Topic 8: Variant Calling

Learning outcomes

- The problem with duplicates and how it's addressed
- Benefits of genotyping with GATK (e.g.the N+1 problem)
- Overview of GATK best practices
- VCF file structure

Alignment + variant calling

1. Get a reference genome
2. Index it
3. Map/align reads to it
4. Mark duplicates
5. Call variants
6. Filter variants

Alignment + variant calling

From the first tutorial

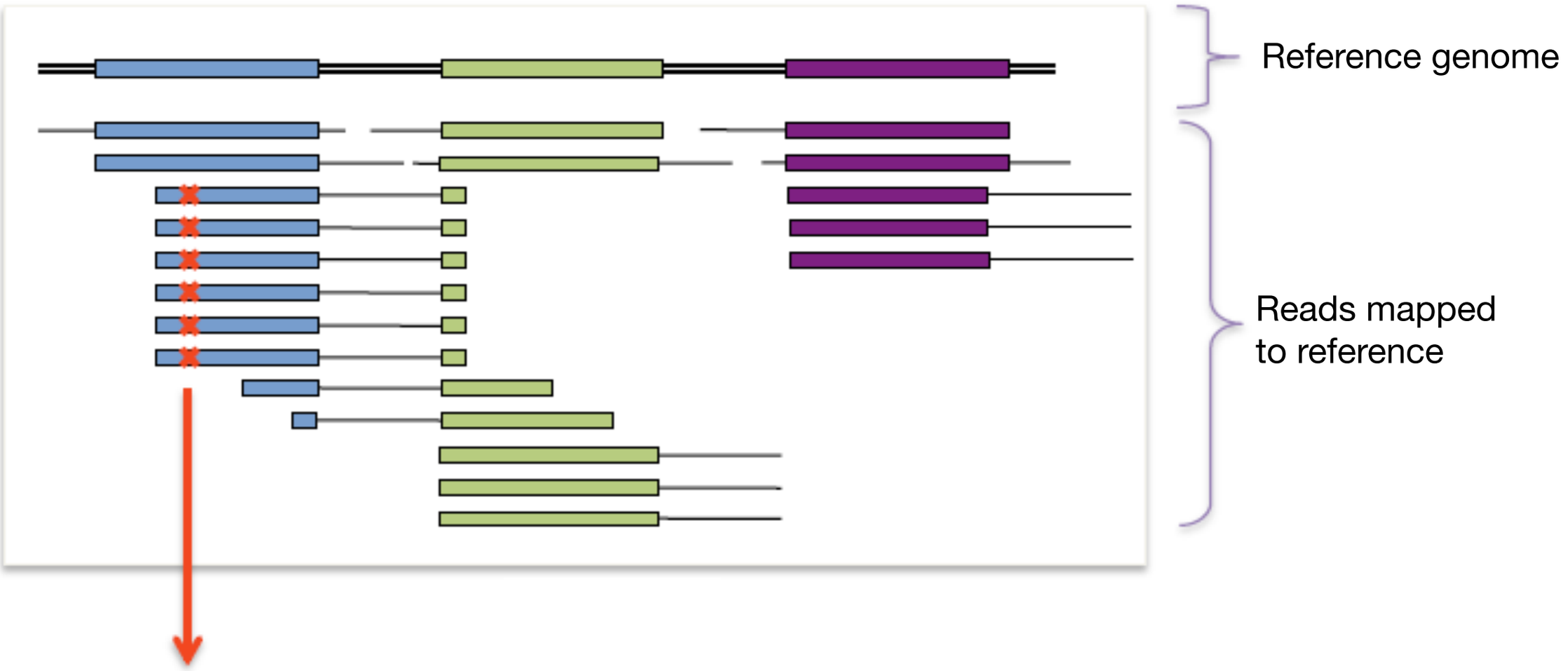
Alignment + variant calling

1. Get a reference genome (we did that already)
2. Index it (and this)
3. Map/align reads to it (and this)
4. **Mark duplicates**
5. **Call variants**
6. Filter variants (next session)

Why would duplicated reads be bad?

Why duplicates are bad for variant calling

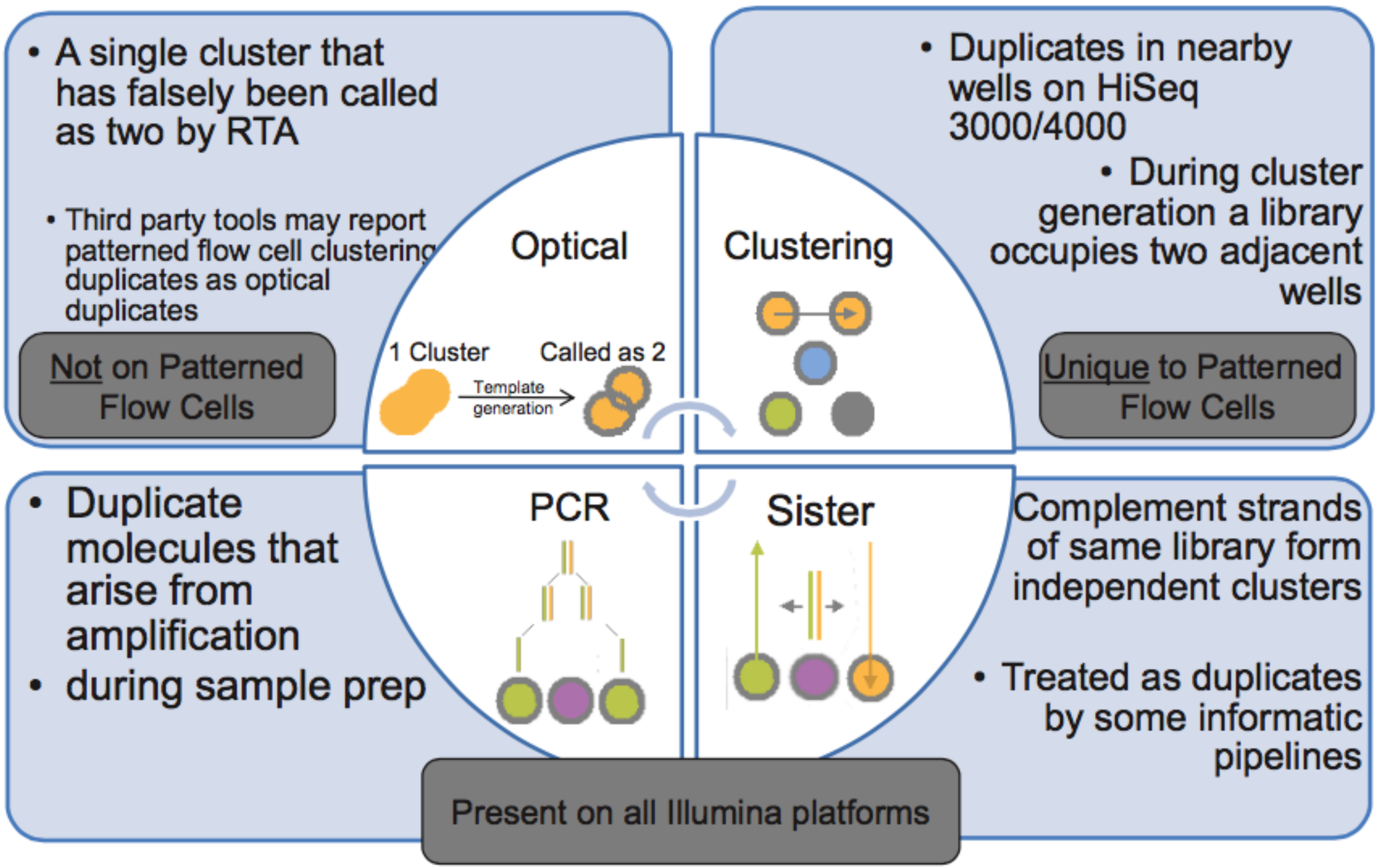
✖ = Sequencing error propagated in duplicates



False variant call (bad)

https://core-genomics.blogspot.com/2016/05/increased-read-duplication-on-patterned.html

