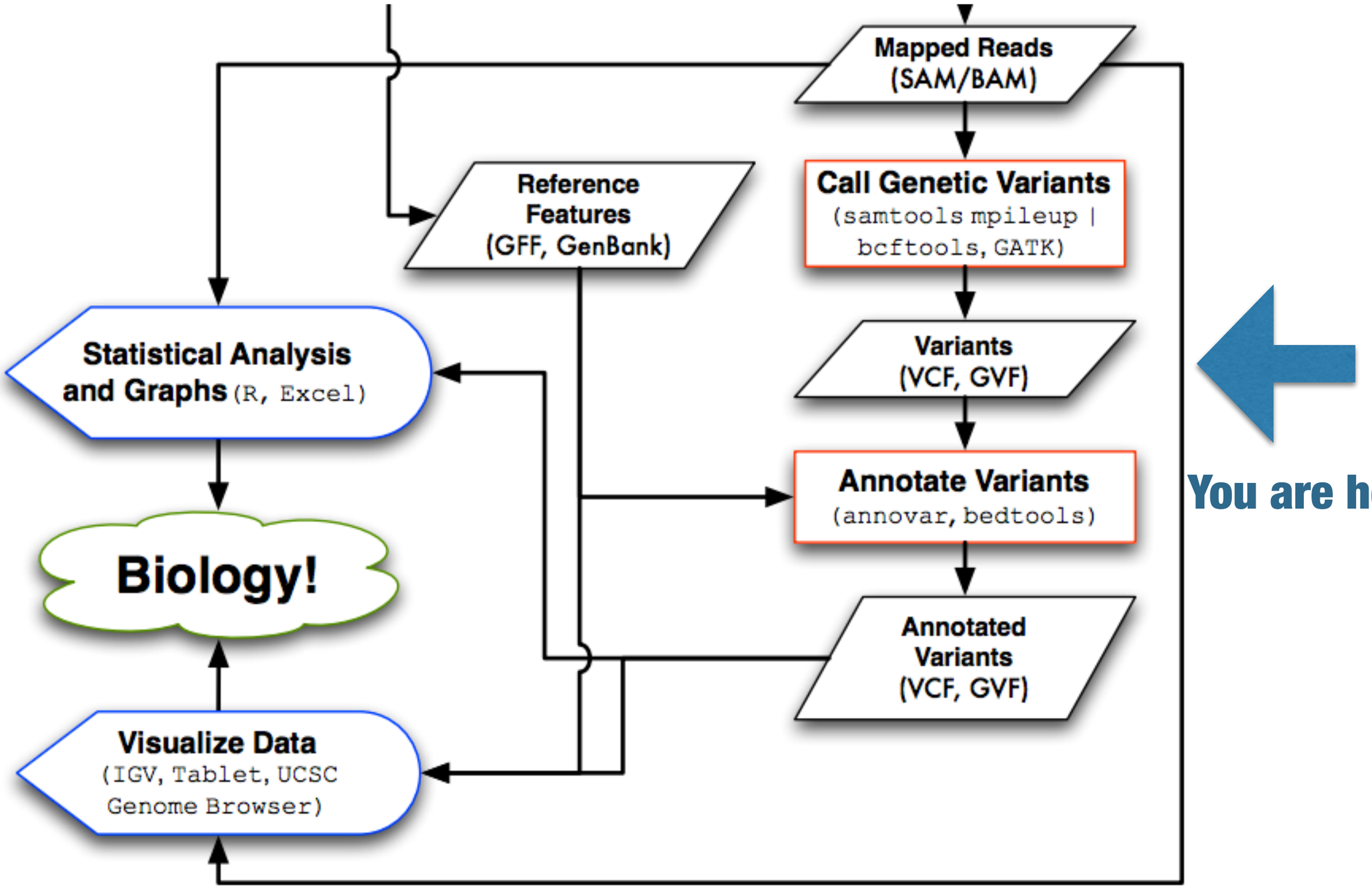
GENOMIKA

# Processamento Pós-alinhamento & Chamada de Variantes

Marcel Caraciolo
marcel@genomika.com.br

# Pipeline




You are here!

# Computational steps for calling genetic variation from NGS

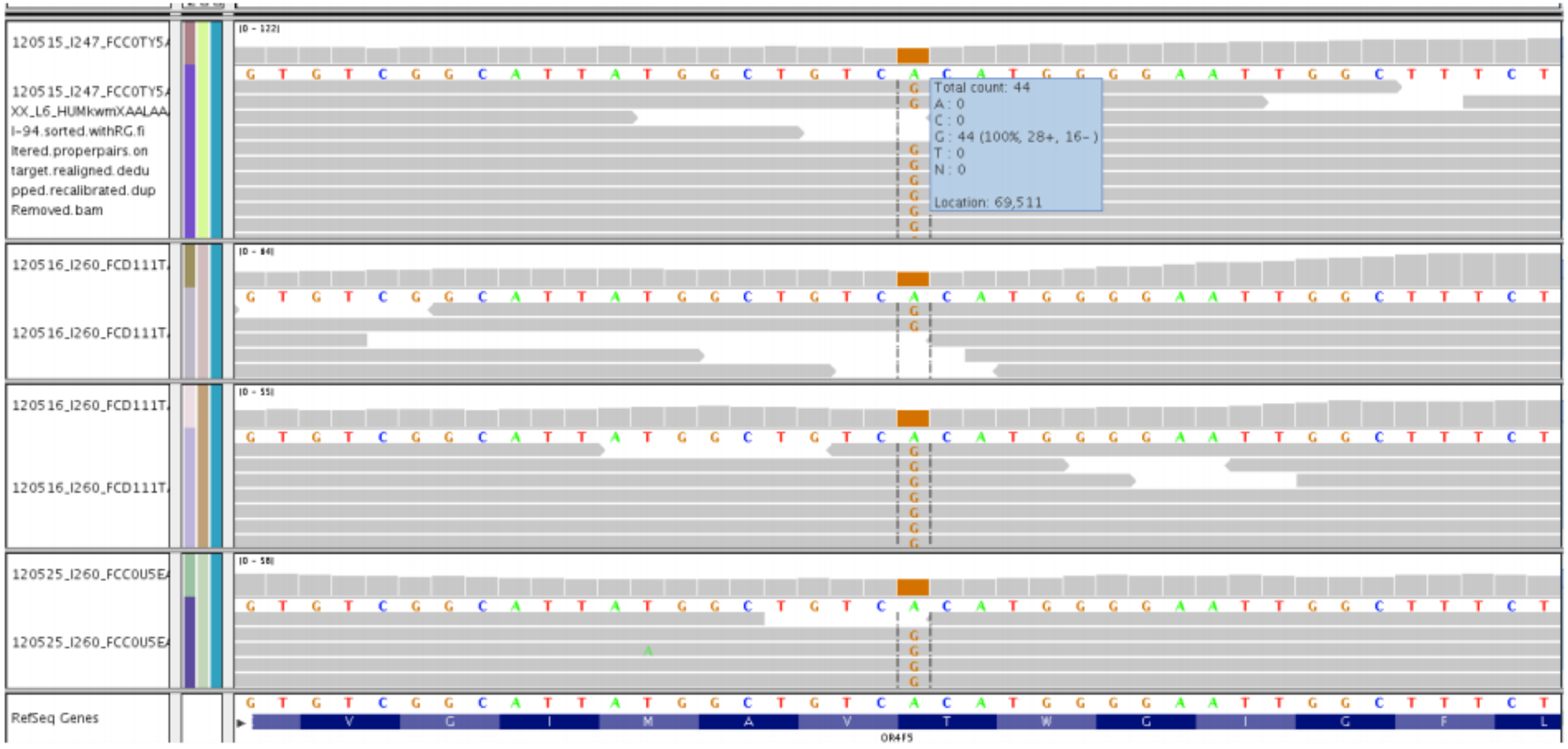

Base calling

Raw sequencing reads (*.**fastq**)

Aligned reads (*.**sam/bam**)

Genetic variation (*.**vcf**)

# In principle it is very simple …

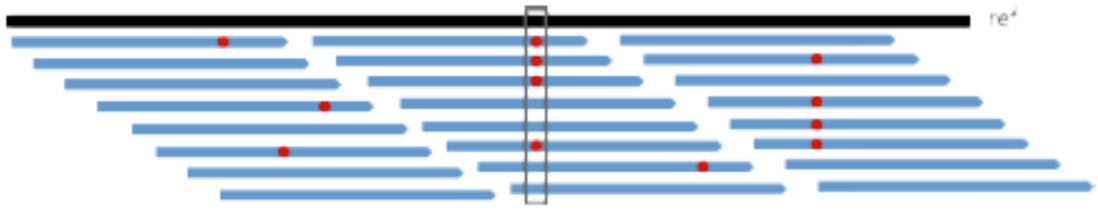



GENOMIKA

# But reality is slightly more complex…

# Objective

Assign a genotype to each position



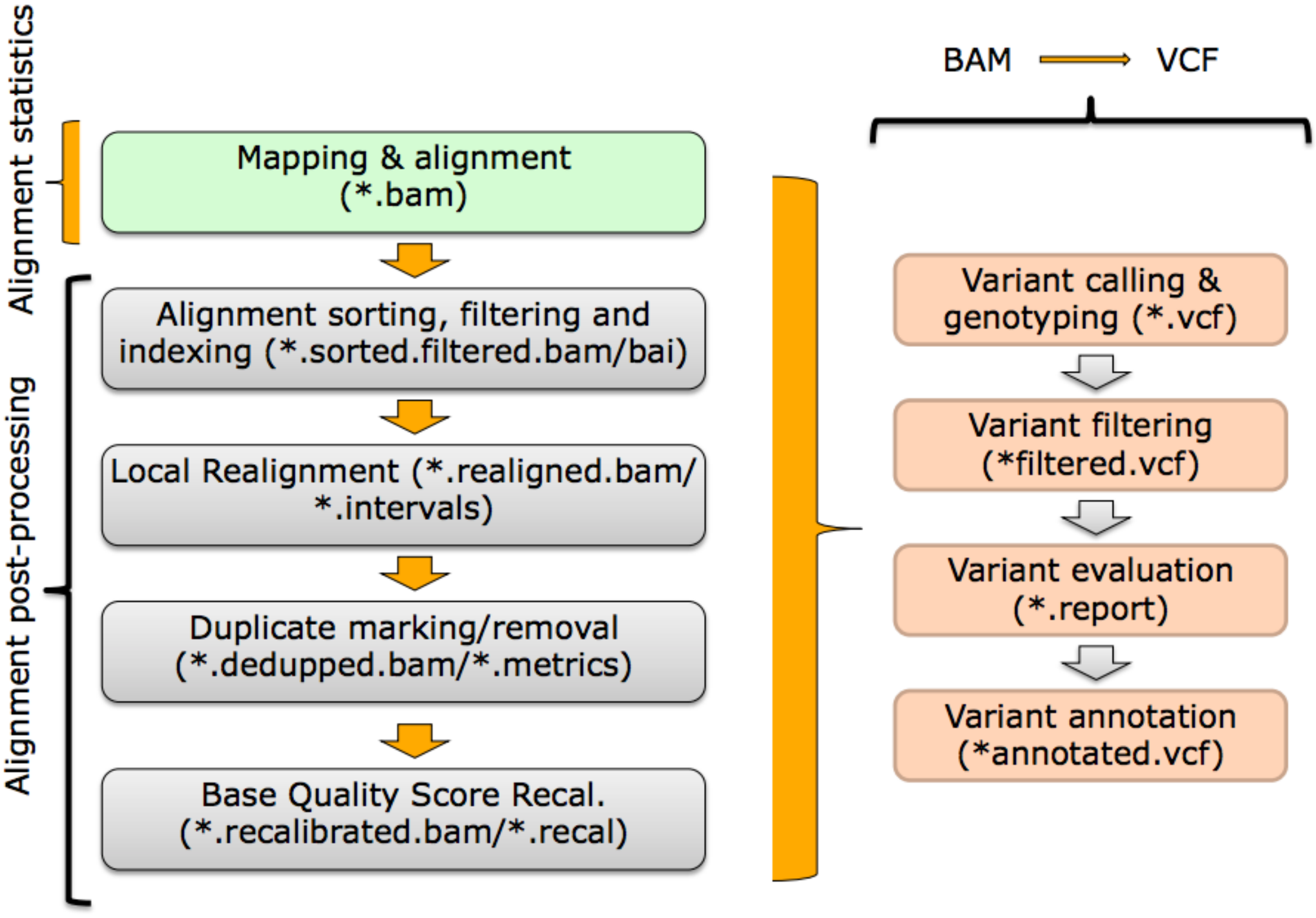
**Problems**

Some variation observed in BAM files is caused by mapping and sequencing artifacts:

- **PCR artifacts**:
  - Mismatches due to errors in early PCR rounds
  - PCR duplicates
- **Sequencing errors**: erroneous call, either for physical reasons or to properties of the sequenced DNA
- **Mapping errors**: often happens around repeats or other low-complexity regions

Separate **true variation** from machine artifacts

# Recommended Workflow




GE

Alignment statistics

Alignment post-processing

Mapping & alignment (*.bam)

Step 1

Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)

Local Realignment (*.realigned.bam/ *.intervals)

Duplicate marking/removal (*.dedupped.bam/*.metrics)

Base Quality Score Recal. (*.recalibrated.bam/*.recal)

BAM ⟶ VCF

Variant calling & genotyping (*.vcf)

Variant filtering (*filtered.vcf)

Variant evaluation (*.report)

Variant annotation (*annotated.vcf)

GENOMIKA

Alignment statistics

Alignment post-processing

Mapping & alignment (*.bam)

Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)

Step 2

Local Realignment (*.realigned.bam/ *.intervals)

Duplicate marking/removal (*.dedupped.bam/*.metrics)

Base Quality Score Recal. (*.recalibrated.bam/*.recal)

BAM → VCF

Variant calling & genotyping (*.vcf)

Variant filtering (*filtered.vcf)

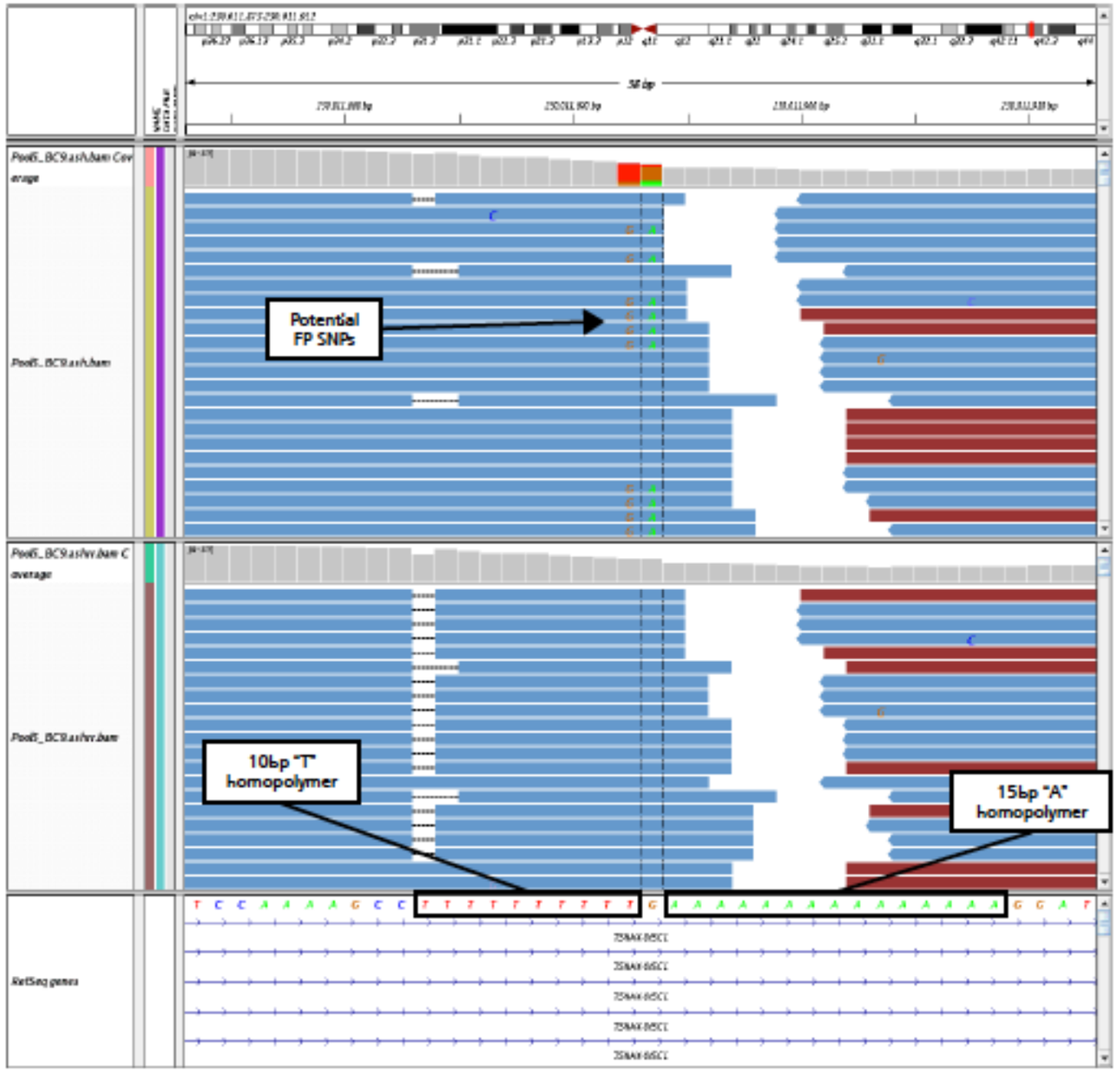
Variant evaluation (*.report)

Variant annotation (*annotated.vcf)

GENOMIKA

# Local realignment around INDELS

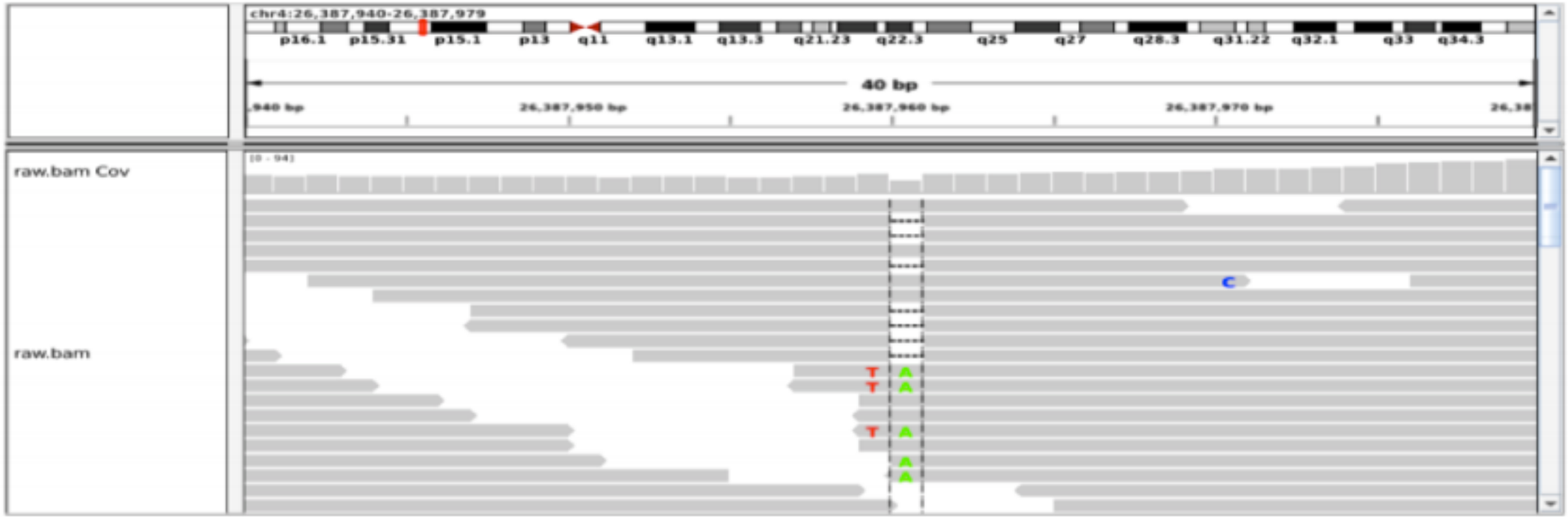
- Reads **near INDELS** are mapped with mismatches
- **Realignment** can identify the most consistent placement for these reads
    1. **Identify** problematic regions
    2. **Determine the optimal** consensus sequence
- **Minimizes mismatches** with the reference sequence
- **Refines** location of **INDELS**




GENOMIKA

# Local realignment around INDELS

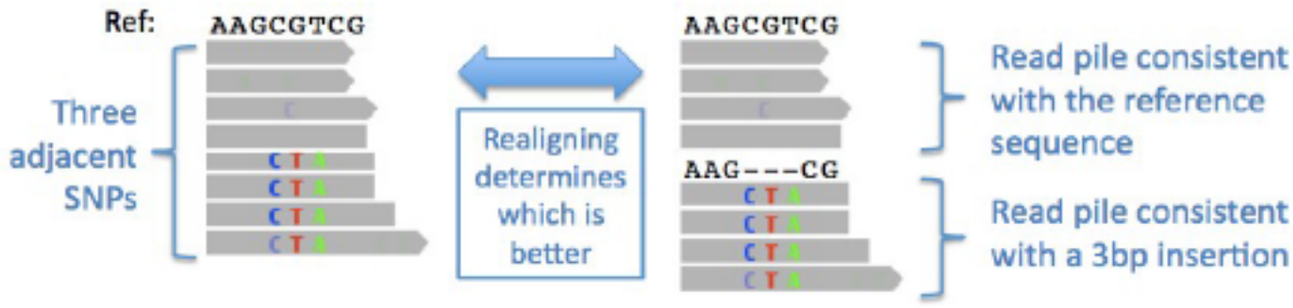
- Each read is independently aligned to the reference genome to cut down on the computational cost.
- But this means that many reads spanning indels are likely to be misaligned.
- We see this as regions containing indels as well as clusters of mismatching bases.




GENOMIKA

# Local realignment around INDELS

- Reads **near INDELS** are mapped with mismatches
- **Realignment** can identify the most consistent placement for these reads
  1. **Identify** problematic regions
  2. **Determine the optimal** consensus sequence
- **Minimizes mismatches** with the reference sequence
- **Refines** location of **INDELS**




GENOMIKA

# Local realignment[19,20]

1 – At least one read contains an indel.
2 – A cluster of mismatching bases exists.
3 – An already known indel segregates at the site.

- How does it work?
  - Step 1: Identify regions likely in need of realignment.
  - Step 2: Perform a multiple sequence realignment in this region, such that the number of mismatching bases is minimized across all reads.

HiSeq data, raw BWA alignments

HiSeq data, after MSA

**RAW**

**REALIGNED**

GENOMIKA

Alignment statistics

Mapping & alignment (*.bam)

Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)

Local Realignment (*.realigned.bam/ *.intervals)

Step 3

Duplicate marking/removal (*.dedupped.bam/*.metrics)

Base Quality Score Recal. (*.recalibrated.bam/*.recal)

Alignment post-processing

BAM ⟶ VCF

Variant calling & genotyping (*.vcf)

Variant filtering (*filtered.vcf)

Variant evaluation (*.report)

Variant annotation (*annotated.vcf)

# 1. Mark duplicates

- The same DNA molecule can be **sequenced several times during PCR**
- **Not informative**
- **Not** to be counted as **additional evidence** for or against a putative variant
- Can result in **false variant calls**

## Tools

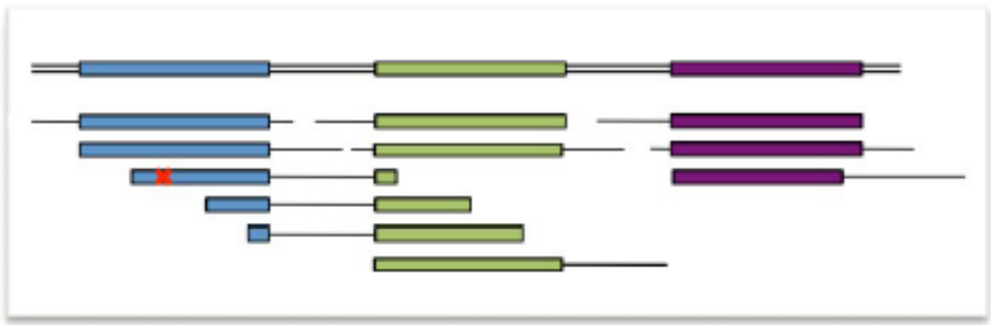
- Samtools: samtools rmdup or samtools rmdupse
- Picard: MarkDuplicates

# Marking Duplicates

✖ = sequencing error propagated in duplicates


**FP variant call (bad)**

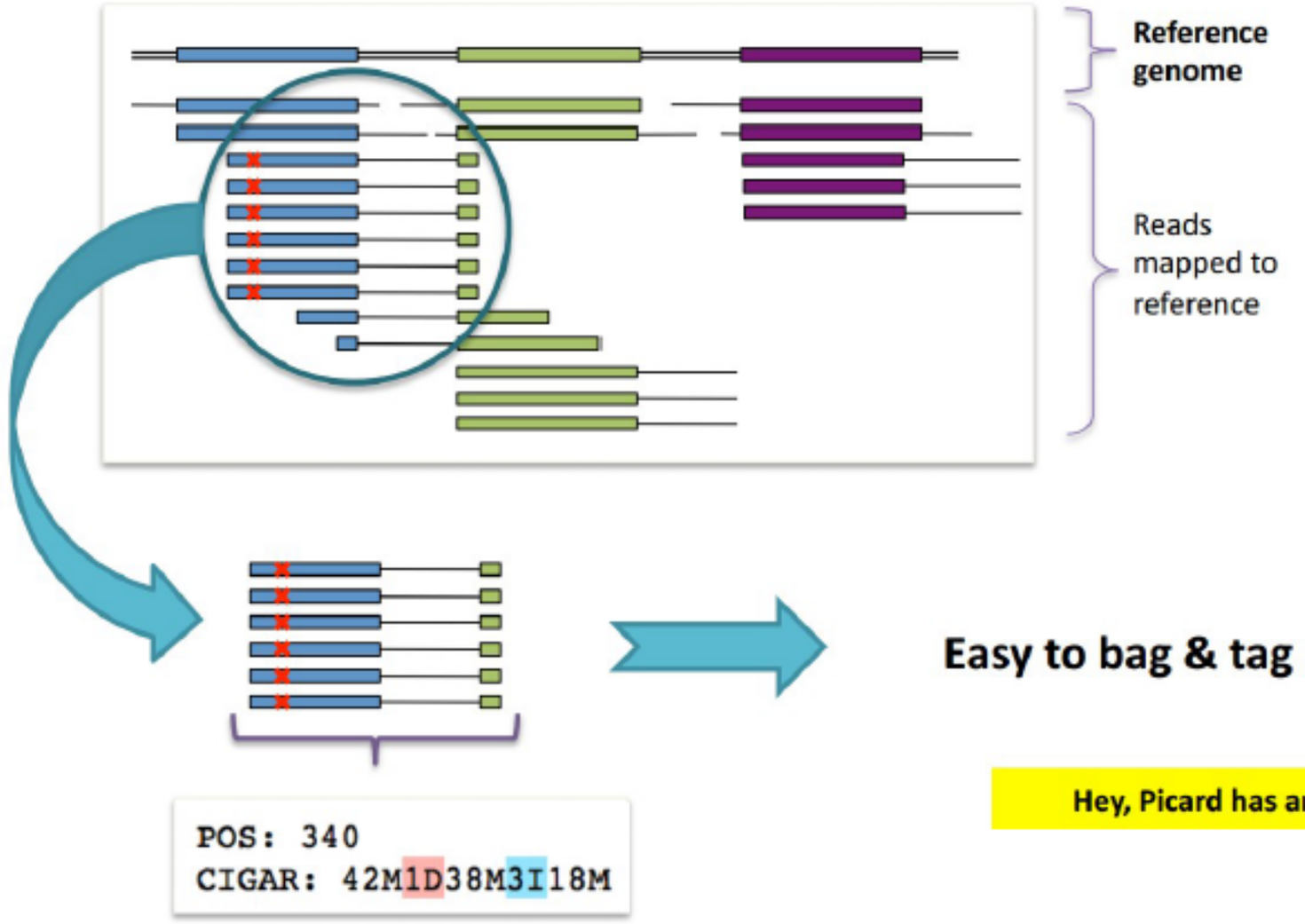
**After marking duplicates, the GATK will only see :**



**… and thus be more likely to make the right call**

# Marking Duplicates

Duplicates have the **same starting position** and the **same CIGAR** string




**Easy to bag & tag**

**Hey, Picard has an app for that!**

# Marking Duplicates

## Duplicate marking/ removal

Why do we have duplicates in our data?

