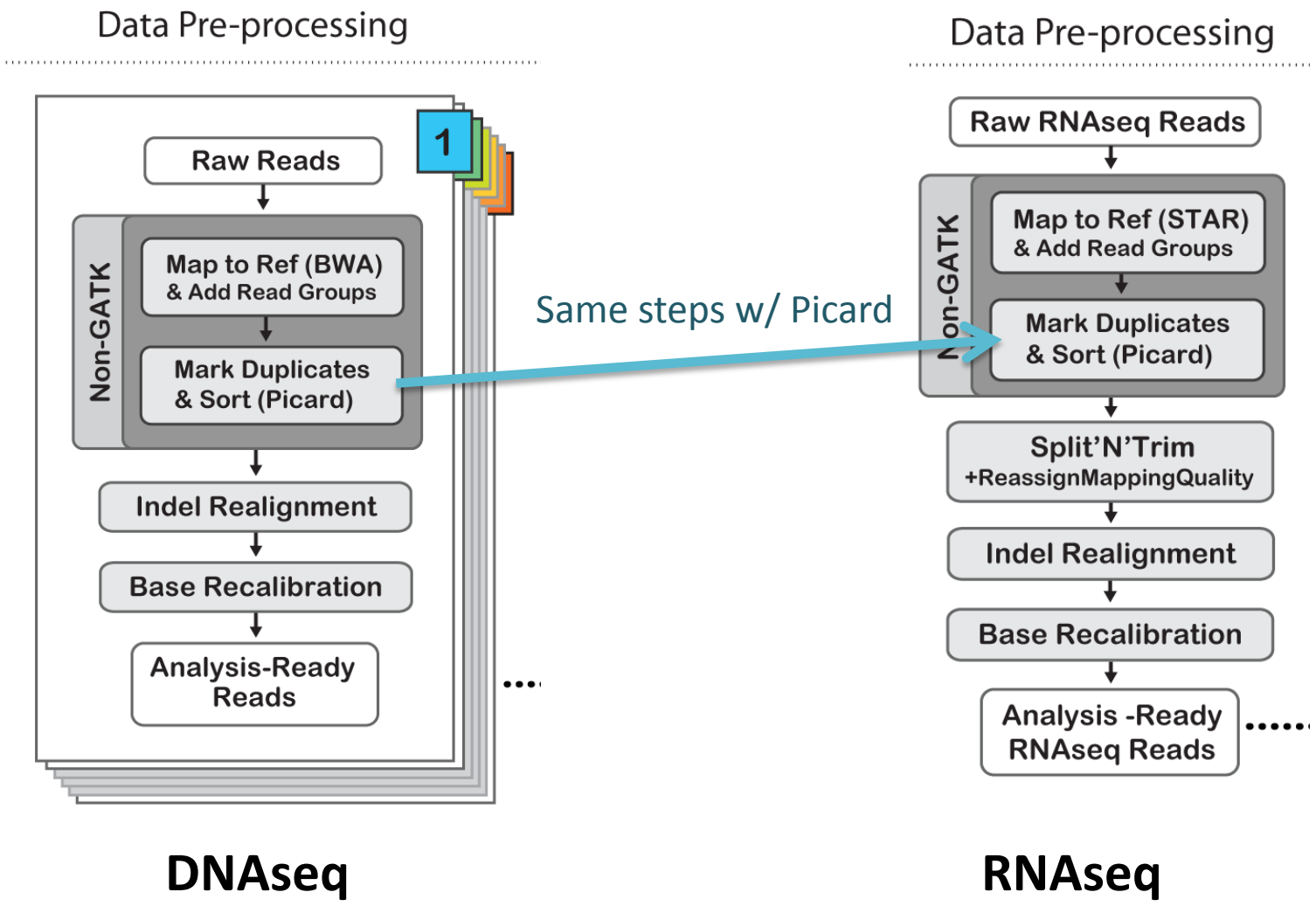
Several types of problems in these regions

Cannot simply use GATK Best Practices that were developed for DNAseq!