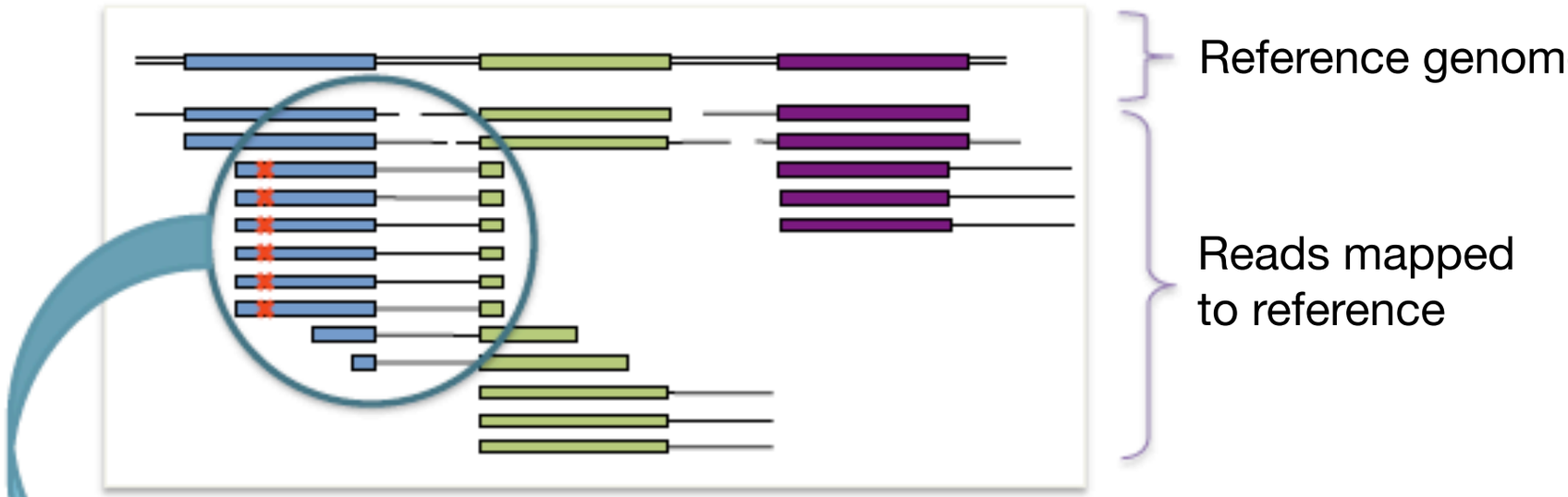
Where does duplication come from?

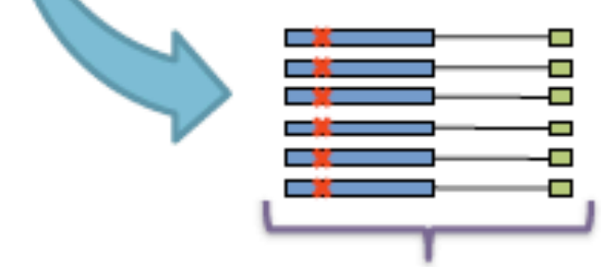


https://core-genomics.blogspot.com/2016/05/increased-read-duplication-on-patterned.html

How to identify duplicates?

Actually pretty easy - duplicated reads would have the same start/end positions and CIGAR strings



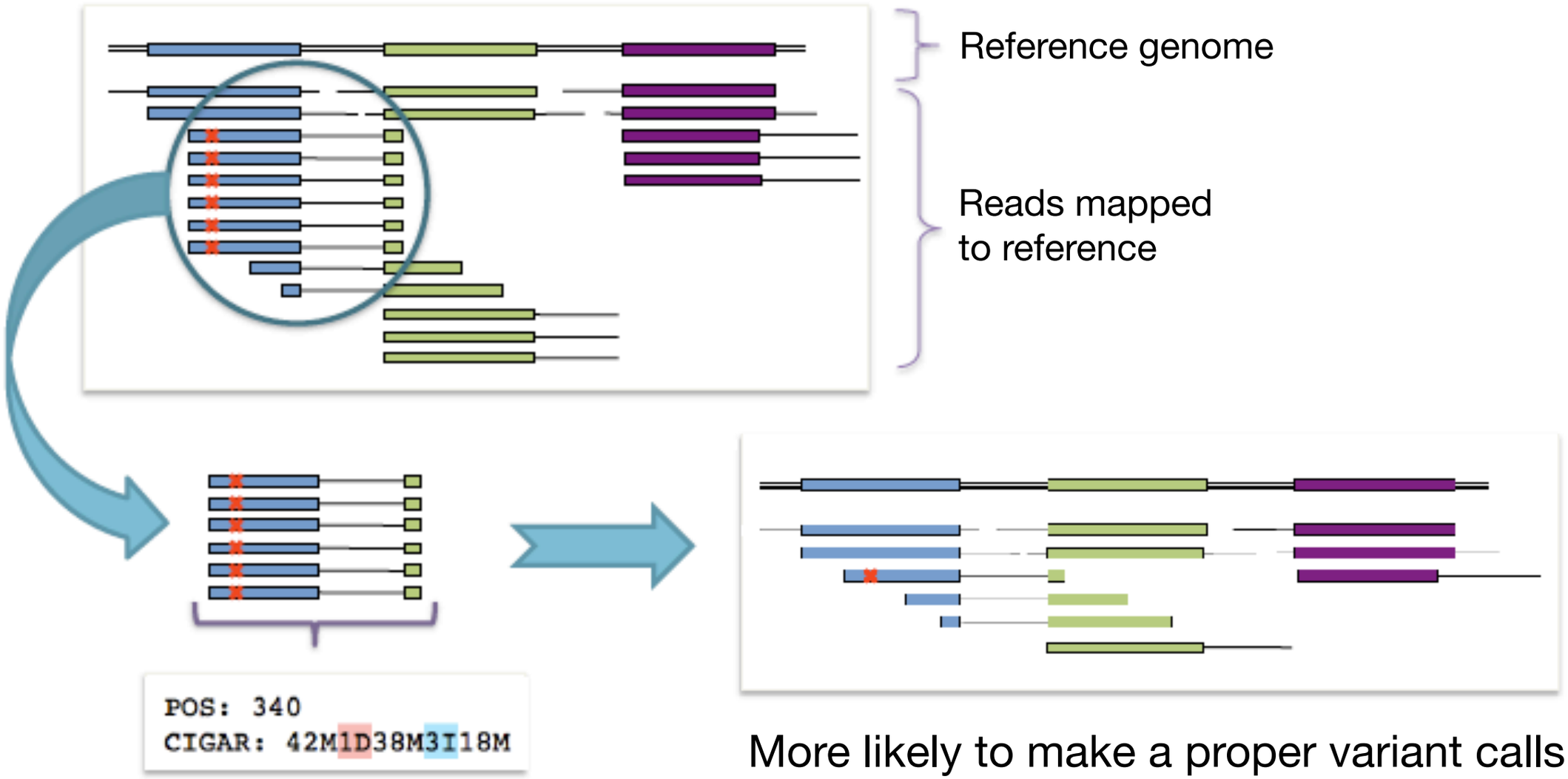


Bag em and tag em

```
POS: 340
CIGAR: 42M1D38M3I18M
```

How to identify duplicates?

Actually pretty easy - duplicated reads would have the same start/end positions and CIGAR strings



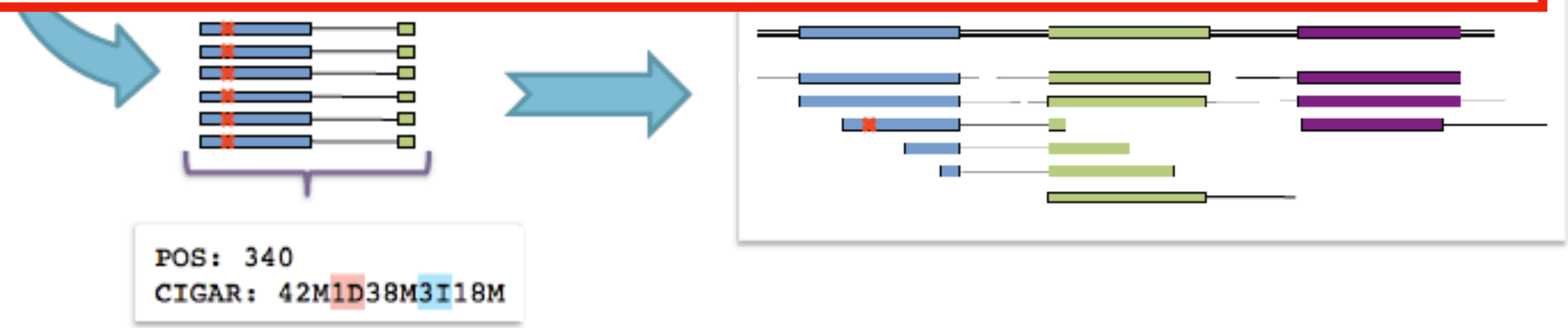
More likely to make a proper variant calls after duplicate removal!