- The PCR amplification step included in the majority of NGS library construction techniques can introduce duplicates in the data.

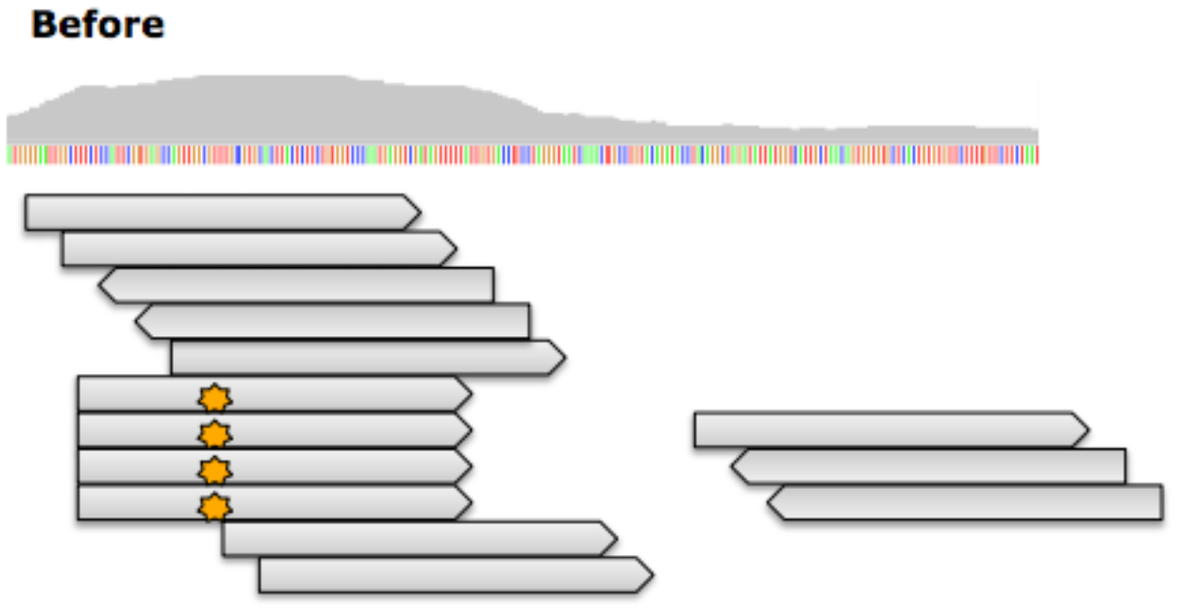
Why do we need to remove these?

- It will bias our variant calls.
- PCR errors are propagated and sampled many times giving rise to FP variant calls.

Basic concepts of duplicate marking algorithm:

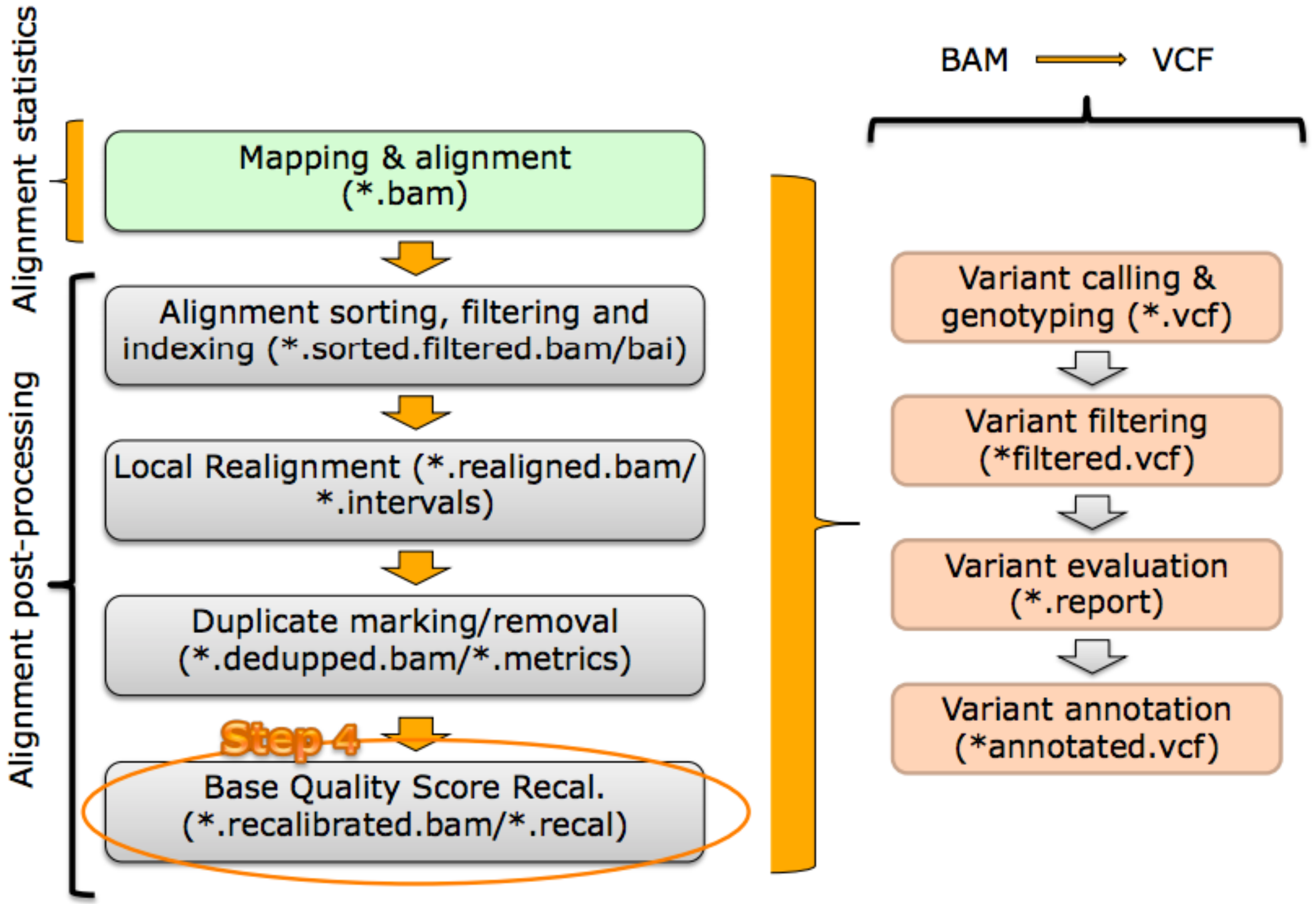
1. Identify genomic position and strand for 5'-most bases.
2. Mark reads that are duplicates of each other.
3. Within a group of duplicate reads the read with the highest sum of base quality scores is retained.

http://picard.sourceforge.net/




But it is not perfect…

- ❖ Does not account for sequencing errors.
- ❖ Does not account for natural duplicates.
- ❖ Does not account for duplicate reads with different mapping locations.


BAM
VCF
Alignment statistics
Alignment post-processing
Mapping & alignment (*.bam)
Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)
Local Realignment (*.realigned.bam/ *.intervals)
Duplicate marking/removal (*.dedupped.bam/*.metrics)
Step 4
Base Quality Score Recal. (*.recalibrated.bam/*.recal)
Variant calling & genotyping (*.vcf)
Variant filtering (*filtered.vcf)
Variant evaluation (*.report)
Variant annotation (*annotated.vcf)

# 3. Base quality score recalibration

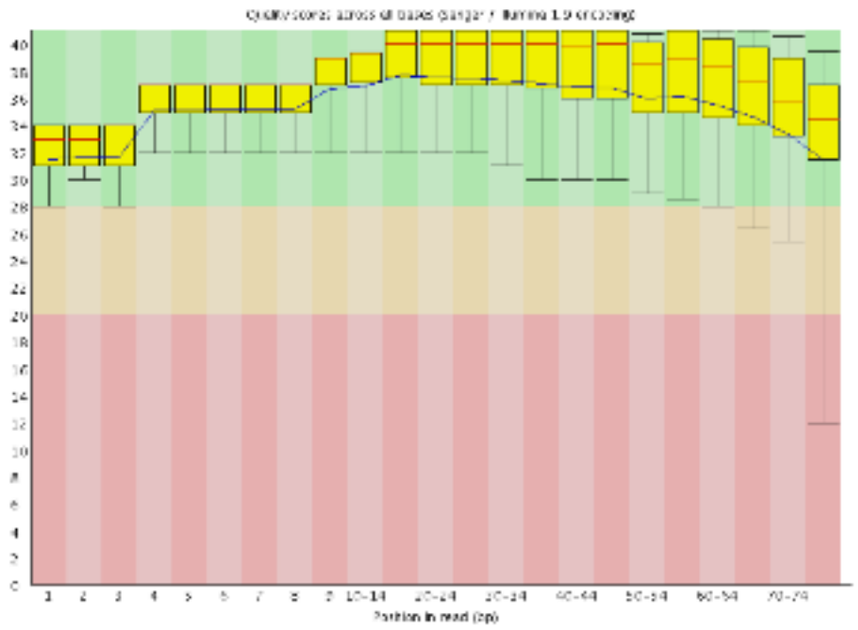
- **Calling algorithms rely** heavily on the **quality scores** assigned to the individual base calls in each sequence read
- Unfortunately, the scores produced by the machines are subject to various sources of **systematic error**, leading to over- or under-estimated base quality scores in the data

How?

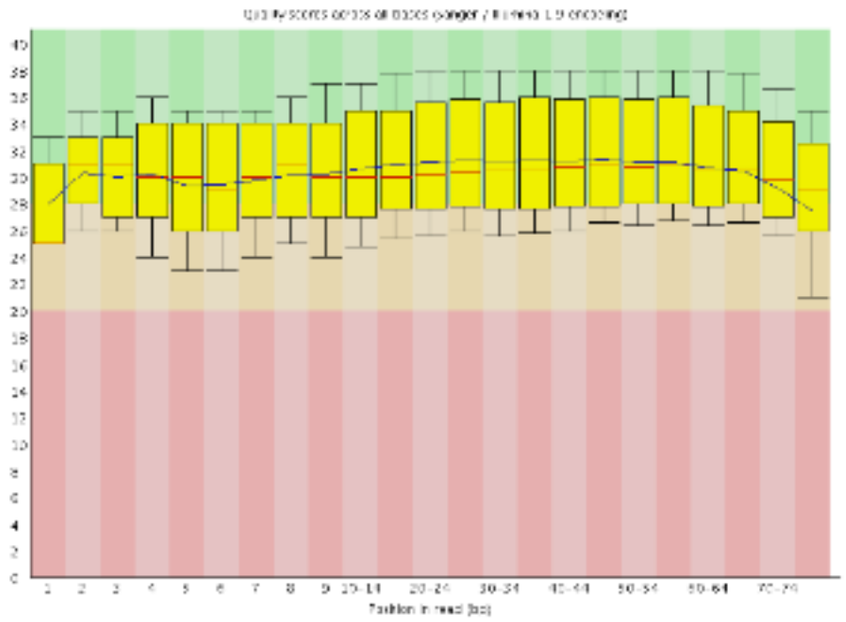
1. **Analyze covariation** among several features of a base:
   - Reported quality score
   - Position within the read
   - Preceding and current nucleotide
2. Use a set of **known variants** (i.e.: dbSNP) to model error properties of real polymorphism and determine the **probability that novel sites are real**
3. **Adjust** the quality scores of all reads in a BAM file