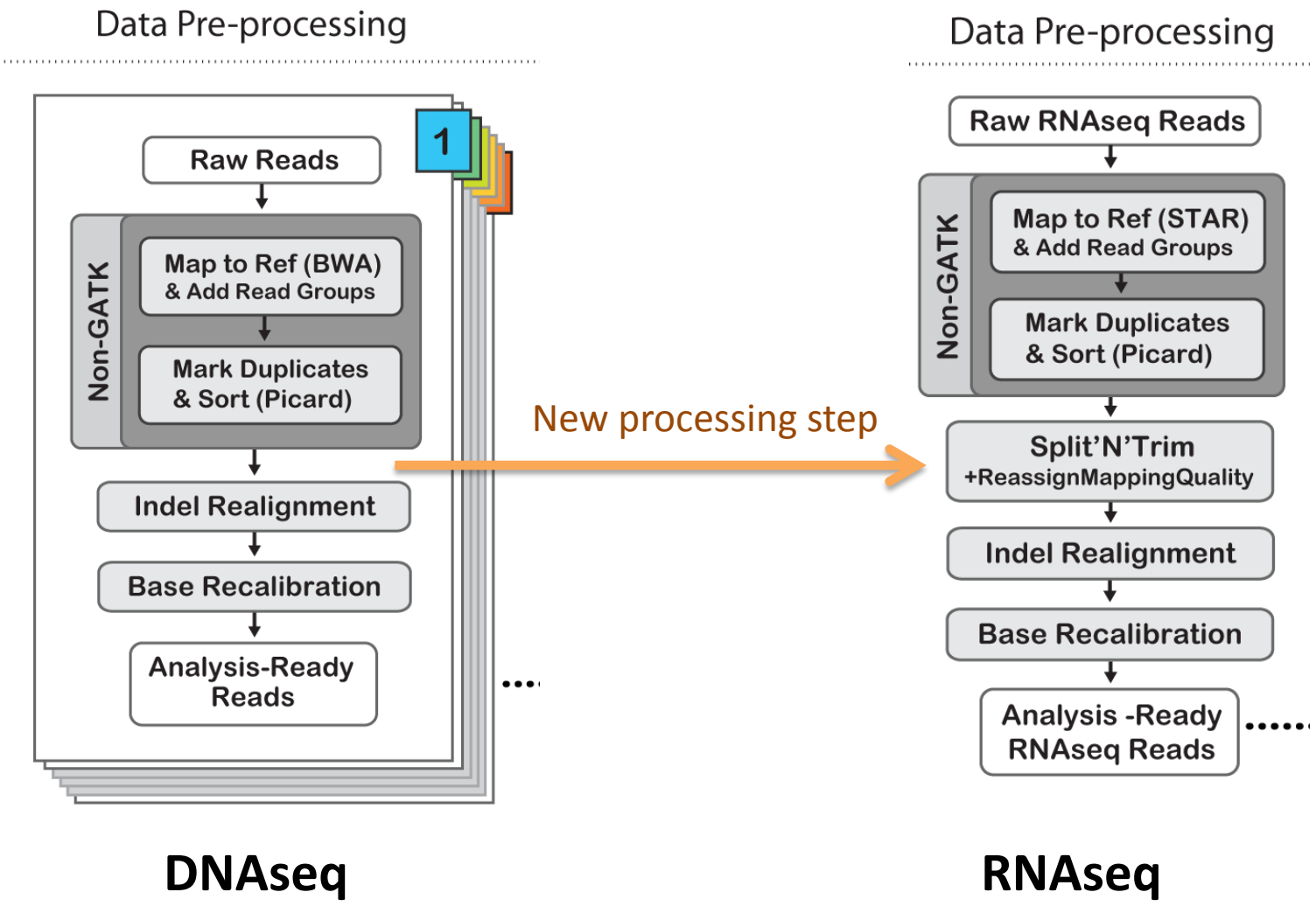
RNAseq needs to be mapped with specific software