How to identify duplicates?

Actually pretty easy - duplicated reads would have the same start/end positions and CIGAR strings

There are standard tools to remove duplicates as part of SNP calling pipelines

Are there cases where we wouldn't want to do this?



Alignment + variant calling

1. Get a reference genome (we did that already)
2. Index it (and this)
3. Map/align reads to it (and this)
4. **Mark duplicates** ✓
5. **Call variants**
6. Filter variants (next session)

Variant Callers

Research article | Open access | Published: 22 February 2022

Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery

Yury A. Barbitoff, Ruslan Abasov, Varvara E. Tvorogova, Andrey S. Glotov & Alexander V. Predeus

BMC Genomics **23**, Article number: 155 (2022) | Cite this article

There are lots of variant callers out there, with pros and cons

Probably the most widely used method is GATK, which has similar performance to other methods that are perhaps a little slower

Overview of GATK Pipeline

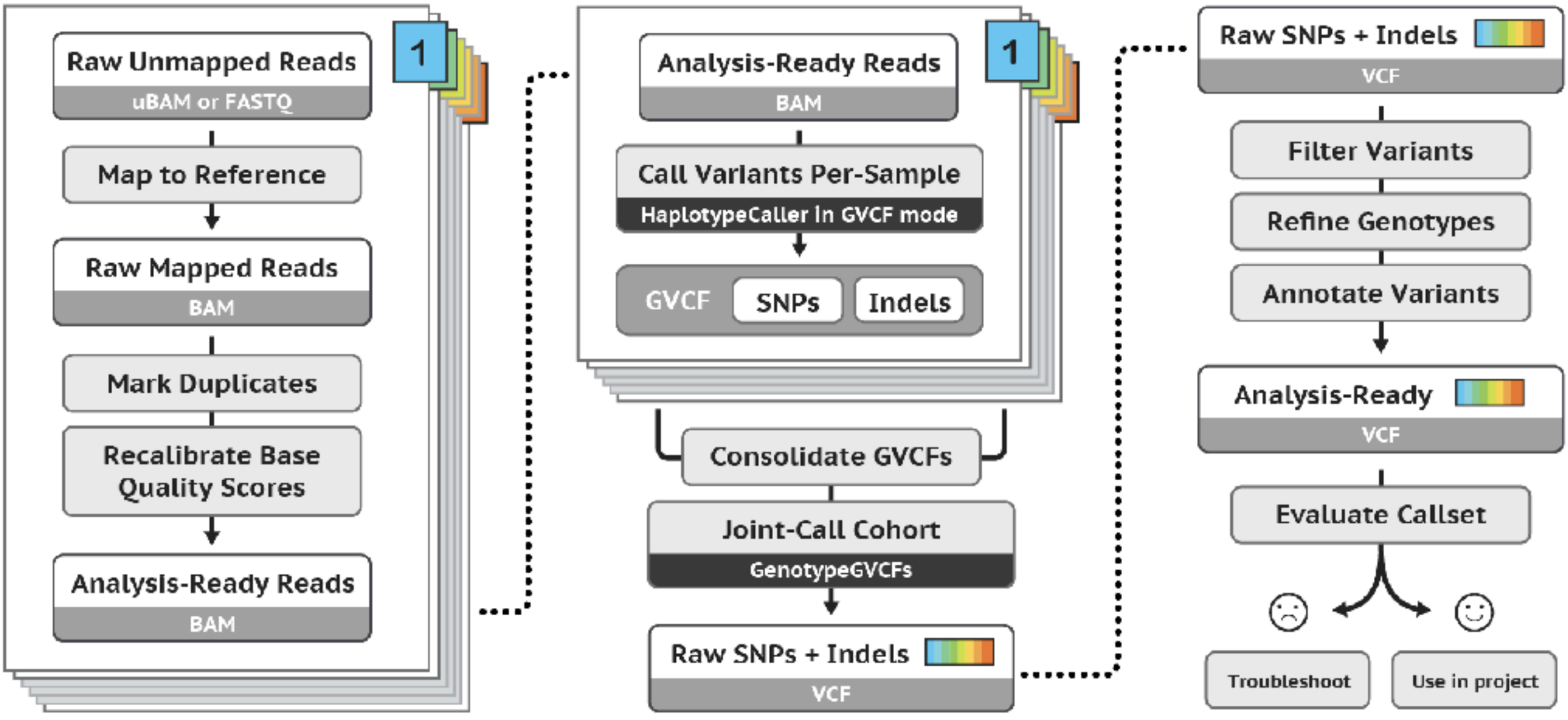


Figure from "*Best Practices for Germline SNP & INDEL Discovery*" Broad Institute 2020

Overview of GATK Pipeline

Base Quality Score Recalibration - a step to adjust the quality scores within the BAM file before attempting to call variants. Not always recommended (particularly with large samples)

It is an attempt to identify and rectify systematic issues in base quality score in the BAM. More details are here:

https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR

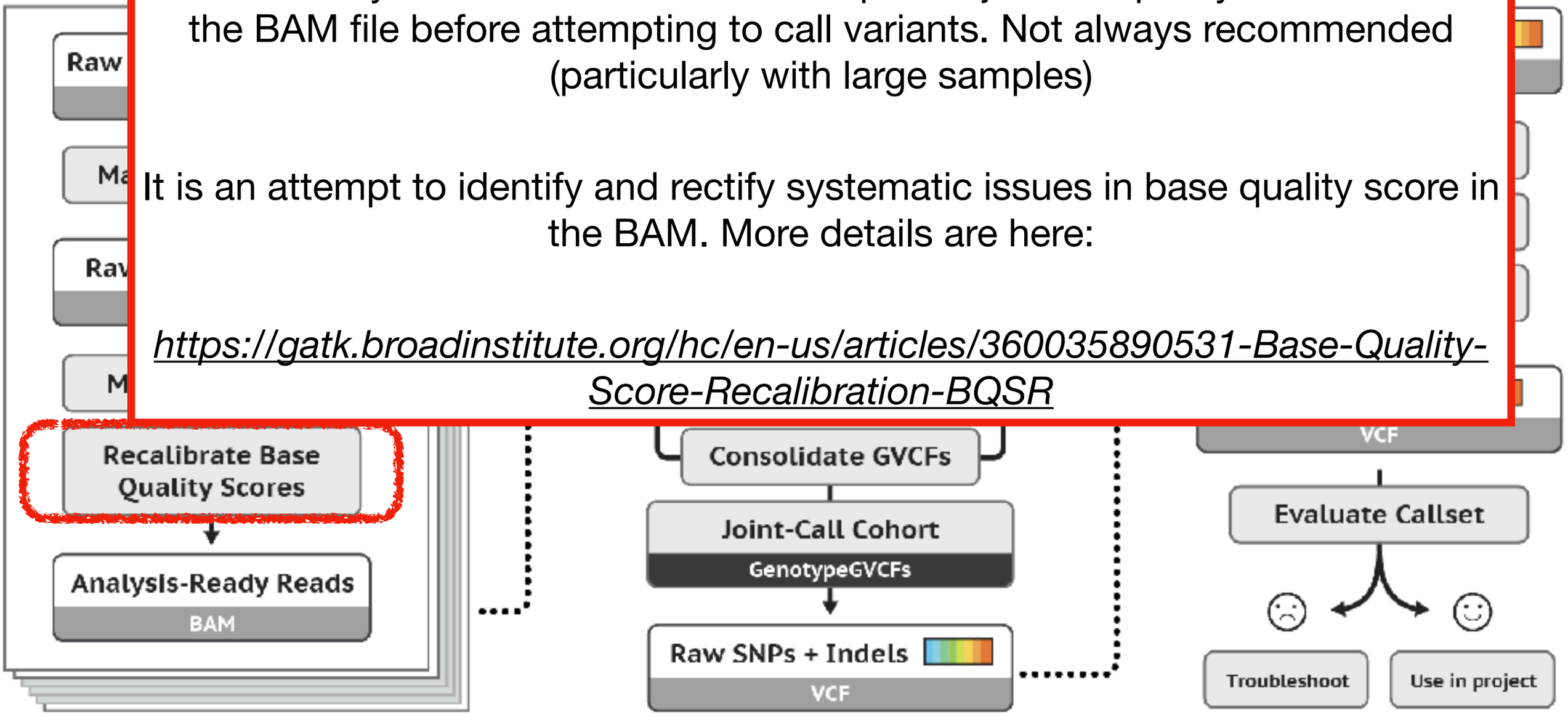


Figure from "*Best Practices for Germline SNP & INDEL Discovery*" Broad Institute 2020