# 3. Base quality score recalibration

## Before



## After



## Phred Quality score:

$$Q_{\text{Phred}} = -10 \log_{10} P(\text{error}).$$

A score of 20 corresponds to 1% error rate in base calling

GENOMIKA

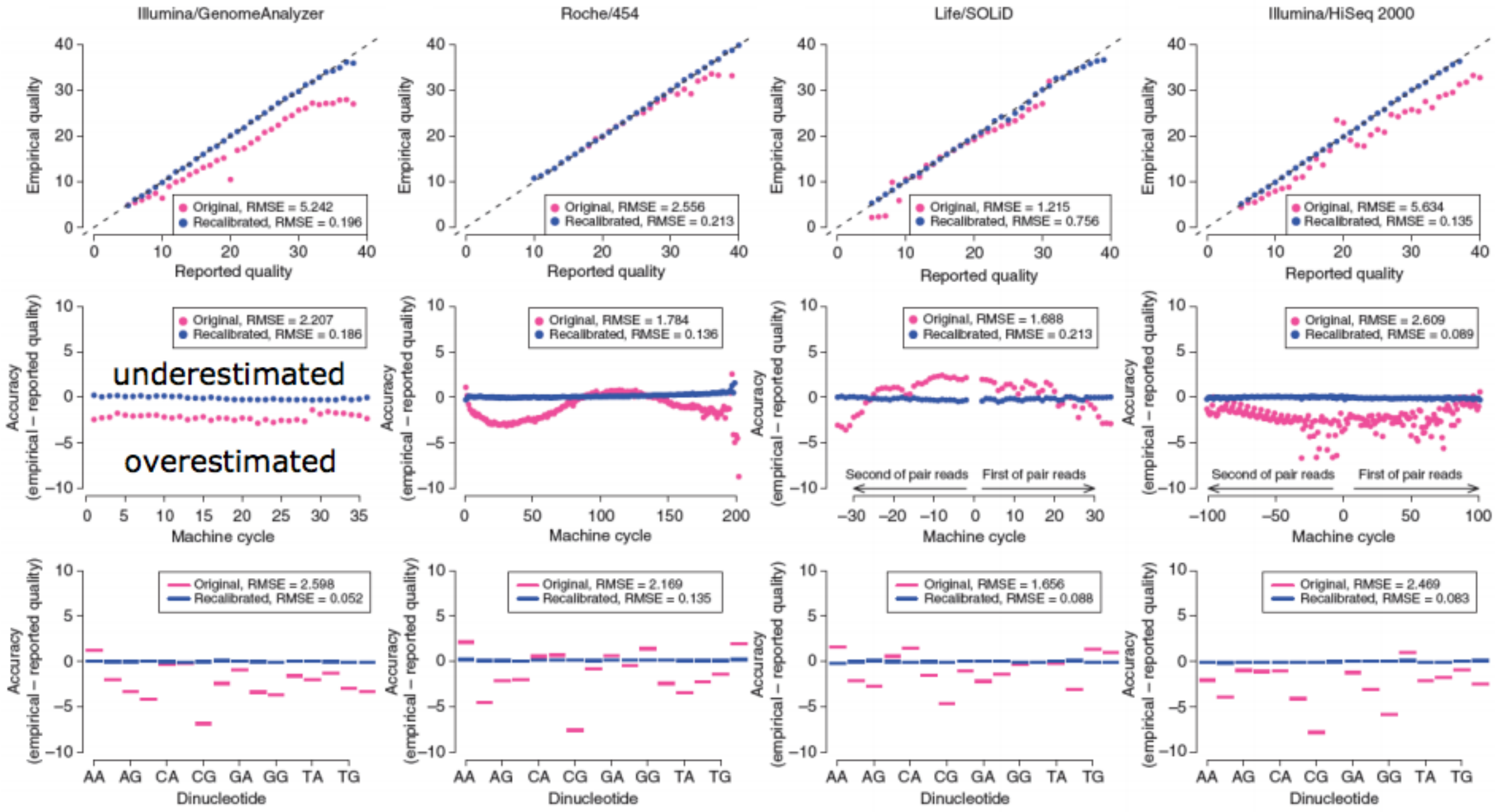
# Base Score Recalibration

- Base-calling algorithms produce per-base quality scores by using noise estimates from image analysis.
- The raw Phred-scaled quality scores produced by base-calling algorithms may not accurately reflect the true base-calling error rates.
- Obtaining well-calibrated base quality scores is important as SNP and genotype calling depends on both the base calls and the per-base quality scores.

```
@A80CRCABXX:2:1:11125:1940#CCCCCCCC/1
ATCATAAAGAAAATATATTTTGCAAATGAAGTCATTAATAATTTGAGAT
+
ccccc ddadcdd__dd\ddcddddddTdd`dd^ddddcddddcddaddd
@A80CRCABXX:2:1:12491:1939#TGTGGCTT/1
CTGACAGCATTTGTTCTGTTGCAGGATTACGCTCCCTTAGATCGGAAGA
+
fffffeffffffffffffffffffeeffffffffffffefffdffd
@A80CRCABXX:2:1:13158:1938#AAAAAAAA/1
ATGAGTGAAAAGCGTCTAATTCTCTATGCCATGCCTATTTTCTTTGTAA
+
dacdcacdeed`^c^dadd\``bbbc`aaac\^```b```cbc__[\bb
@A80CRCABXX:2:1:14354:1937#ACGGTTTT/1
CTCTCTTTCTCTGGCTGACTGCCTGTCTCTCTCTCTCTCTCTCTCTC
+
ffffffffffffffffeffeefceccedeeffffdfffffffffffdf
@A80CRCABXX:2:1:14546:1939#AACCGCTT/1
AGTAGTTTCAATACTCAAATTACCTCTACAGCCATGATGATAACAGCAG
+
fffffffffffefffffefffffffffffffefffffffffffffef
@A80CRCABXX:2:1:14819:1939#AACTAGAA/1
ATAGATGTTATTCAACTCCTTCAGGTTGTCTTGAACTGACTGACTCATG
+
ffffffffffffffffffffffffffffffffffffffffffffffff
```

GENOMIKA

# Base Score Recalibration




GENOMIKA

# Base Score Recalibration

- How it works!

1. Collect information regarding the following features of the bases:
   - Reported quality score
   - Position within the read (machine cycle)
   - Dinucleotide context (current base plus previous base)

2. **Count the number of times a site was a mismatch to the reference (excluding known polymorphic sites).**

3. Estimate new quality score as:

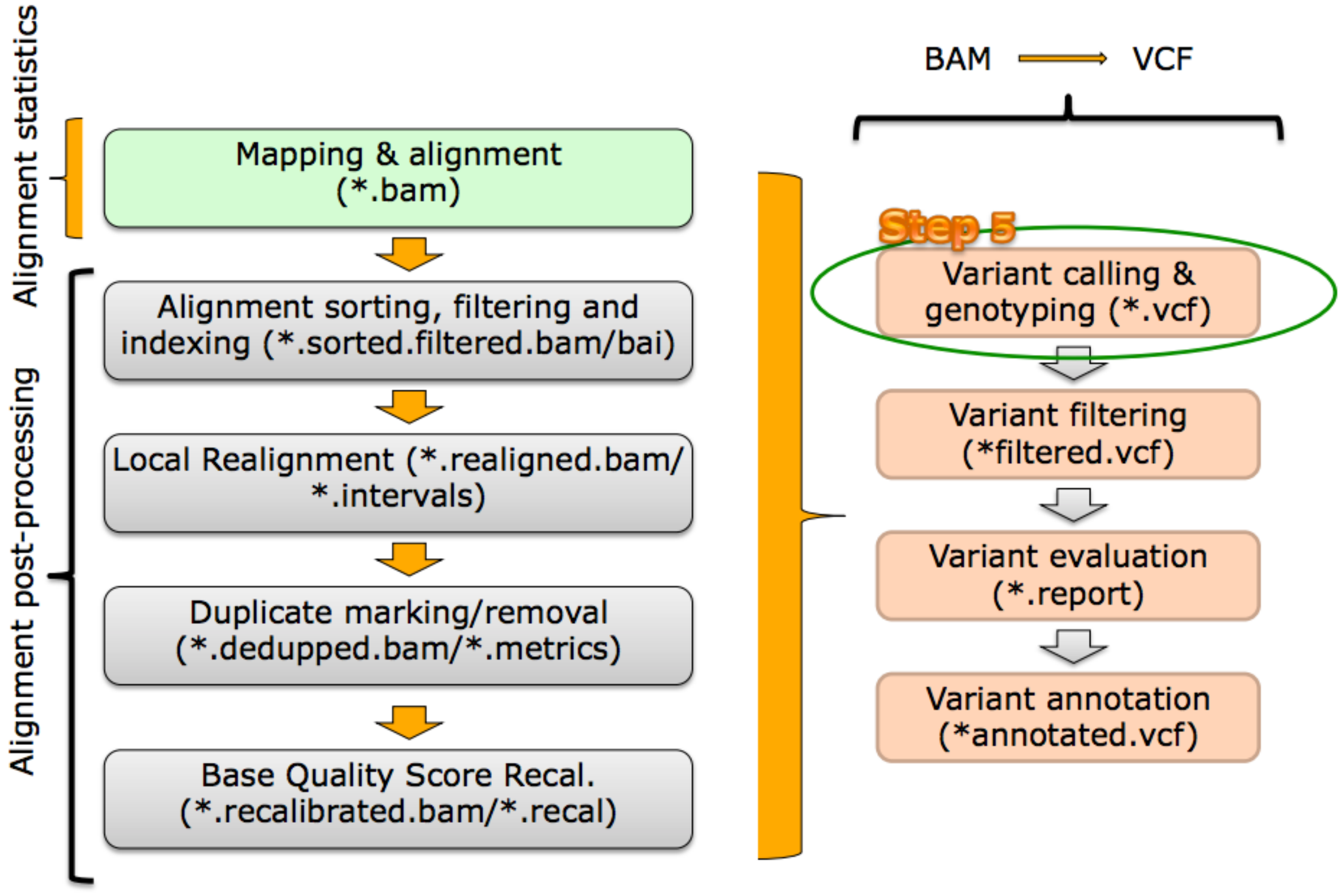
$$\text{Phred-scaled quality score} = \frac{\#\text{ of reference mismatches } +1}{\#\text{ of observed bases } +2}$$

Example:

We observed A [AA, pos. 35, Q20] a 1,000,000 times.

A in this context mismatches the reference a 1,000 times.

This gives us: Q value = -10log10((1,000+1)/(1,000,000+2)) ~ **Q30**


BAM
VCF
Alignment statistics
Alignment post-processing
Mapping & alignment (*.bam)
Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)
Local Realignment (*.realigned.bam/ *.intervals)
Duplicate marking/removal (*.dedupped.bam/*.metrics)
Base Quality Score Recal. (*.recalibrated.bam/*.recal)
Step 5
Variant calling & genotyping (*.vcf)
Variant filtering (*filtered.vcf)
Variant evaluation (*.report)
Variant annotation (*annotated.vcf)

# 4. Variant calling

## Variant discovery process

**Steps**

1. **Variant calling:** Identify the positions that differ from the reference
2. **Genotype calling**: calculate the genotypes for each sample at these sites

### Initial approach

**Independent** base assumption

Counting the number of times each allele is observed

### Evolved approach

**Bayesian inference** → Compute genotype likelihood

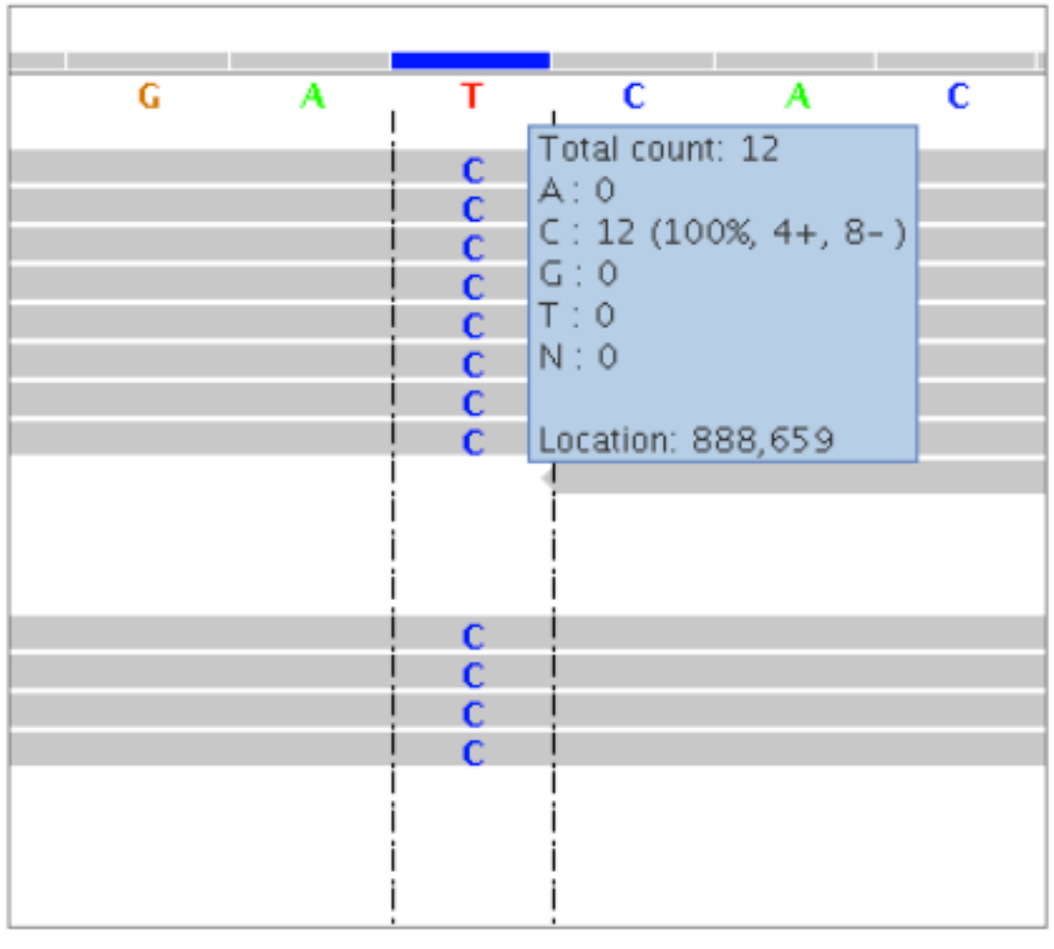
Advantages:

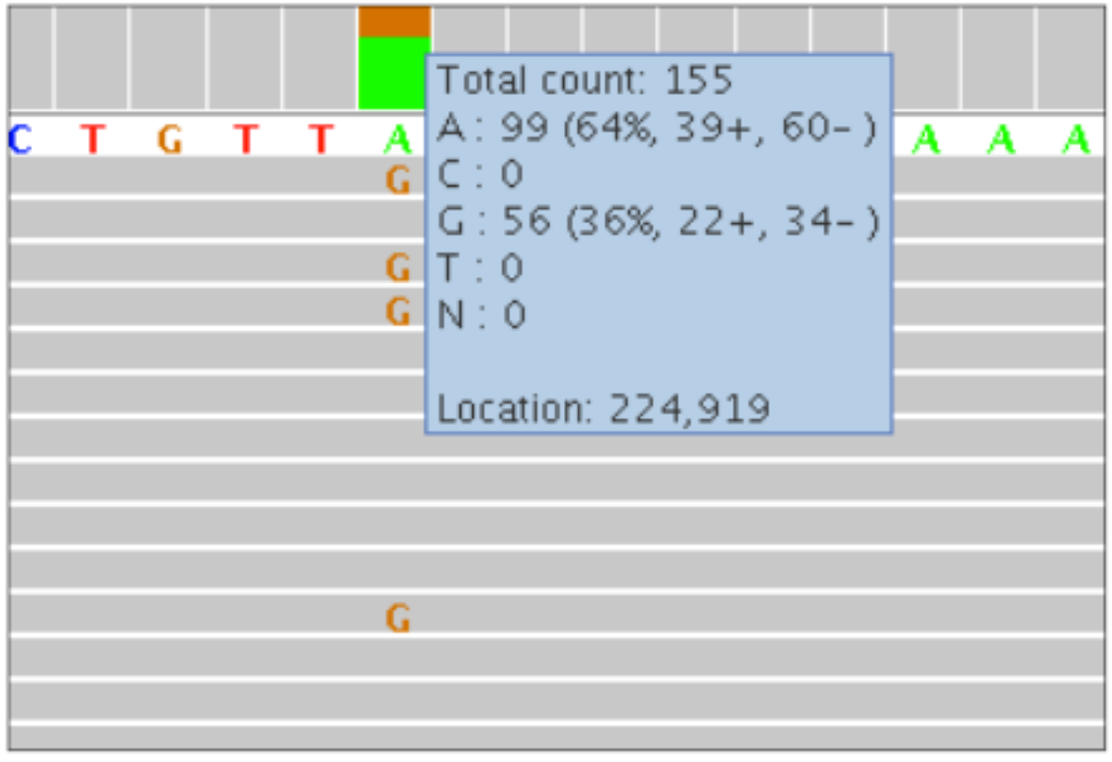
Provide statistical measure of **uncertainty**

Lead to **higher accuracy** of genotype calling

# Variant Calling and Genotyping

- Aims
  - Variant calling: Identify polymorphic sites => sites that differs from the reference.
  - Genotyping: Determine the genotype for a certain individual at such sites.




- Early methods
  - Works by simply counting the alleles at each site, and then identifying a variant by use of simple cutoff rules.

# Variant Calling and Genotyping

- Several Bayesian genotyping methods available:
- Use the information on base counts, base qualities, mapping quality
- Calculate genotype likelihoods

**Angsd**

**Samtools**[21]

**UnifiedGenotyper (GATK)**[9]

**FreeBayes**[11]

**HaplotypeCaller (GATK)**[10]

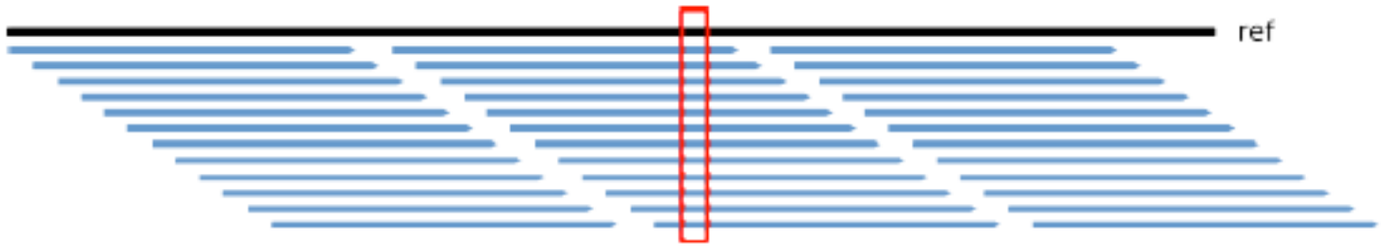
# 4. Variant calling

## Variant discovery process


**Reference = A**

# 4. Variant calling

## Variant discovery process




Reference = A

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  *N*=30,  *X*=0

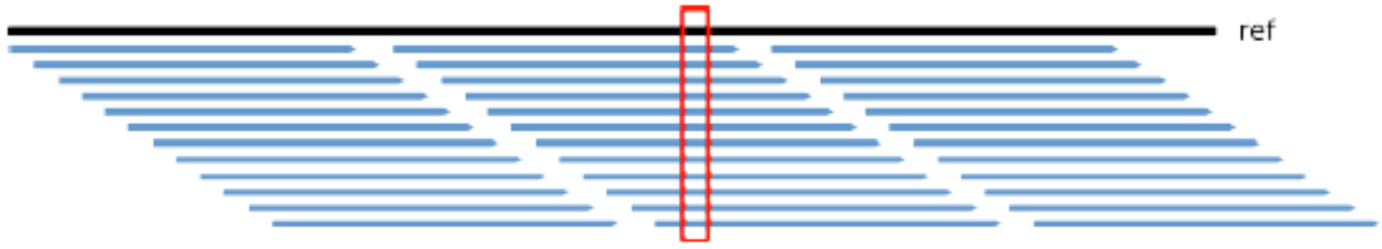
*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR  RV  VV

GENOMIKA

# 4. Variant calling

## Variant discovery process




Reference = A

| | |
|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, *X*=0 |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, *X*=30 |

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

# 4. Variant calling

## Variant discovery process


Reference = A

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  *N*=30,  *X*=0

GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG  *N*=30,  *X*=30

AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG  *N*=30,  *X*=15

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR  RV  VV

GENOMIKA

# 4. Variant calling

## Variant discovery process


**Reference =** A

| | |
|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, *X*=0 |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, *X*=30 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG | *N*=30, *X*=15 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGCT | *N*=30, *X*=12 |

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

GENOMIKA

# 4. Variant calling

## Variant discovery process


**Reference =** A

| | |
|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, *X*=0 |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, *X*=30 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG | *N*=30, *X*=15 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGCT | *N*=30, *X*=12 |
| AAAGGGCCTT | *N*=10, *X*=3 |

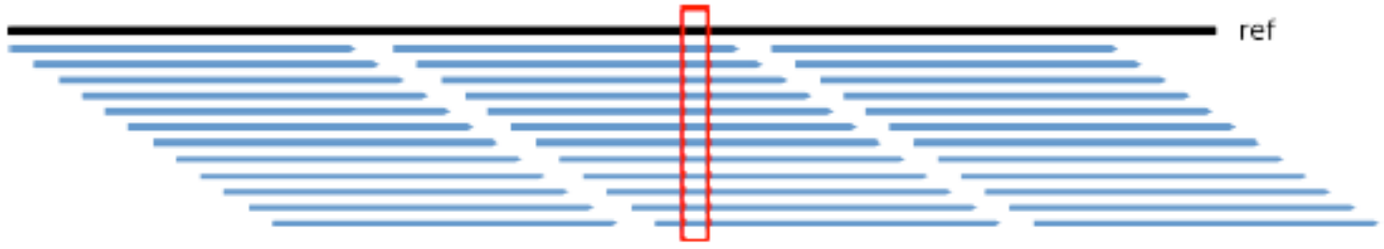
*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

GENOMIKA

# 4. Variant calling

## Variant discovery process




**Reference =** A

| | |
|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, *X*=0 |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, *X*=30 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGGG | *N*=30, *X*=15 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGCT | *N*=30, *X*=12 |
| AAAGGGCCTT | *N*=10, *X*=3 |

Cutoff for $X$ → value or proportion

- $c = 30\%$ $\quad X \leq c \rightarrow$ **RR**, $\quad X > c \rightarrow$ **RV**

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

GENOMIKA

# 4. Variant calling

## Variant discovery process


Reference = A

| | |
|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, *X*=0 |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, *X*=30 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGGG | *N*=30, *X*=15 |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGCT | *N*=30, *X*=12 |
| AAAGGGCCTT | *N*=10, *X*=3 |

Cutoff for $X$ → value or proportion

- $c = 30\%$ $\quad X \leq c \rightarrow$ **RR**, $\quad X > c \rightarrow$ **RV**
- $c_1 = 10\%, c_2 = 30\%$ $\quad X \leq c_1 \rightarrow$ **RR**; $\quad c_1 < X < c_2 \rightarrow$ **RV**; $\quad X \geq c_2 \rightarrow$ **RR**

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

GENOMIKA

# 4. Variant calling

## Variant discovery process


Reference = A

| | | |
|---|---|---|
| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA | *N*=30, | *X*=0 → **RR** |
| GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG | *N*=30, | *X*=30 → **VV** |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGGG | *N*=30, | *X*=15 → **RV** |
| AAAAAAAAAAAAAAAGGGGGGGGGGGGCT | *N*=30, | *X*=12 → **RV** |
| AAAGGGCCTT | *N*=10, | *X*=3 → **RV?** |

Cutoff for $X$ → value or proportion

- $c = 30\%$ $\quad X \leq c \rightarrow$ **RR**, $\quad X > c \rightarrow$ **RV**
- $c_1 = 10\%, c_2 = 30\%$ $\quad X \leq c_1 \rightarrow$ **RR**; $\quad c_1 < X < c_2 \rightarrow$ **RV**; $\quad X \geq c_2 \rightarrow$ **RR**

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

# 4. Variant calling

## Variant discovery process


**Bayesian approximation**

α = nucleotide-base error rate

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

$$P(G=RR, X|N, \alpha) = \text{P of all R calls being correct and all V calls being wrong}$$

$$P(G=VV, X|N, \alpha) = \text{P of all V calls being correct and all R calls being wrong}$$

$$P(G=RV, X|N, \alpha) = \text{P of all R and V calls being correct}$$

# 4. Variant calling

## Variant discovery process


**Bayesian approximation**

α = nucleotide-base error rate

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

$$P(G=RR, X|N,\alpha) = \binom{N}{X}\alpha^{X}(1-\alpha)^{N-X}$$

$$P(G=VV, X|N,\alpha) = \binom{N}{X}(1-\alpha)^{X}\alpha^{N-X}$$

$$P(G=RV, X|N,\alpha) = \binom{N}{X}\left(\frac{1}{2}\right)^{N}$$

# 4. Variant calling

## Variant discovery process


### Bayesian approximation

α = nucleotide-base error rate

$p_W$

$p_{VV}$

$p_{VR}$

Prior probabilities

*N* = nucleotides
*G* = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

$$P(G=RR, X|N, \alpha) = \binom{N}{X}\alpha^{X}(1-\alpha)^{N-X}(1-p_{VV}-p_{RV})$$

$$P(G=VV, X|N, \alpha) = \binom{N}{X}(1-\alpha)^{X}\alpha^{N-X}p_{VV}$$

$$P(G=RV, X|N, \alpha) = \binom{N}{X}\left(\frac{1}{2}\right)^{N}p_{RV}$$

- Probabilistic methods – a simple Bayesian genotyper[5]
- The posterior probability of each genotype given the pileup of sequence reads is given by Bayes theorem:

$$p(G|D) = \frac{p(G)p(D|G)}{p(D)}$$

$$p(D|G) = \prod_{b \in \text{pileup}} p(b|G)$$

$$p(b|G) = p(b|\{A_1, A_2\}) = \tfrac{1}{2}p(b|A_1) + \tfrac{1}{2}p(b|A_2)$$

$$p(b|A) = \begin{cases} \frac{e}{3} : b \neq A \\ 1 - e : b = A \end{cases},$$

p(G|D): Prob. of genotype given data (posterior probability)

p(G): Prior prob. based on allele freqs. from HW

p(D|G): Prob. of data given genotype (genotype likelihood)

p(D): Prob. of data

p(b|G): Prob. of genotype given observed base at a site

*e* is the reversed Phred scaled quality score of the base

## We see 3 reads: T (Q20), T (Q20), G (Q10)

Determine for each possible genotype

**1. Calculate Probability of each base given the allele p(b|A):**

For reads 1 and 2: T (Q20)
p(A|A) = 0.01/3
p(C|A) = 0.01/3
p(G|A) = 0.01/3
p(T|A) = 0.99