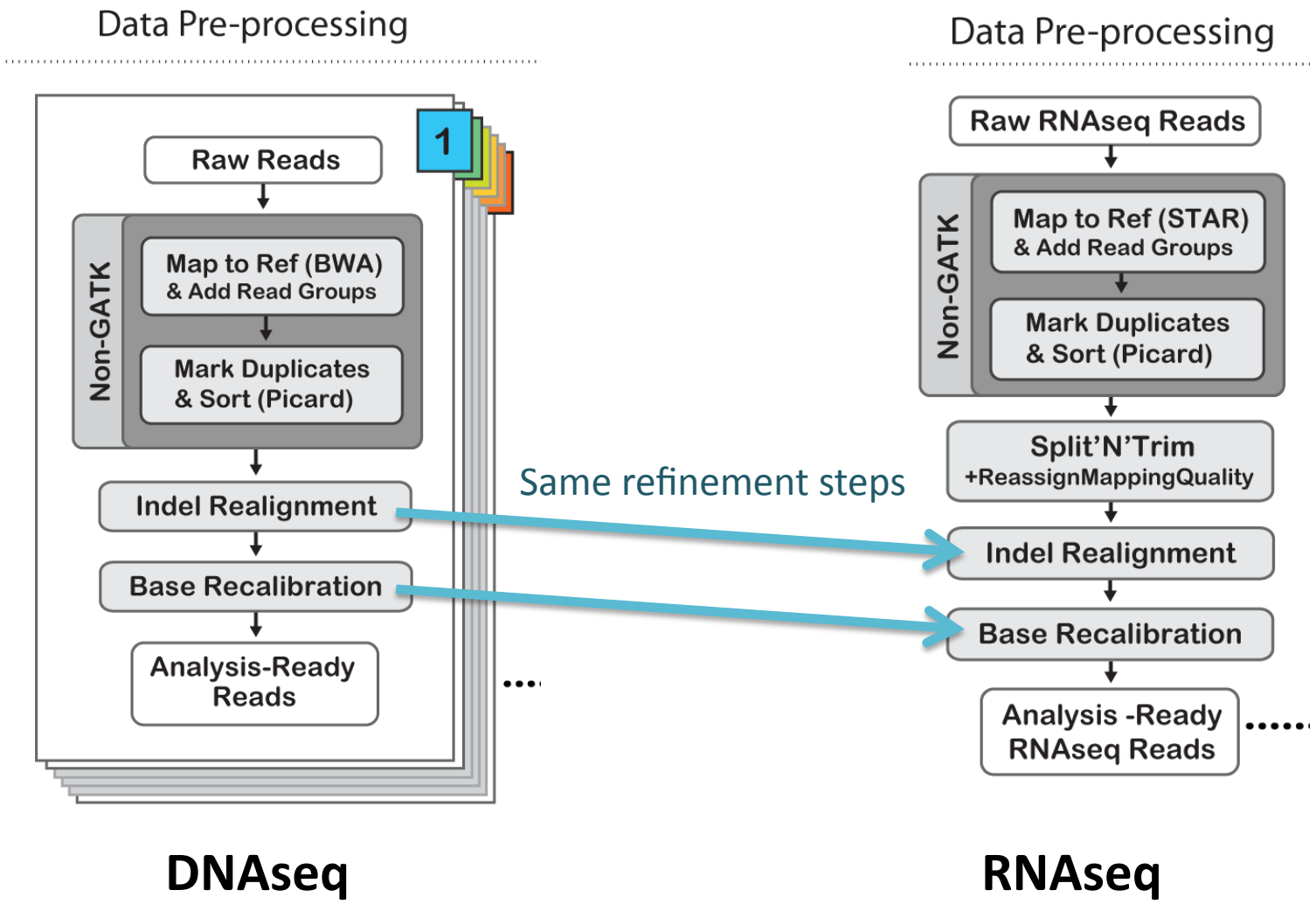
Mark/dedup, sorting etc are the same