GATK: Variant callers

Unified Genotyper (sunsetted)

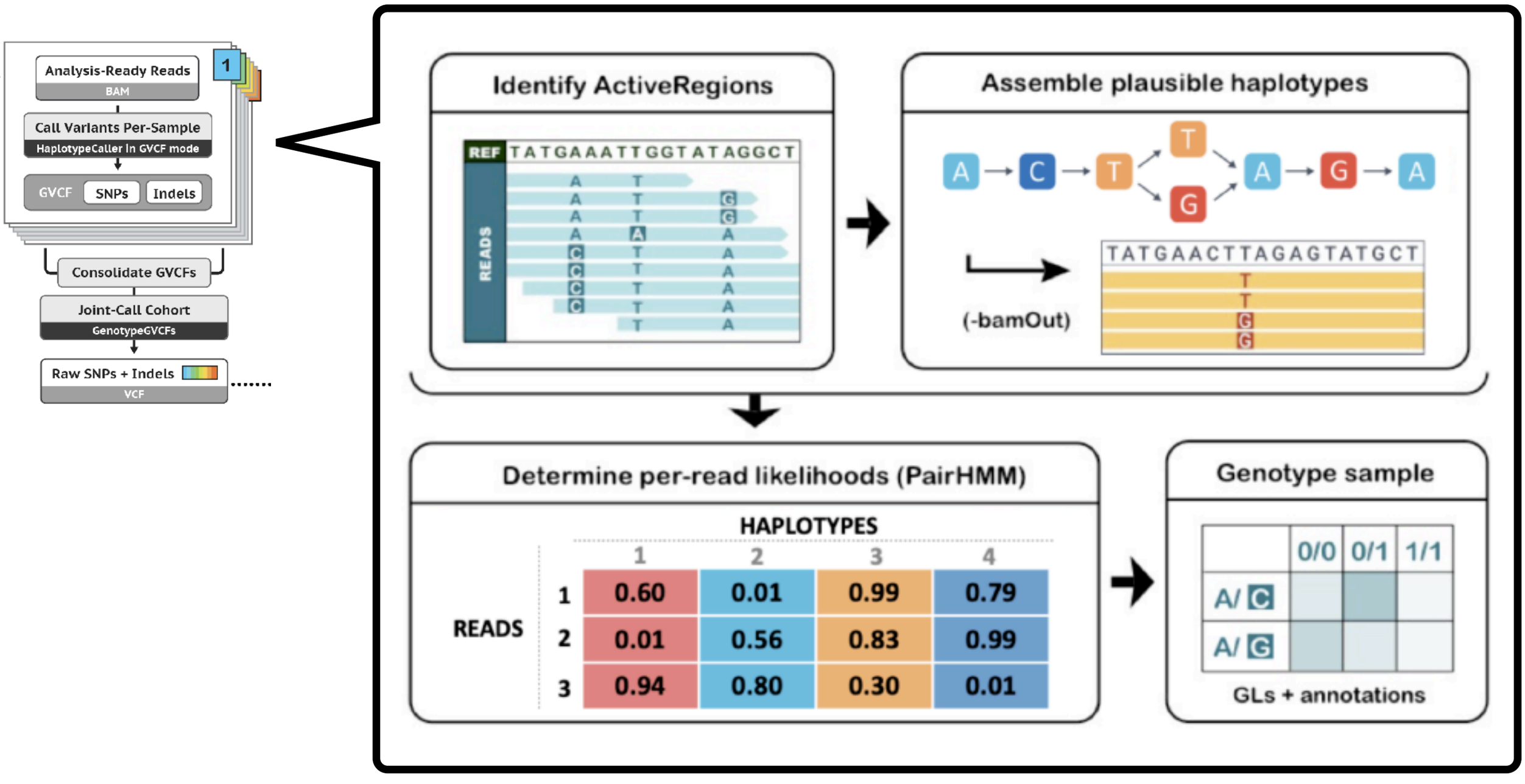
Calls SNPs and in/dels separately

(it did handle multiple ploidy levels and pooled data though)

Haplotype Caller

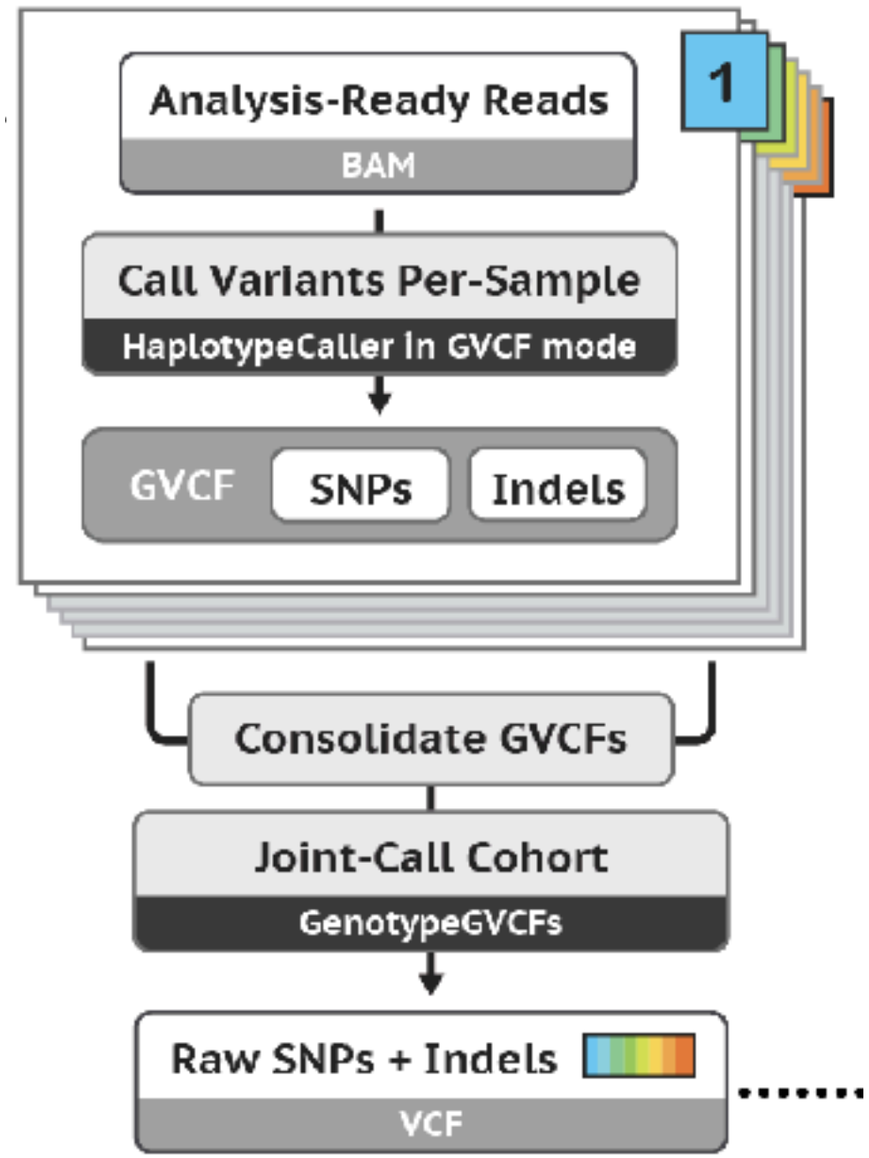
Calls SNPs, in/dels and small structural variants by doing local re-assembly and considering haplotypes