For read 3: G (Q10):
p(A|A) = 0.1/3
p(C|A) = 0.1/3
p(G|A) = 0.9
p(T|A) = 0.1/3

$$p(G|D) = \frac{p(G)p(D|G)}{p(D)}$$

$$p(D|G) = \prod_{b \in \text{pileup}} p(b|G)$$

$$p(b|G) = p(b|\{A_1, A_2\}) = \tfrac{1}{2}p(b|A_1) + \tfrac{1}{2}p(b|A_2)$$

$$p(b|A) = \begin{cases} \frac{e}{3} : b \neq A \\ 1 - e : b = A \end{cases},$$

**We see 3 reads: T (Q20), T (Q20), G (Q10)**

**2. Calculate prob. of each base given genotype p(b|G):**

$$p(G|D) = \frac{p(G)p(D|G)}{p(D)}$$

For reads 1 and 2:
TT: 0.99/2 + 0.99/2 = 0.99
TG: 0.99/2 + 0.01/(3*2) = 0.49
GG: 0.01/(3*2) + 0.01/(3*2) = 0.003

For read 3:
TT: 0.1/(3*2) + 0.1/(3*2) = 0.03
TG: 0.1/(3*2) + 0.9/2 = 0.47
GG: 0.9/2 + 0.9/2 = 0.9

$$p(D|G) = \prod_{b \in \text{pileup}} p(b|G)$$

$$p(b|G) = p(b|\{A_1, A_2\}) = \tfrac{1}{2}p(b|A_1) + \tfrac{1}{2}p(b|A_2)$$

$$p(b|A) = \begin{cases} \frac{e}{3} : b \neq A \\ 1 - e : b = A \end{cases},$$

**We see 3 reads: T (Q20), T (Q20), G (Q10)**

**3. Calculate prob. of genotype p(D|G):**

TT: 0.99 * 0.99 * 0.03 = 0.029

TG: 0.49 * 0.49 * 0.47 = 0.11

GG: 0.003 * 0.003 * 0.9 = 8e-06

$$p(G|D) = \frac{p(G)p(D|G)}{p(D)}$$

$$p(D|G) = \prod_{b \in \text{pileup}} p(b|G)$$

**4. Calculate posterior prob. p(G|D):**

p(D) = 0.029 + 0.11 + 8e-06 = 0.139

$$p(b|G) = p(b|\{A_1, A_2\}) = \frac{1}{2}p(b|A_1) + \frac{1}{2}p(b|A_2)$$

p(G): for simplicity we assume uniform prior

$$p(b|A) = \begin{cases} \frac{e}{3} : b \neq A \\ 1 - e : b = A \end{cases},$$

Posterior probability p(G|D):

TT: 0.029/p(D) = 0.21

TG: 0.11/p(D) = 0.79

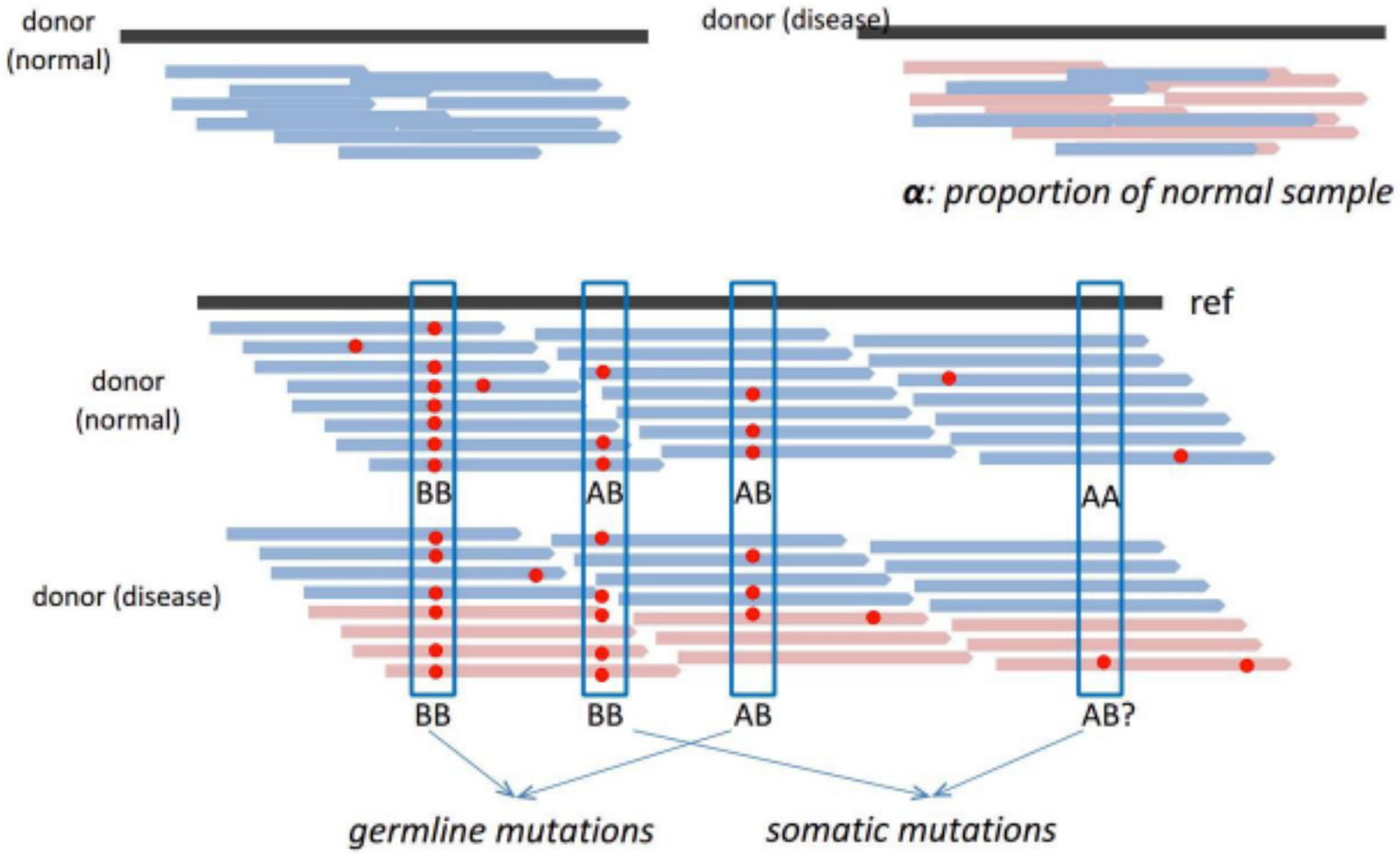
GG: 8e-06/p(D) = 5.8e-05

# Somatic calling

## Detecting somatic SNVs in cancer

**Challenges:**

- Somatic variants occur at low frequency in genome
- Most tumors are impure and heterogeneous




GENOMIKA

# VCF file format

- Specification defined by the 1000 genomes (current version **4.2**):
  http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41
- Commonly **compressed and indexed** with bgzip/tabix
- Single-sample or multi-sample VCF

```
##fileformat=VCFv4.1
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##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">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

GENOMIKA

# VCF file format

```
#CHROM  POS     ID        REF    ALT     QUAL   FILTER   INFO
20      14370   rs6054257 G      A       29     PASS     NS=3;DP=14;AF=0.5;DB;H2

FORMAT        NA00001          NA00002          NA00003
GT:GQ:DP:HQ   0|0:48:1:51,51   1|0:48:8:51,51   1/1:43:5:.,.
```

genotype genotype quality read depth haplotype qualities

- **CHROM**: chromosome
- **POS**: position
- **ID**: identifier
- **REF**: reference base(s)
- **ALT**: non-reference allele(s)
- **QUAL**: quality score of the calls (phed scale)
- **FILTER**: “PASS” or a filtering tag
- **INFO**: additional information
- **FORMAT**: describes the information given by sample

# Software

| Software | Available from | Calling method | Prerequisites | Comments | Refs |
|---|---|---|---|---|---|
| SOAP2 | http://soap.genomics.org.cn/index.html | Single-sample | High-quality variant database (for example, dbSNP) | Package for NGS data analysis, which includes a single individual genotype caller (SOAPsnp) | 15 |
| realSFS | http://128.32.118.212/thorfinn/realSFS/ | Single-sample | Aligned reads | Software for SNP and genotype calling using single individuals and allele frequencies. Site frequency spectrum (SFS) estimation | - |
| Samtools | http://samtools.sourceforge.net/ | Multi-sample | Aligned reads | Package for manipulation of NGS alignments, which includes a computation of genotype likelihoods (samtools) and SNP and genotype calling (bcftools) | 53 |
| GATK | http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit | Multi-sample | Aligned reads | Package for aligned NGS data analysis, which includes a SNP and genotype caller (Unifed Genotyper), SNP filtering (Variant Filtration) and SNP quality recalibration (Variant Recalibrator) | 32,33 |
| Beagle | http://faculty.washington.edu/browning/beagle/beagle.html | Multi-sample LD | Candidate SNPs, genotype likelihoods | Software for imputation, phasing and association that includes a mode for genotype calling | 42 |
| IMPUTE2 | http://mathgen.stats.ox.ac.uk/impute/impute_v2.html | Multi-sample LD | Candidate SNPs, genotype likelihoods | Software for imputation and phasing, including a mode for genotype calling. Requires fine-scale linkage map | 44 |
| QCall | ftp://ftp.sanger.ac.uk/pub/rd/QCALL | Multi-sample LD | 'Feasible' genealogies at a dense set of loci, genotype likelihoods | Software for SNP and genotype calling, including a method for generating candidate SNPs without LD information (NLDA) and a method for incorporating LD information (LDA). The 'feasible' genealogies can be generated using Margarita (http://www.sanger.ac.uk/resources/software/margarita) | 54 |
| MaCH | http://genome.sph.umich.edu/wiki/Thunder | Multi-sample LD | Genotype likelihoods | Software for SNP and genotype calling, including a method (GPT_Freq) for generating candidate SNPs without LD information and a method (thunder_glf_freq) for incorporating LD information | - |

A more complete list is available from http://seqanswers.com/wiki/Software/list. LD, linkage disequilibrium; NGS, next-generation sequencing.

# GATK (Genome Analysis ToolKit)

http://www.broadinstitute.org/gatk/

- Probabilistic method: **Bayesian estimation** of the most likely genotype
- Calculates many **parameters** for each position of the genome
- INDEL realignment
- Base quality recalibration
- SNP and INDEL calling
- **Multi-sample** calling
- Uses standard input and output files
- Used in **many NGS projects**, including the 1000 Genomes Project, The Cancer Genome Atlas, etc.

# GATK prerequisites

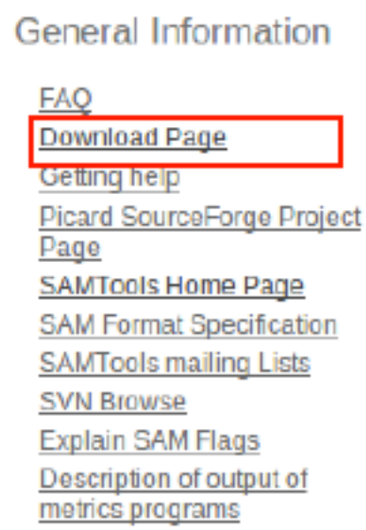
- **Requires Java** (http://www.oracle.com/technetwork/java/javase/downloads/index.html)
  - Check your java version

```
java -version
```

GATK ≥ 2.6 → Requires Java version 1.7

- **Picard**
  - Website: http://picard.sourceforge.net/
  - Go to Download page and select

Download picard-tools-1.114.zip (48.0 MB)




  - Testing:

```
java -jar AddOrReplaceReadGroups.jar -h
```

  - Usage

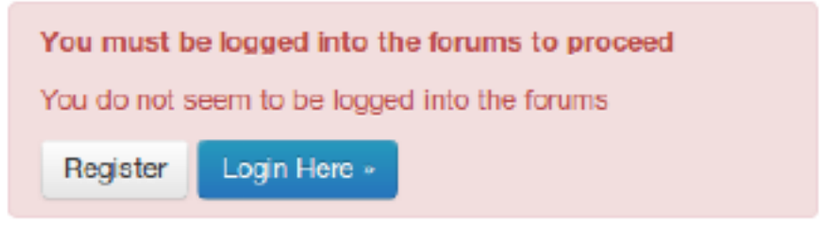
```
java -jar <ToolName> [options]
```

# GATK installation

- **GATK 3.2-2 download**

  http://www.broadinstitute.org/gatk/

  - We need to register before download
  - Go to Downloads and click 
  - Accept the license agreement
  - Extract the file in the applications folder




- Check if GATK is working

Show GATK help

```
java -jar GenomeAnalysisTK.jar -h
```

- Usage

```
java -jar GenomeAnalysisTK.jar -T <ToolName> [arguments]
```

# Variant calling tools

- UnifiedGenotyper

  Call SNPs and indels separately by considering each variant locus independently

  - Accepts any ploidy
  - Pooled calling
  - High sample numbers

- HaplotypeCaller

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

  - More accurate, especially for indels
  - Will eventually replace UG

# Haplotype Caller

- In discovery mode (default), outputs variants called above confidence thresholds