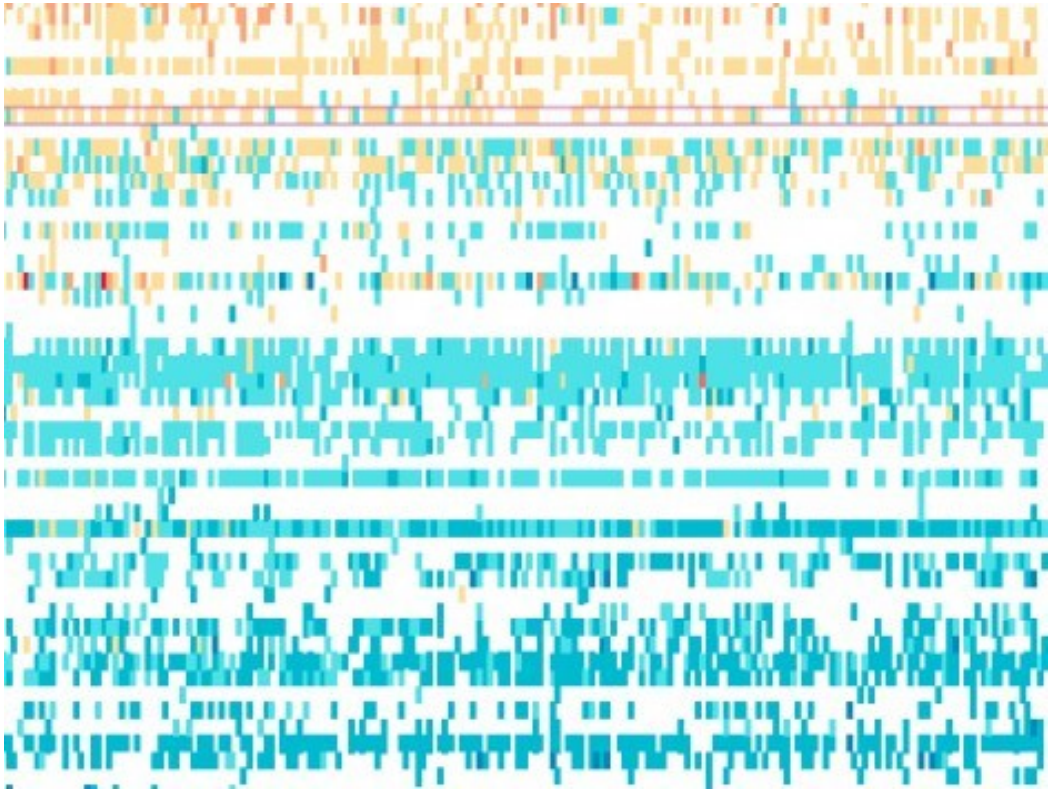
New step specific for RNAseq to deal with splicing junctions