GATK: Haplotype caller



**Is capable of phasing data as well!*

Joint Discovery

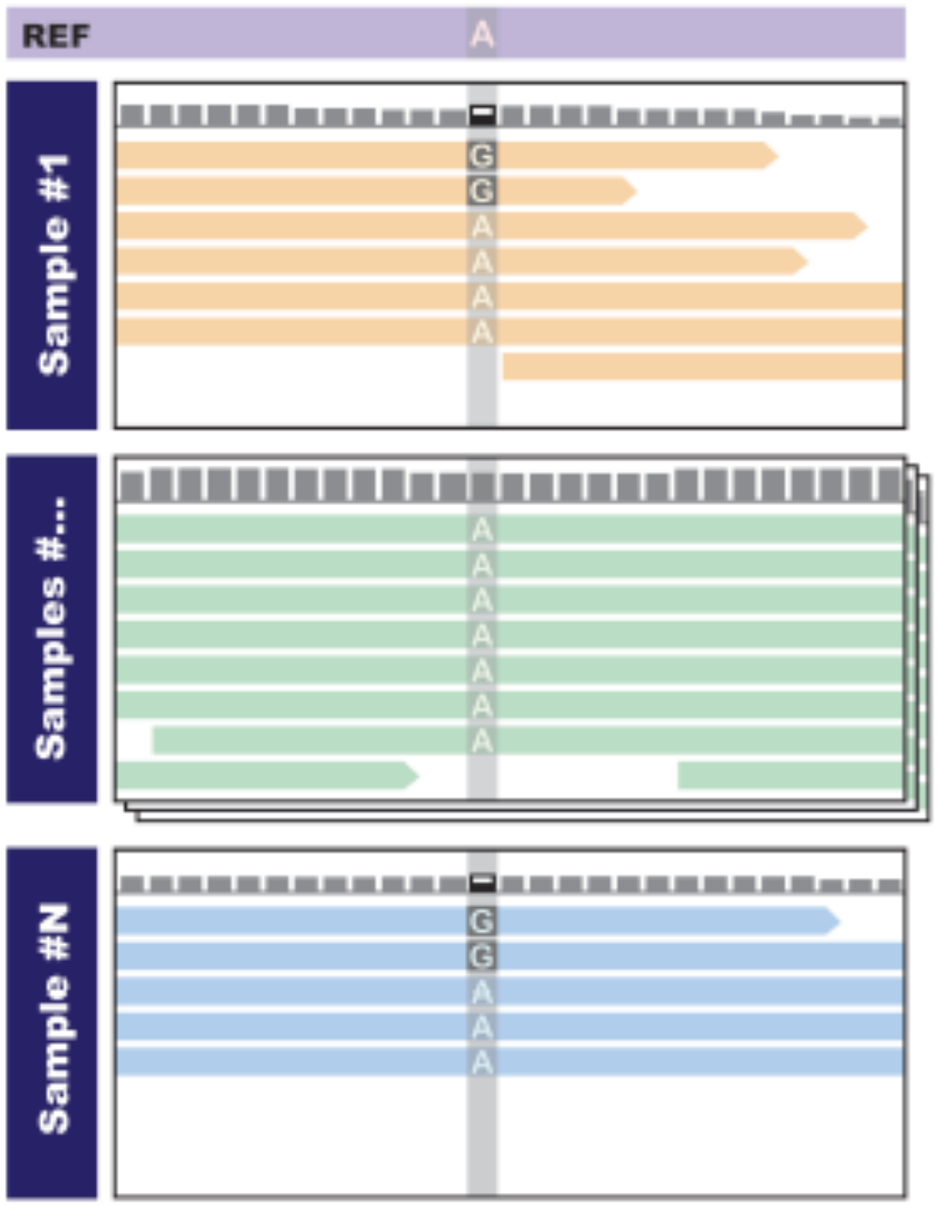


Why analyse the data separately, and then together?

Why would we compare information across samples?

Figure from "*Best Practices for Germline SNP & INDEL Discovery*" Broad Institute 2020

Joint Discovery

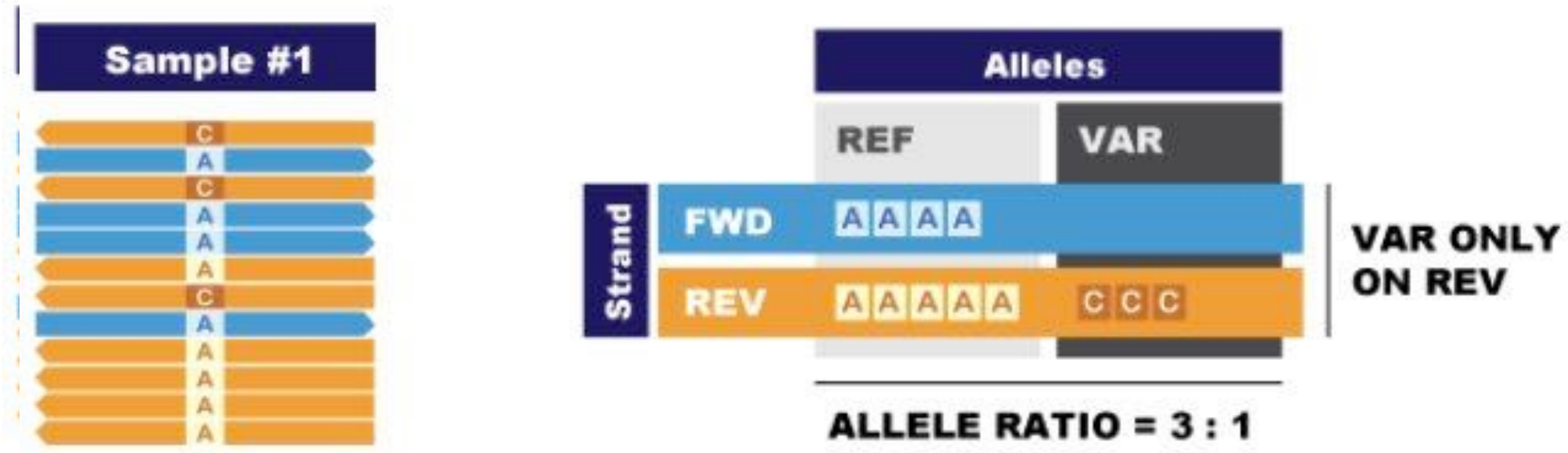


If we analyse Sample #1 or Sample # N alone we may not be confident that the variant is legitimate