```
java –jar GenomeAnalysisTK.jar –T HaplotypeCaller \
        –R human.fasta \
        –I input.bam \
        –o output.vcf \
        –stand_call_conf 30 \
        –stand_emit_conf 10 \
        –minPruning 3
```

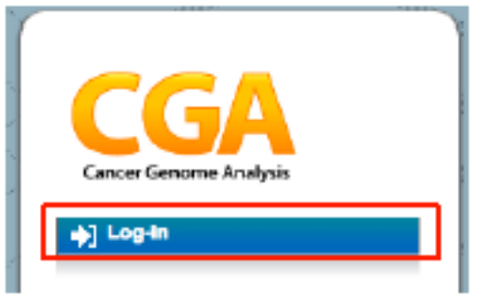
# MuTect installation

- **MuTect download**

  http://www.broadinstitute.org/cancer/cga/mutect

  - Click ***Log-in*** and go to the ***Create new account*** tab
  - Fill the form
  - Go to ***How do I get mutect*** and accept the license agreement
  - Download the latest version

    muTect-1.1.4-bin.zip
  - Extract the file in the applications folder





- **Check if MuTect is working**

```
java -jar muTect-1.1.4.jar -h
```

- **Usage**

```
java -jar muTect-1.1.4.jar --analysis_type MuTect [arguments]
```

Alignment statistics

Alignment post-processing

Mapping & alignment (*.bam)

Alignment sorting, filtering and indexing (*.sorted.filtered.bam/bai)

Local Realignment (*.realigned.bam/ *.intervals)

Duplicate marking/removal (*.dedupped.bam/*.metrics)

Base Quality Score Recal. (*.recalibrated.bam/*.recal)

BAM ⟶ VCF

Variant calling (*.vcf)

Step 6

Variant filtering (*filtered.vcf)

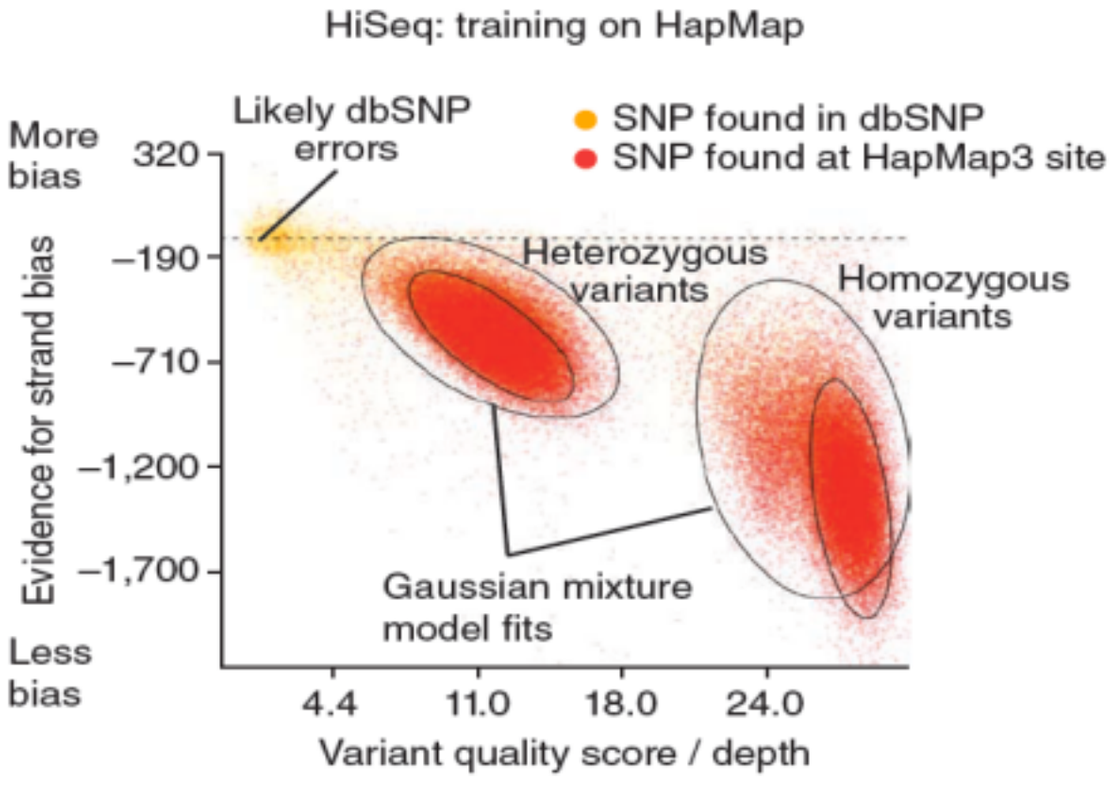
Variant evaluation (*.report)

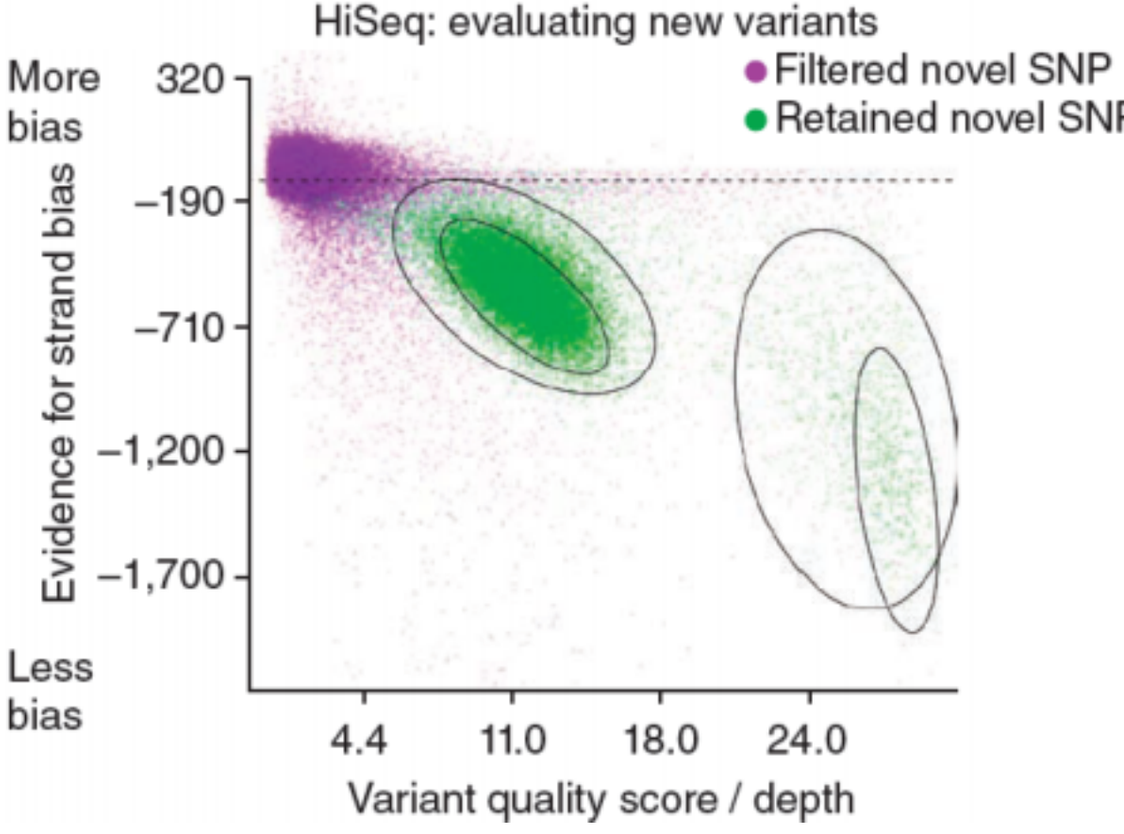
Variant annotation (*annotated.vcf)

GENOMIKA

# Variant Filtration

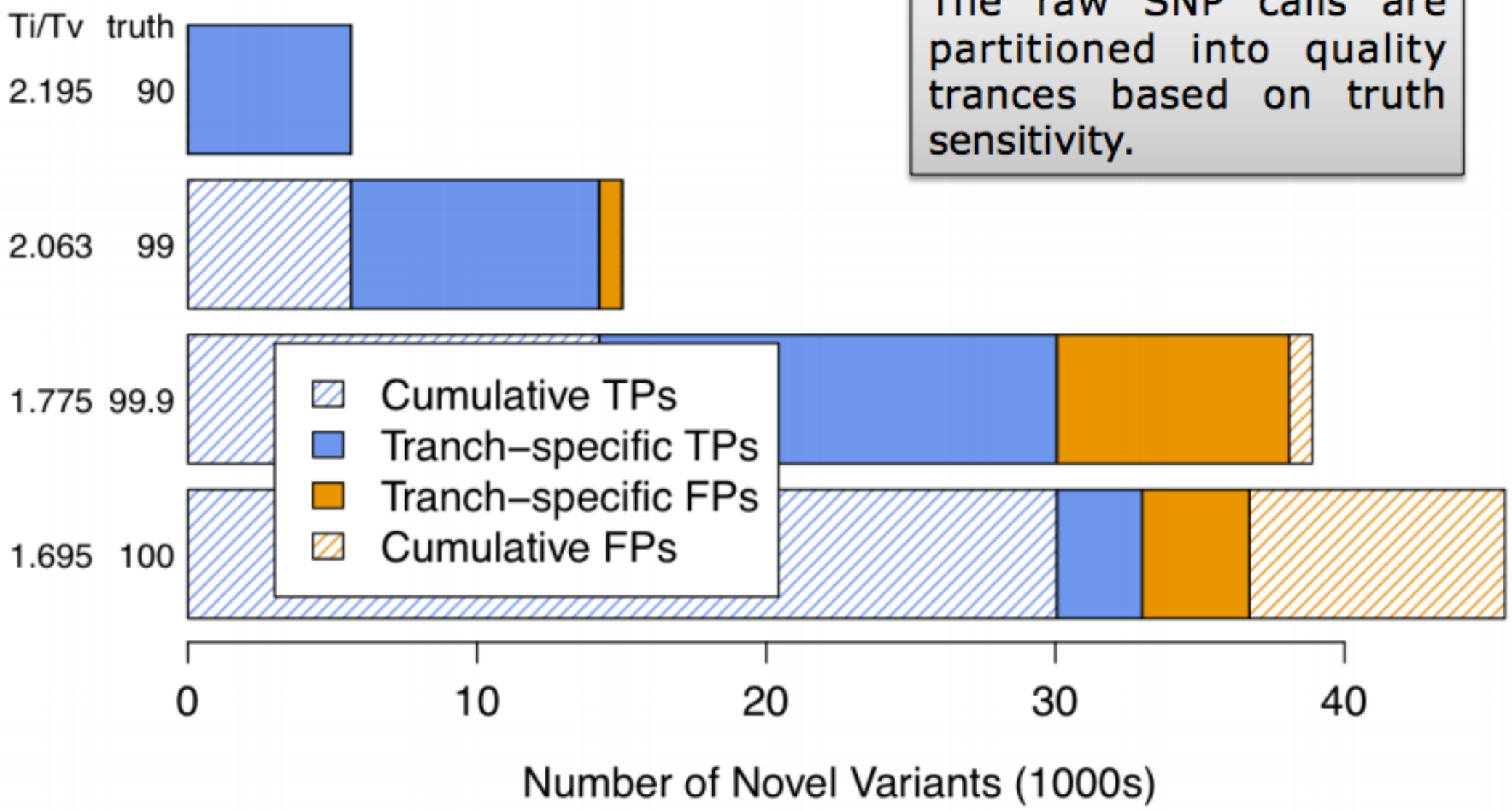
- **Variant quality score recalibration (GATK)**
  - **How do we remove false positive calls?**
  - **Use known polymorphic sites to estimate what a real variant and a false variant "looks like"**
  - Learn how does the known sites (=truth set) look like in our data