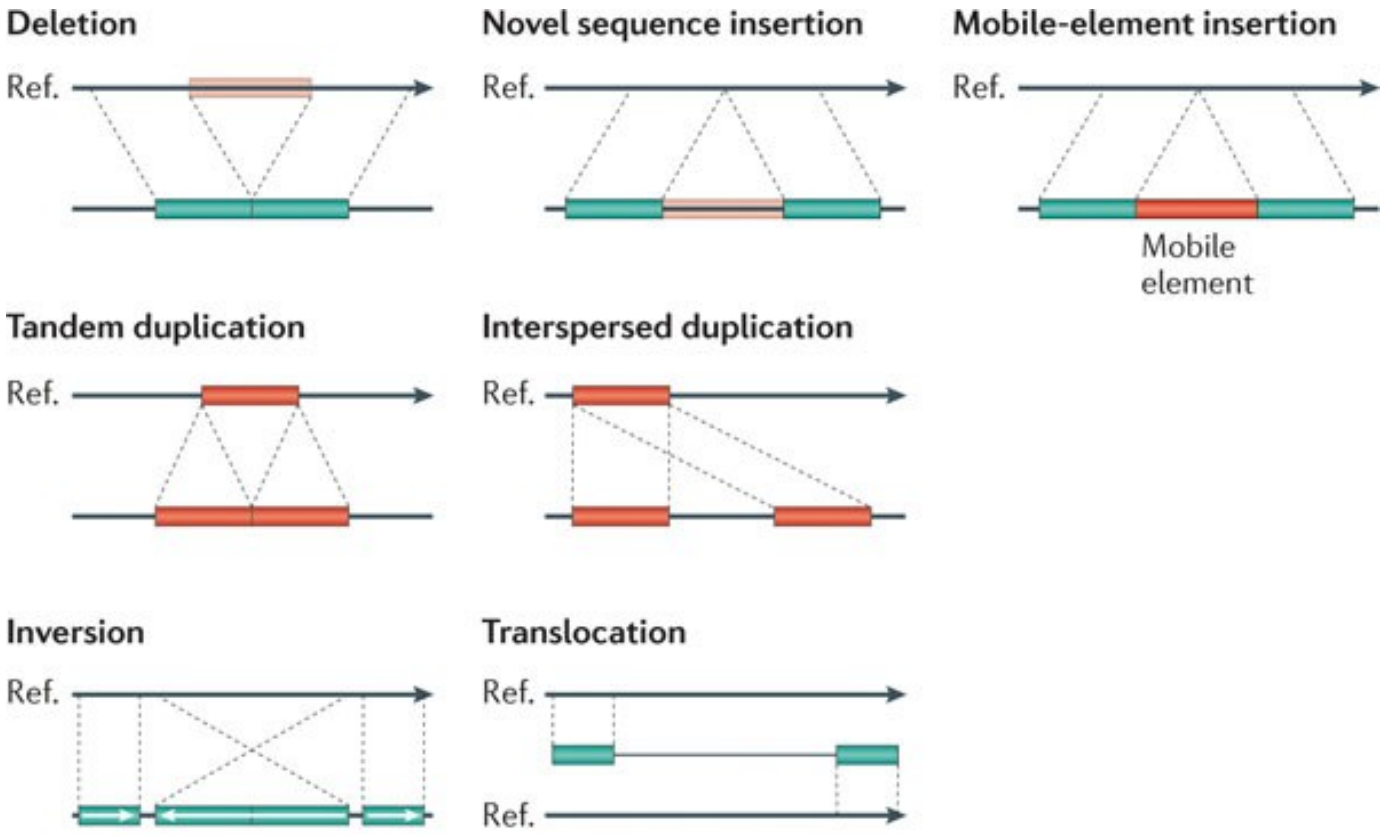
Indel realignment and BQSR are unchanged