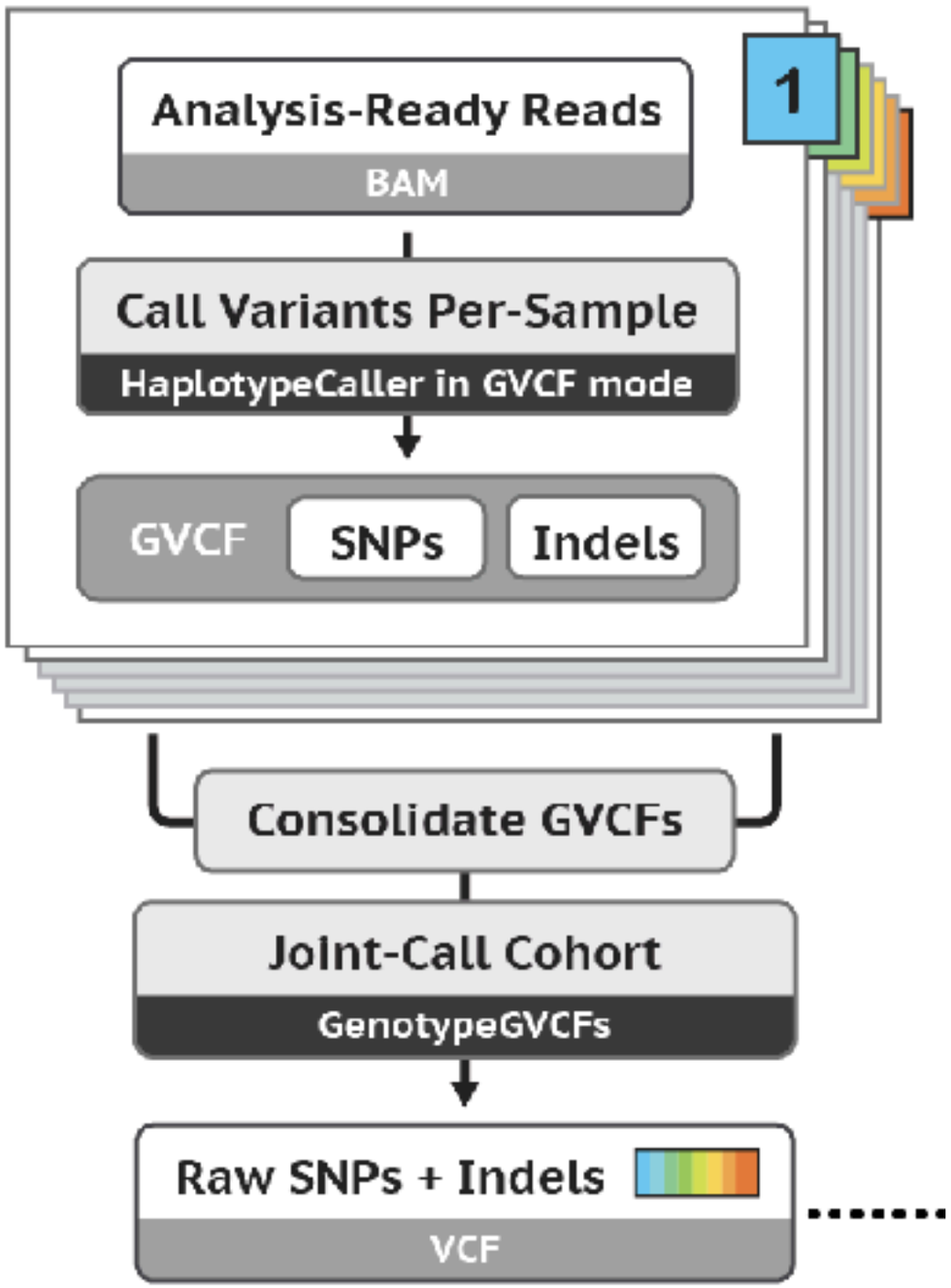
If we see the same variant in multiple individuals we are more confident that it is real and not a sequencing artefact

Joint Discovery

For example, in the case of strand bias



The N+1 Problem



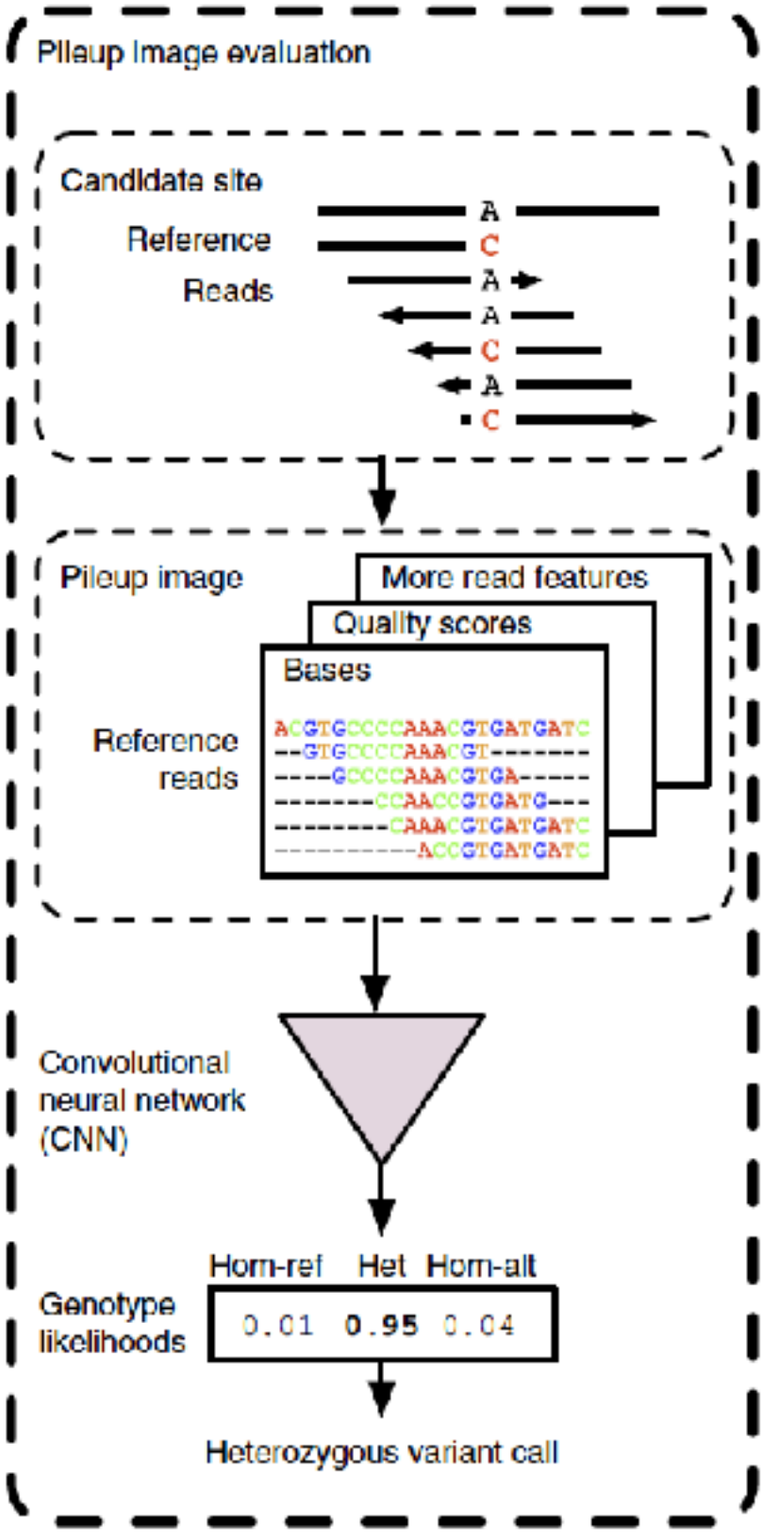
Often, we may receive our genomic data in batches as it is generated by the sequencing facility

Or, extra money is made available so additional samples can be sequenced

Variant calling is computationally intensive so having to re-run from scratch each time additional samples are added would be a waste of resources

This is the N+1 problem

Other Variant Callers

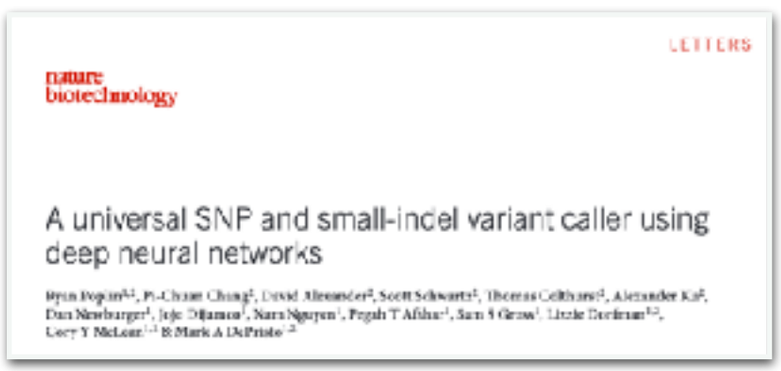


DeepVariant

A Machine Learning based variant caller developed by Google

Uses CNNs trained on images of alignments to identify and call variants and extract statistics

