Population Genomics

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

MER 2030
2020-2026

With the support of the Erasmus+ Programme of the European Union

Luis J. Chueca
Postdoctoral researcher
@LuisjaChueca

Basque Centre for Climate Change (BC3)
luisjavier.chueca@bc3research.org

Sofia Marcos
Postdoctoral researcher

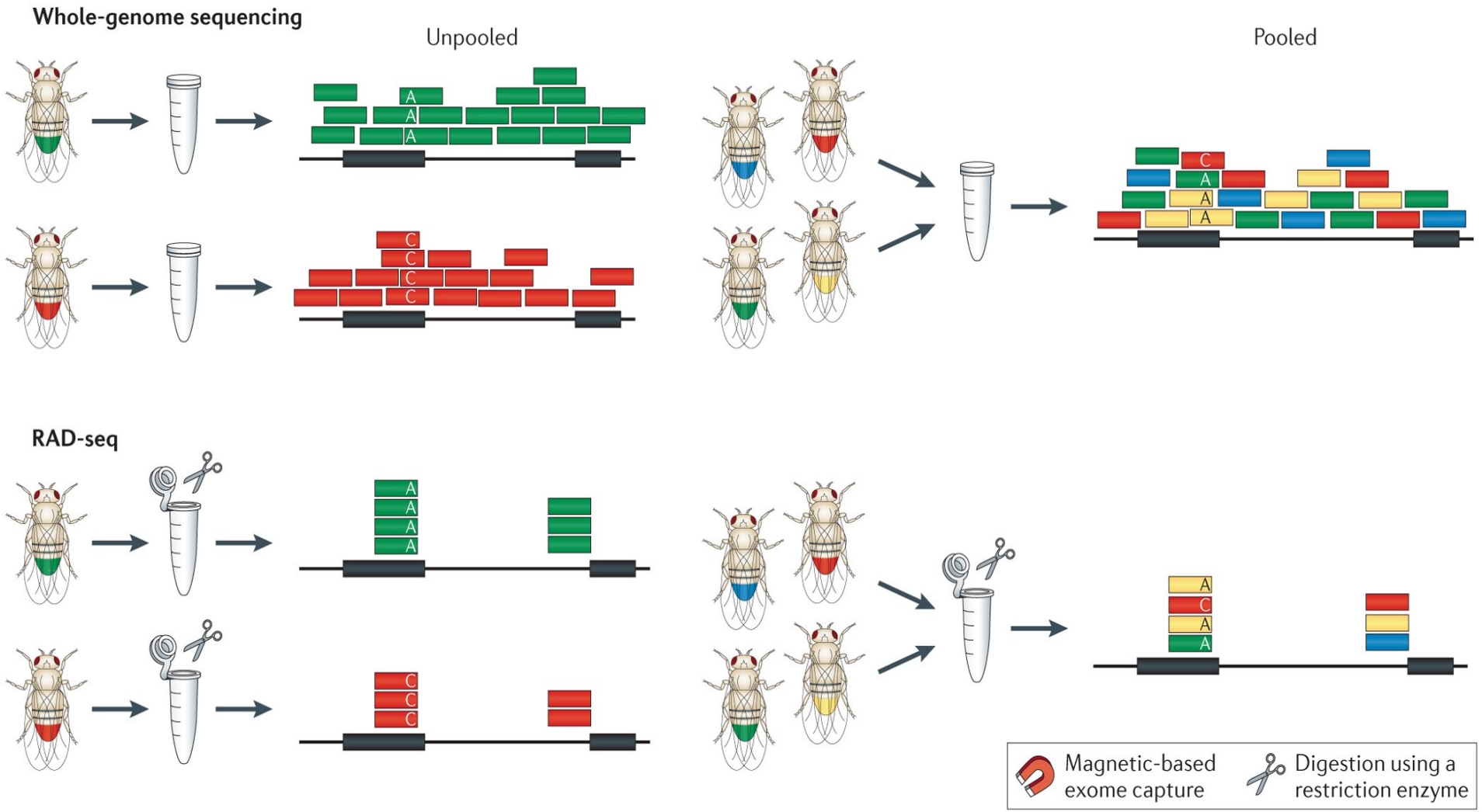
University of the Basque Country (UPV/EHU)
sofia.marcos@ehu.eus

Population genomics

It is a large scale comparison of DNA sequence of populations.