Other types of variation

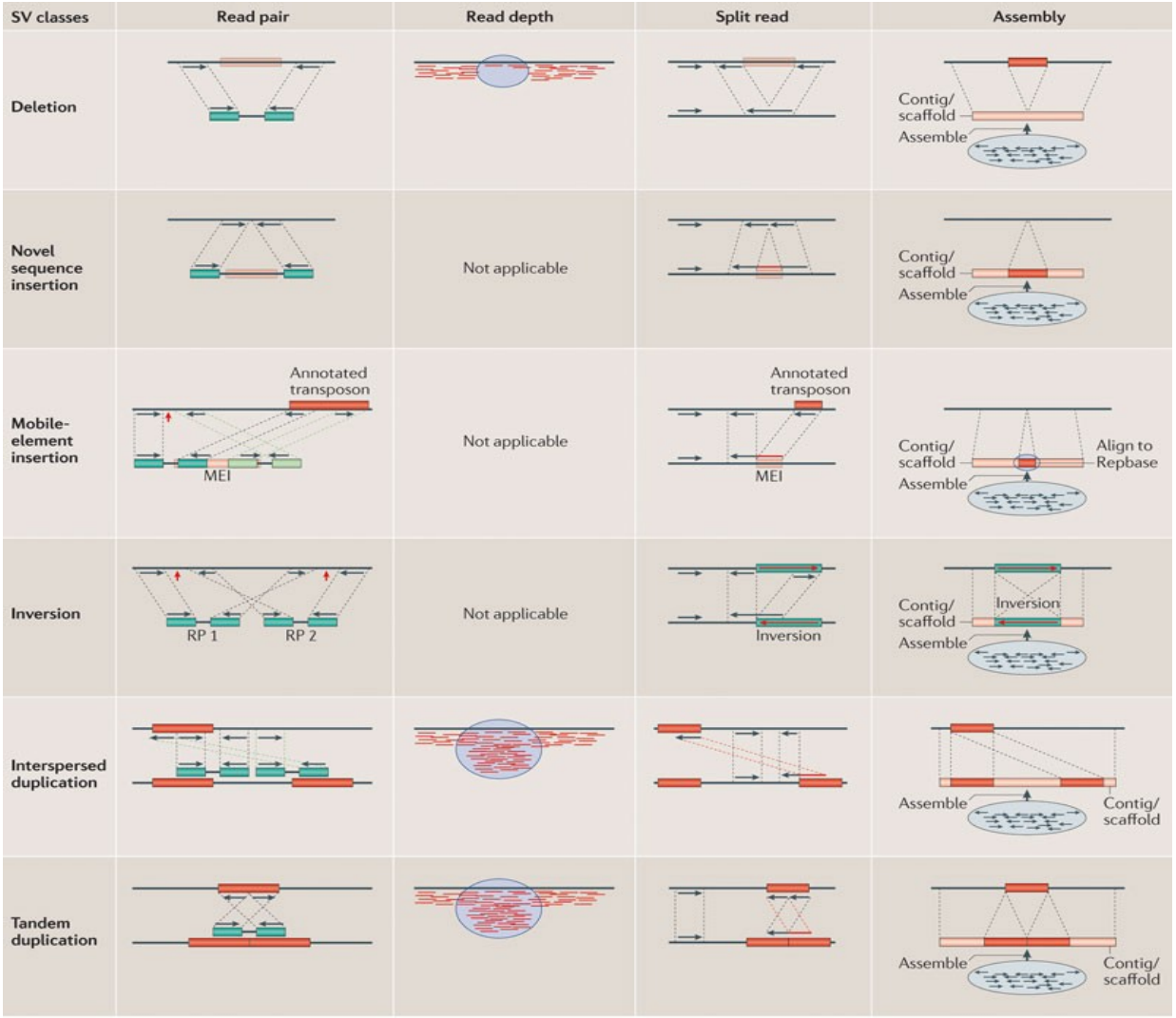


Types of structural variation



Alkan et al. 2011

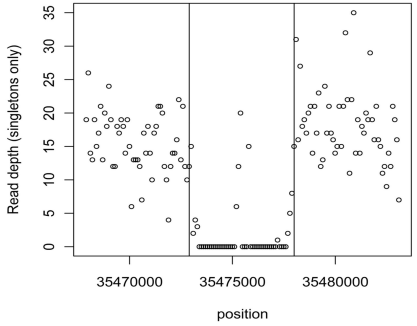
Ho to detect structural variation



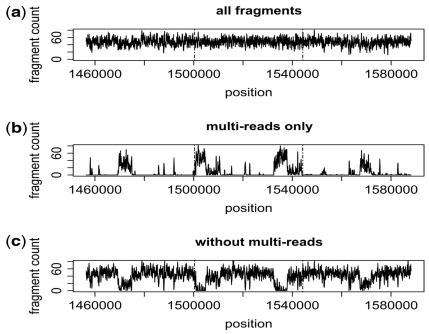
Alkan et al. 2011

Ho to detect copy number variation

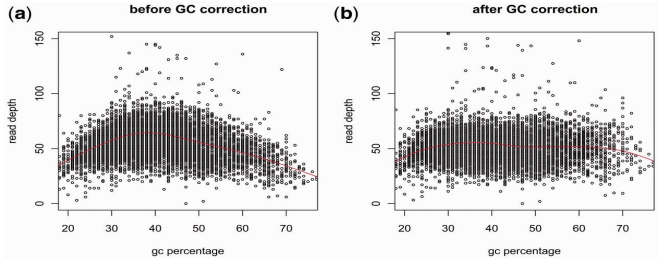
Depth of coverage (DOC)



Duplicated regions (repetitive sequences, PCR)