Limited to individual samples or trios

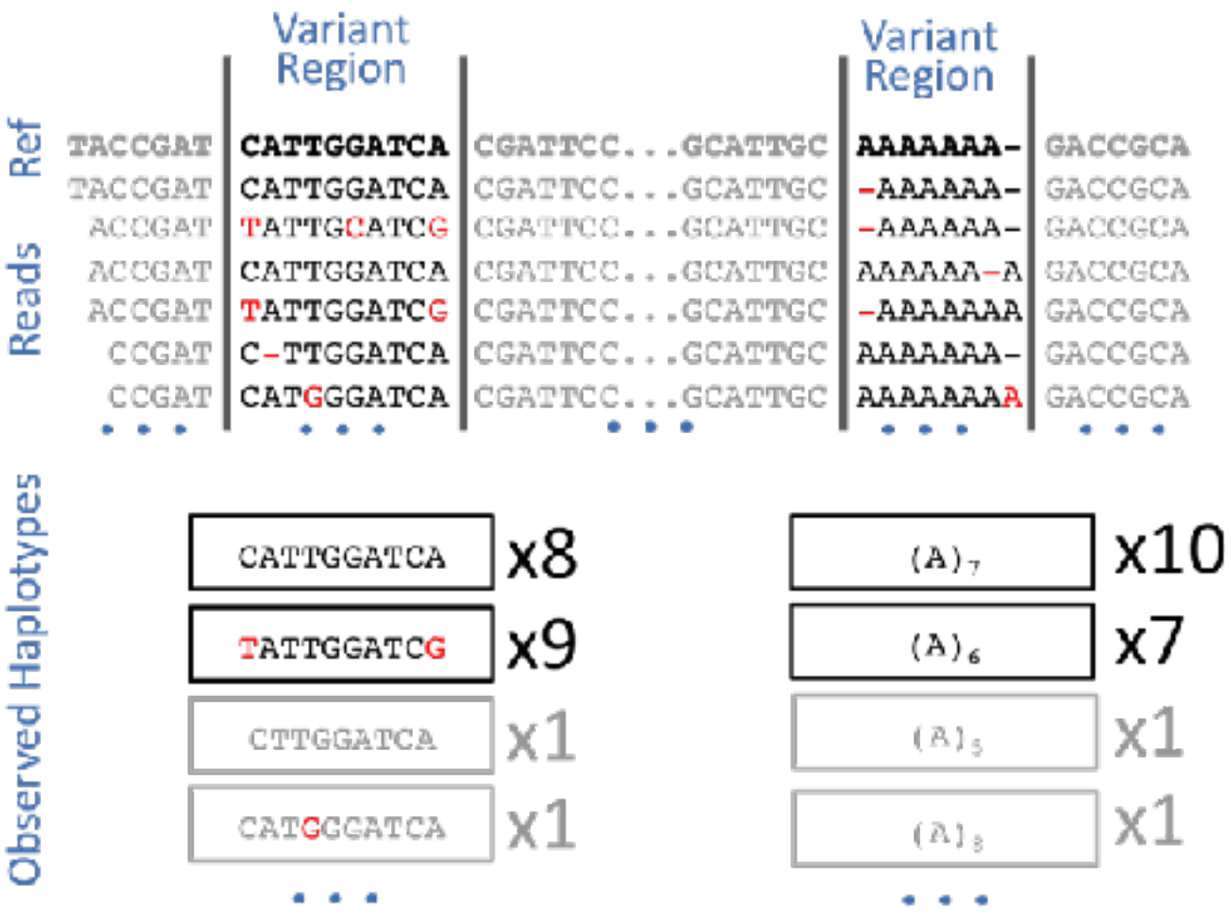

LETTERS

nature biotechnology

A universal SNP and small-indel variant caller using deep neural networks

Other Variant Callers

freebayes



Uses the reads themselves, rather than the alignment of the reads

Generally faster than GATK (more RAM intensive)

No solution to N+1

Provides lots of summary statistics

Garrison and Marth 2012 arXiv

Other Variant Callers

ANGSD

Calls SNPs based on BAM - no realignment

Outputs genotype likelihoods - comes with a suite of analyses that link with it

Recommended for low coverage data (e.g. ancient DNA)

Korneliussen et al 2015 BMC Bioinformatics

Structural Variation

For structural variation, the choice of software is dependant on the data type and the kind of variation you're looking for (e.g. CNVs, inversions or deletions)

Article Open access | Published: 14 March 2024

Benchmarking long-read aligners and SV callers for structural variation detection in Oxford nanopore sequencing data

Asmaa A. Helal, Bishoy T. Saad ✉, Mina T. Saad, Gamal S. Mosaad & Khaled M. Aboshanab ✉

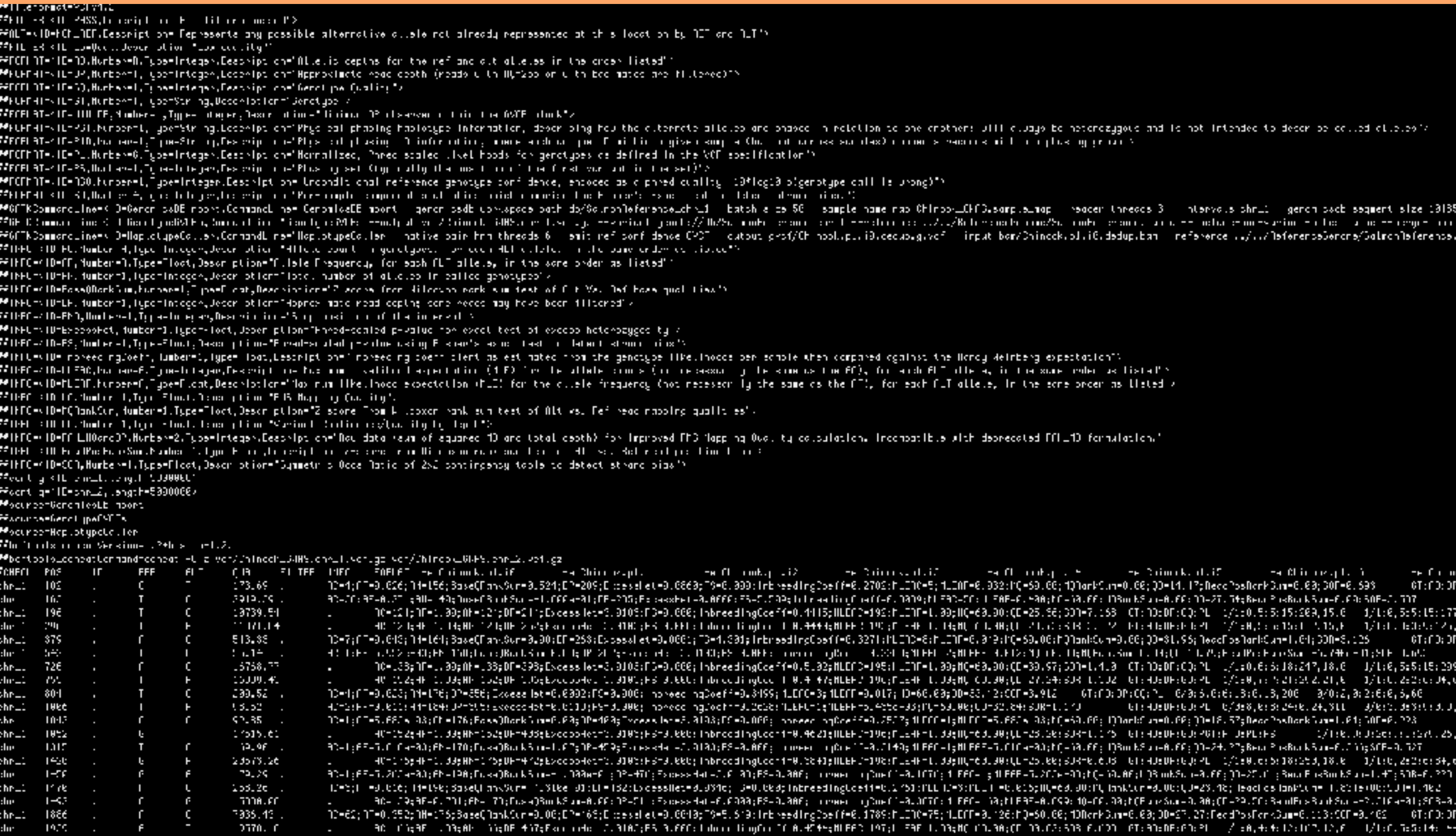
Scientific Reports **14**, Article number: 6160 (2024) | Cite this article

Comparison of structural variant callers for massive whole-genome sequence data

Soobok Joe, Jong-Lyul Park, Jun Kim, Sangok Kim, Ji-Hwan Park, Min-Kyung Yeo, Dongyoon Lee, Jin Ok Yang ✉ & Seon-Young Kim ✉