  - Examples can be: quality/depth, MQ, BaseReadPos, strand bias
  - Evaluate on all our data, filter sites that look different!



# Variant Filtration

The raw SNP calls are partitioned into quality trances based on truth sensitivity.

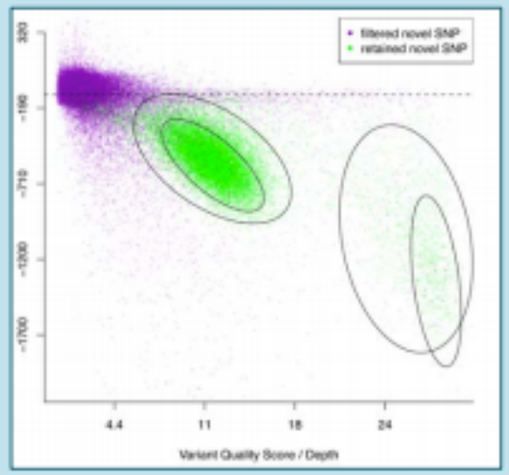



Truth: Accept all sites until X% of the truth sites have been found

# Variant Filtration

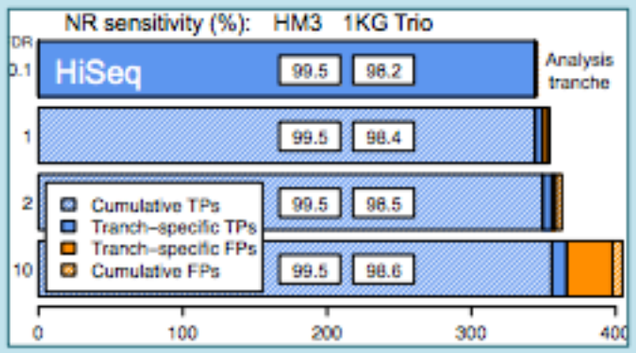
- Build the Gaussian mixture model

→ **VariantRecalibrator**




- Apply filters and write new annotated VCF

→ **ApplyRecalibration**




GENOMIKA

# Variant Filtration

- Hard filtering[16] (when VQSR is not possible*)
  - Variant quality score /depth
  - Mapping quality
  - Strand bias (the variant being seen only on the forward strand or only on the reverse strand)
  - Depth

- Some recommendations[17]

For SNPs:

- `QD < 2.0`
- `MQ < 40.0`
- `FS > 60.0`
- `HaplotypeScore > 13.0`
- `MQRankSum < -12.5`
- `ReadPosRankSum < -8.0`

For indels:

- `QD < 2.0`
- `ReadPosRankSum < -20.0`
- `InbreedingCoeff < -0.8`
- `FS > 200.0`

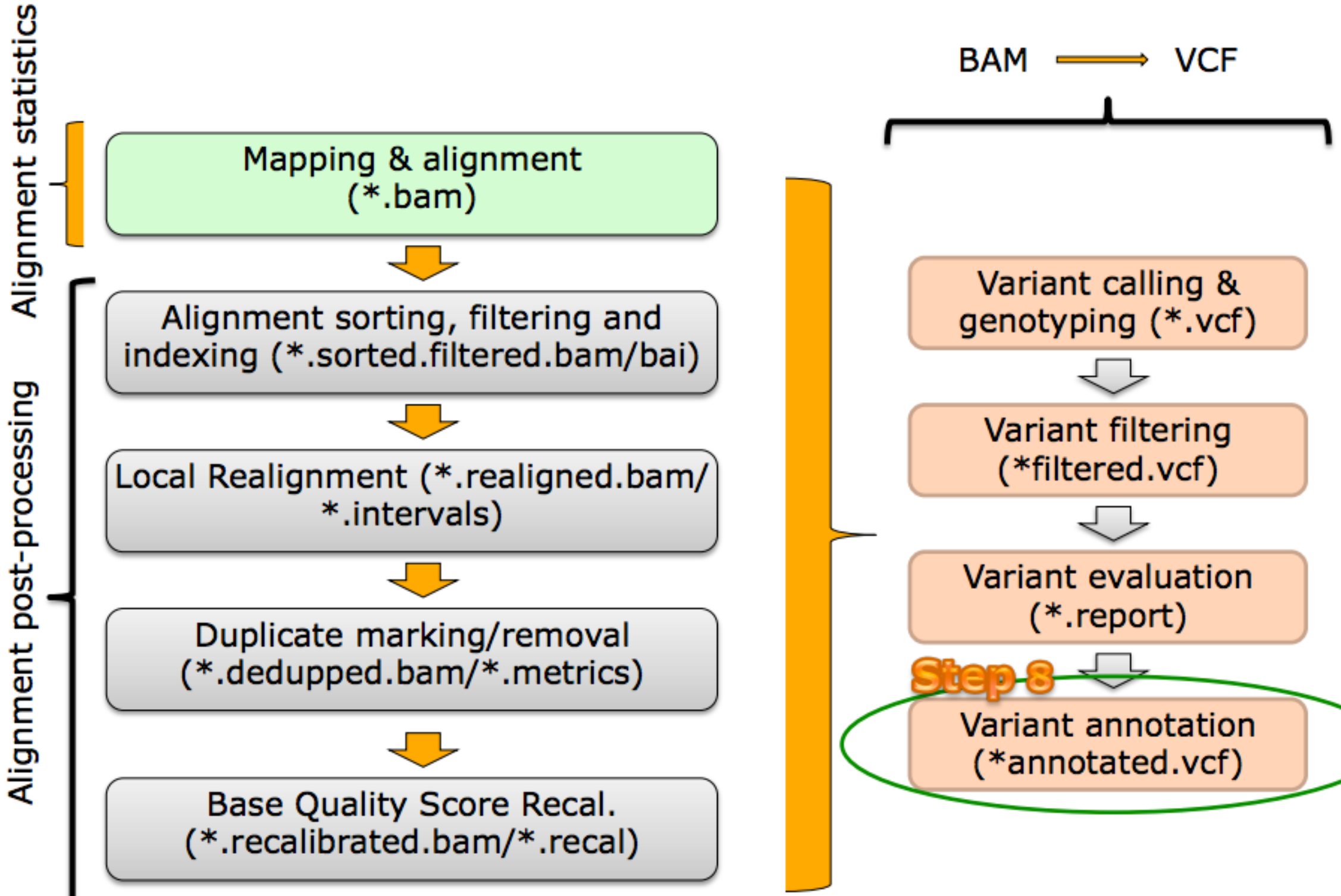
↘

10 or more samples!

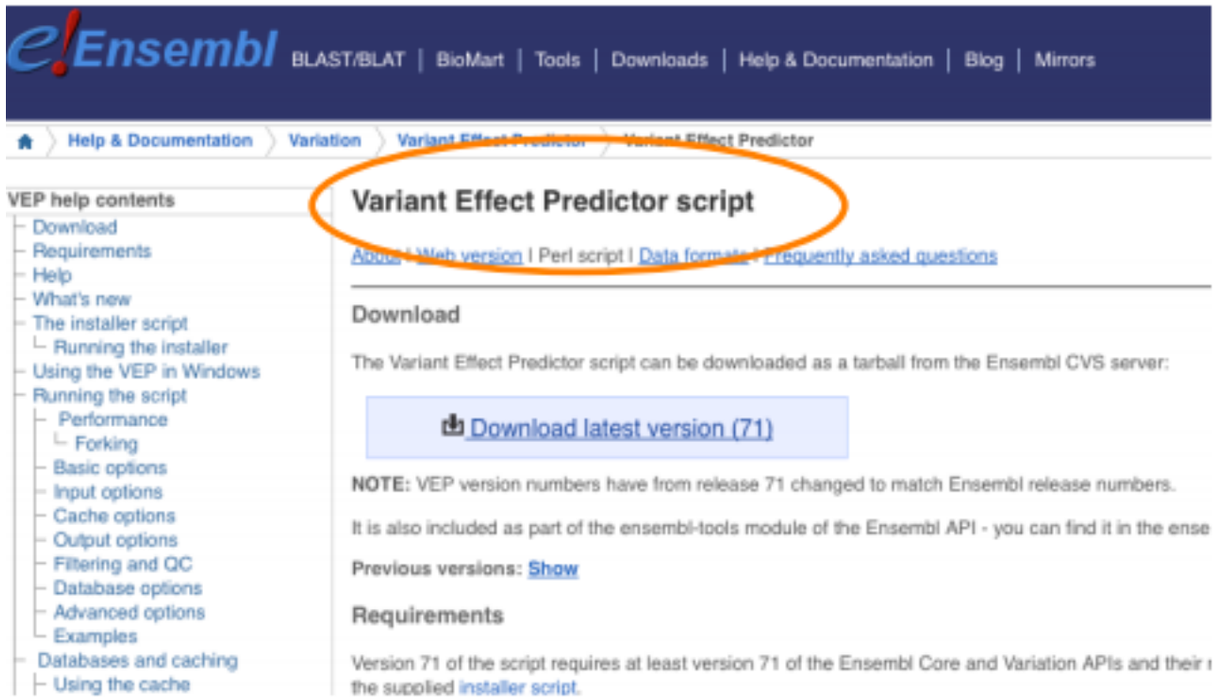
*number of samples < 30 or if your doing targeted resequencing of a small region, non-model organism.

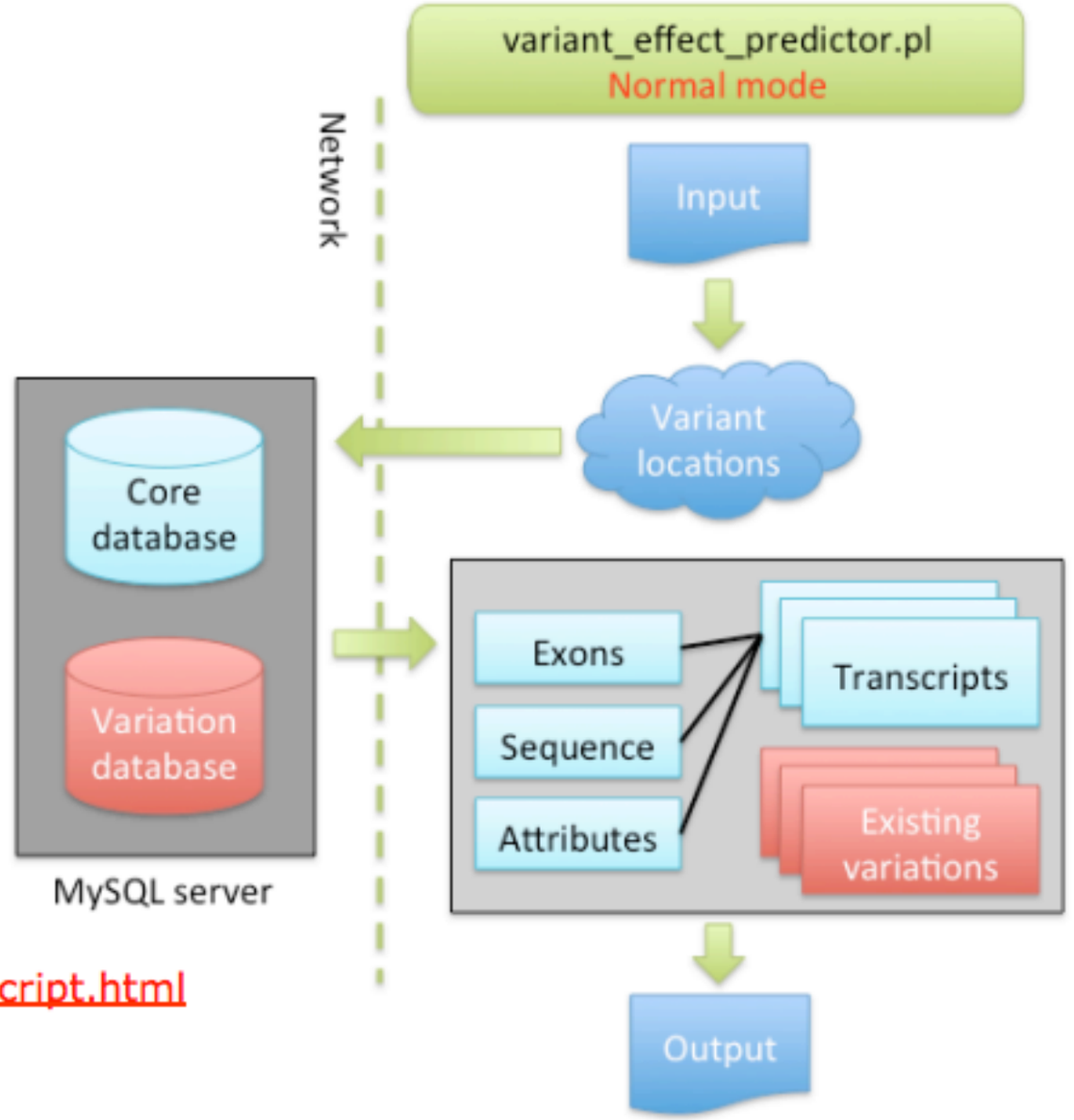
GENOMIKA

# Variant Filtration

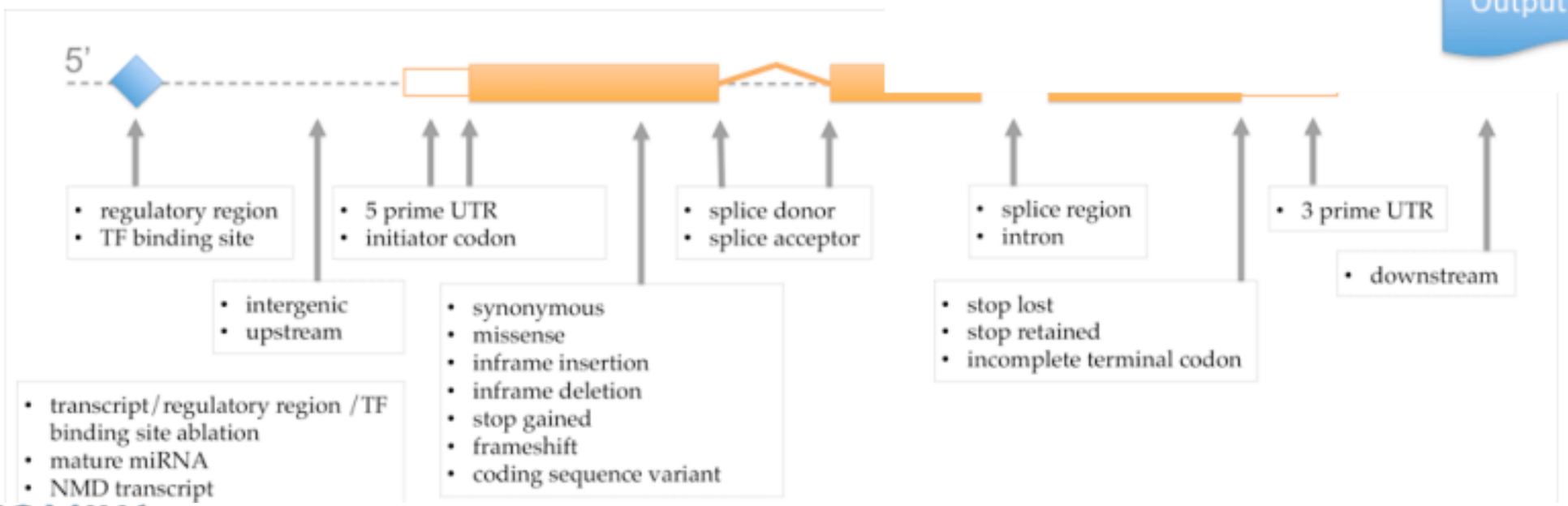


# Variant annotation[8]




Link: http://www.ensembl.org/info/docs/variation/vep/vep_script.html




GENOMIKA

GENOMIKA

# Processamento Pós-alinhamento & Chamada de Variantes

Marcel Caraciolo
marcel@genomika.com.br