GC content



Teo et al. 2012

Tools to detect SV and CNV

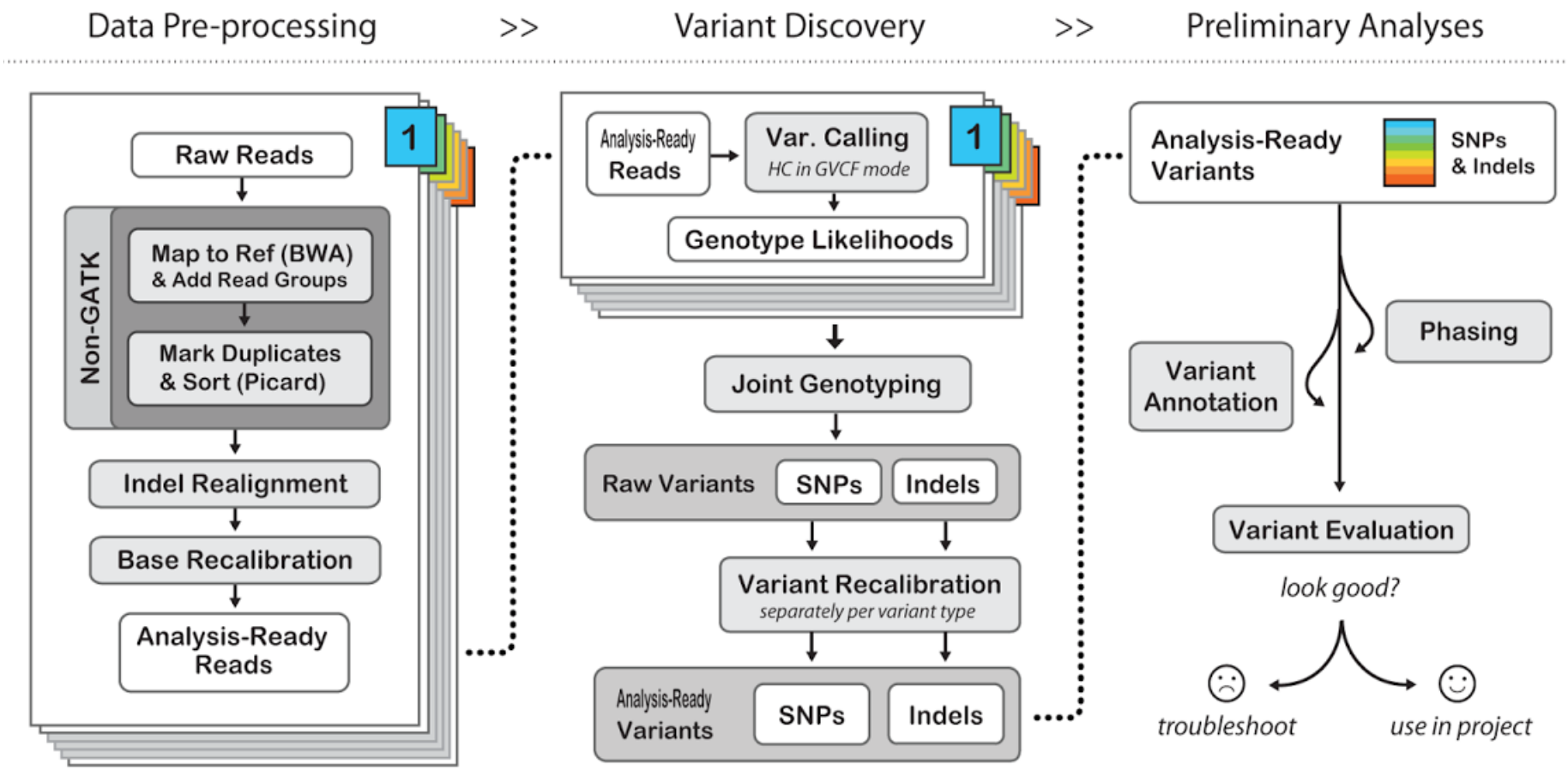
- DELLY
- BreakDancer
- HYDRA
- SOAPsv
- GenomeStrip
- Pindal

Structural variation

- cn.MOPS
- CNVnator
- CNV-seq
- BIC-seq
- Cnv-HMM
- ExomeCopy

Copy number variation

Best Practices for Variant Discovery in DNAseq



Exercise

- Make alignments — BWA/Bowtie2
- Remove duplicates — Samtools
- Sort by coordinates — Samtools
- Call variants — Samtools
- Check VCF — R