BMC Genomics **25**, Article number: 318 (2024) | Cite this article

The Variant Call Format



The Variant Call Format

VCF: Header

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele not already represented at this
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad m
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=MIN_DP,Number=1,Type=Integer,Description="Minimum DP observed within the GVCF block">
##FORMAT=<ID=PGT,Number=1,Type=String,Description="Physical phasing haplotype information, describing how
phased in relation to one another; will always be heterozygous and is not intended to describe called alle
##FORMAT=<ID=PID,Number=1,Type=String,Description="Physical phasing ID information, where each unique ID w
not across samples) connects records within a phasing group">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as d
specification">
##FORMAT=<ID=PS,Number=1,Type=Integer,Description="Phasing set (typically the position of the first variar
##FORMAT=<ID=RGQ,Number=1,Type=Integer,Description="Unconditional reference genotype confidence, encoded a
p(genotype call is wrong)">
##FORMAT=<ID=SB,Number=4,Type=Integer,Description="Per-sample component statistics which comprise the Fish
strand bias.">
```

Contains detailed information on what each column contains, the file version, commands used to generate file etc.

Lines starting with ##

VCF: Records

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele not already represented at this location by REF and ALT">
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=MIN_DP,Number=1,Type=Integer,Description="Minimum DP observed within the GVCF block">
##FORMAT=<ID=PGT,Number=1,Type=String,Description="Physical phasing haplotype information, describing how the alternate alleles are phased in
be heterozygous and is not intended to describe called alleles">
##FORMAT=<ID=PID,Number=1,Type=String,Description="Physical phasing ID information, where each unique ID within a given sample (but not acros
phasing group">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##FORMAT=<ID=PS,Number=1,Type=Integer,Description="Phasing set (typically the position of the first variant in the set)">
##FORMAT=<ID=RGQ,Number=1,Type=Integer,Description="Unconditional reference genotype confidence, encoded as a phred quality -10*log10 p(genot
##FORMAT=<ID=SB,Number=4,Type=Integer,Description="Per-sample component statistics which comprise the Fisher's Exact Test to detect strand bi
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0
```

Contains detailed information on what each column contains, the file version, commands used to generate file etc.

VCF: Records

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0        -e Chinook.p1.i1
chr_1   102     .       C       T       173.69  .
AC=4;AF=0.026;AN=156;BaseQRankSum=0.524;DP=209;ExcessHet=0.0860;FS=0.000;InbreedingCoeff=0.2702;MLEAC=5;MLEAF=0.032;MQ=60.00;MQRankSum=0.0
0;QD=14.47;ReadPosRankSum=0.00;SOR=0.693      GT:AD:DP:GQ:PL  0/0:3,0:3:9:0,9,102     0/0:2,0:2:6

chr_1   163     .       T       C       2919.39 .
AC=38;AF=0.271;AN=140;BaseQRankSum=-1.800e-01;DP=235;ExcessHet=0.0000;FS=5.509;InbreedingCoeff=0.3039;MLEAC=56;MLEAF=0.400;MQ=60.00;MQRank
Sum=0.00;QD=27.54;ReadPosRankSum=0.00;SOR=3.587       GT:AD:DP:GQ:PL  0/0:4,0:4:12:0,12,144
```

Records for two SNPs

How do we know that they are SNPs?

VCF: Records - INFO

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0        -e Chinook.p1.i1
chr_1   102     .       C       T       173.69  .
AC=4;AF=0.026;AN=156;BaseQRankSum=0.524;DP=209;ExcessHet=0.0860;FS=0.000;InbreedingCoeff=0.2
702;MLEAC=5;MLEAF=0.032;MQ=60.00;MQRankSum=0.00;QD=14.47;ReadPosRankSum=0.00;SOR=0.693
GT:AD:DP:GQ:PL  0/0:3,0:3:9:0,9,102     0/0:2,0:2:6
```

```
AC=4
AF=0.026
AN=156
BaseQRankSum=0.524
DP=209
ExcessHet=0.0860
FS=0.000
InbreedingCoeff=0.2702
MLEAC=5
MLEAF=0.032
MQ=60.00
MQRankSum=0.00
QD=14.47
ReadPosRankSum=0.00
SOR=0.693
```

Semi-colon separated data held the INFO field

VCF: Records - INFO

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0        -e Chinook.p1.i1
chr_1   102     .       C       T       173.69  .
AC=4;AF=0.026;AN=156;BaseQRankSum=0.524;DP=209;ExcessHet=0.0860;FS=0.000;InbreedingCoeff=0.2
702;MLEAC=5;MLEAF=0.032;MQ=60.00;MQRankSum=0.00;QD=14.47;ReadPosRankSum=0.00;SOR=0.693
GT:AD:DP:GQ:PL  0/0:3,0:3:9:0,9,102     0/0:2,0:2:6
```

```
AC=4
AF=0.026
AN=156
BaseQRankSum=0.524
DP=209
ExcessHet=0.0860
FS=0.000
InbreedingCoeff=0.2702
MLEAC=5
MLEAF=0.032
MQ=60.00
MQRankSum=0.00
QD=14.47
ReadPosRankSum=0.00
SOR=0.693
```

Semi-colon separated data held the INFO field

```
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in g
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, f
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of a
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="Z-score f
##INFO=<ID=DP,Number=1,Type=Integer,Description="Approximate read
##INFO=<ID=END,Number=1,Type=Integer,Description="Stop position of
##INFO=<ID=ExcessHet,Number=1,Type=Float,Description="Phred-scaled
##INFO=<ID=FS,Number=1,Type=Float,Description="Phred-scaled p-valu
##INFO=<ID=InbreedingCoeff,Number=1,Type=Float,Description="Inbree
per-sample when compared against the Hardy-Weinberg expectation">
##INFO=<ID=MLEAC,Number=A,Type=Integer,Description="Maximum likeli
the same as the AC), for each ALT allele, in the same order as lis
##INFO=<ID=MLEAF,Number=A,Type=Float,Description="Maximum likeliho
the same as the AF), for each ALT allele, in the same order as lis
##INFO=<ID=MQ,Number=1,Type=Float,Description="RMS Mapping Quality
##INFO=<ID=MQRankSum,Number=1,Type=Float,Description="Z-score From
##INFO=<ID=QD,Number=1,Type=Float,Description="Variant Confidence/
##INFO=<ID=RAW_MQandDP,Number=2,Type=Integer,Description="Raw data
Quality calculation. Incompatible with deprecated RAW_MQ formulati
##INFO=<ID=ReadPosRankSum,Number=1,Type=Float,Description="Z-score
bias">
##INFO=<ID=SOR,Number=1,Type=Float,Description="Symmetric Odds Rat
```

The Key to the INFO Field is in the header

VCF: Records - FORMAT

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0        -e Chinook.p1.i1
chr_1   102     .       C       T       173.69  .
AC=4;AF=0.026;AN=156;BaseQRankSum=0.524;DP=209;ExcessHet=0.0860;FS=0.000;InbreedingCoeff=0.2702;MLEAC=5;MLEAF=0.032;MQ=60.00;MQ
RankSum=0.00;QD=14.47;ReadPosRankSum=0.00;SOR=0.693     GT:AD:DP:GQ:PL  0/0:3,0:3:9:0,9,102     0/0:2,0:2:6
```

`GT:AD:DP:GQ:PL`

Colon separated key to the data in the column for each sample

`0/0:3,0:3:9:0,9,102`

Colon separated data for sample “Chinook.p1.i0”

VCF: Records - FORMAT

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  -e Chinook.p1.i0        -e Chinook.p1.i1
chr_1   102     .       C       T       173.69  .
AC=4;AF=0.026;AN=156;BaseQRankSum=0.524;DP=209;ExcessHet=0.0860;FS=0.000;InbreedingCoeff=0.2702;MLEAC=5;MLEAF=0.032;MQ=60.00;MQ
RankSum=0.00;QD=14.47;ReadPosRankSum=0.00;SOR=0.693     GT:AD:DP:GQ:PL  0/0:3,0:3:9:0,9,102     0/0:2,0:2:6
```

`GT:AD:DP:GQ:PL`

`0/0:3,0:3:9:0,9,102`

Colon separated key to the data in the column for each sample

Colon separated data for sample “Chinook.p1.i0”

The Key to abbreviations in the FORMAT field is in the header

Work through the tutorial associated with this session