It studies genome-wide effects to improve our understanding of microevolution so that we may learn the phylogenetic history and demography of a population.

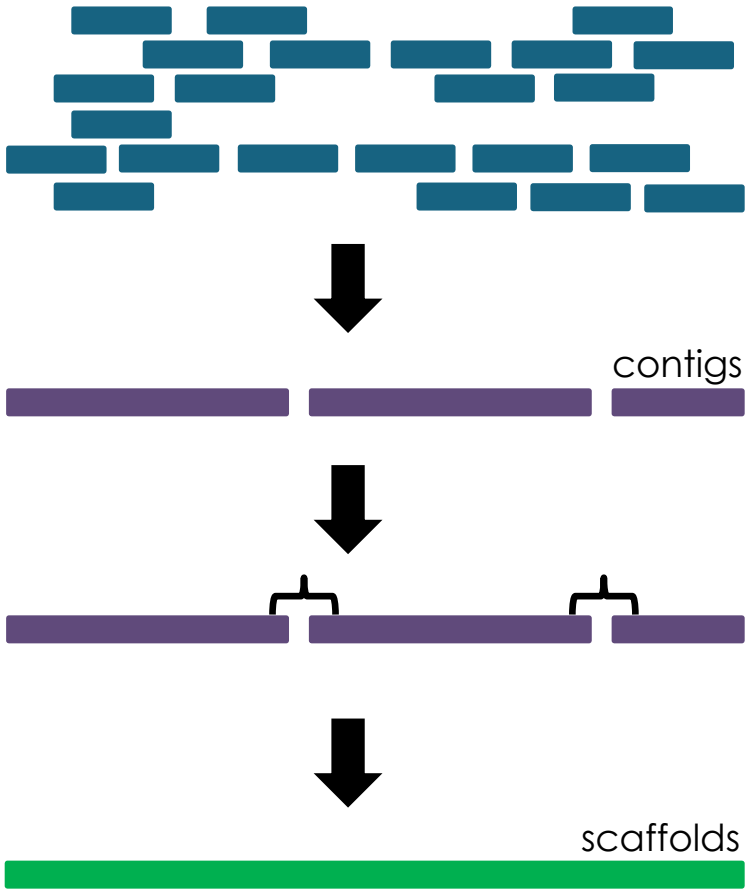
They are based mainly in high-throughput sequencing like RADSeq and Whole-genome sequencing



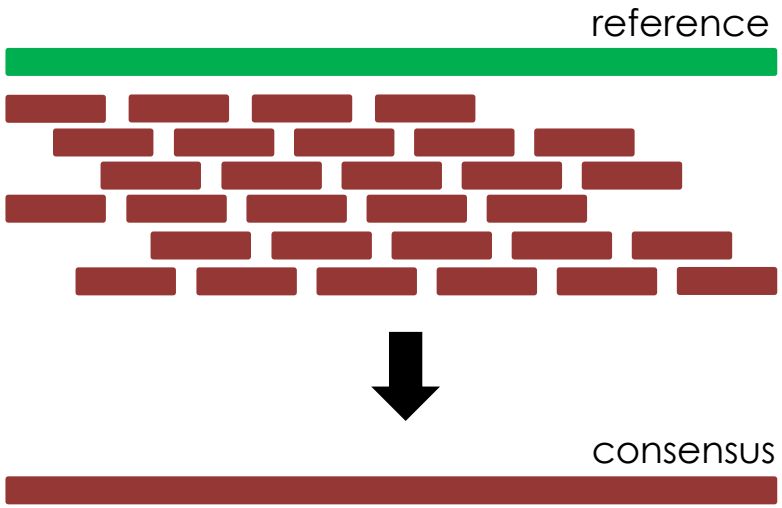
Modify from Schlötter et *al.*, 2014. *Nature Review Genetics*

Planning a WGS project:

De novo genome assembly vs re-sequencing



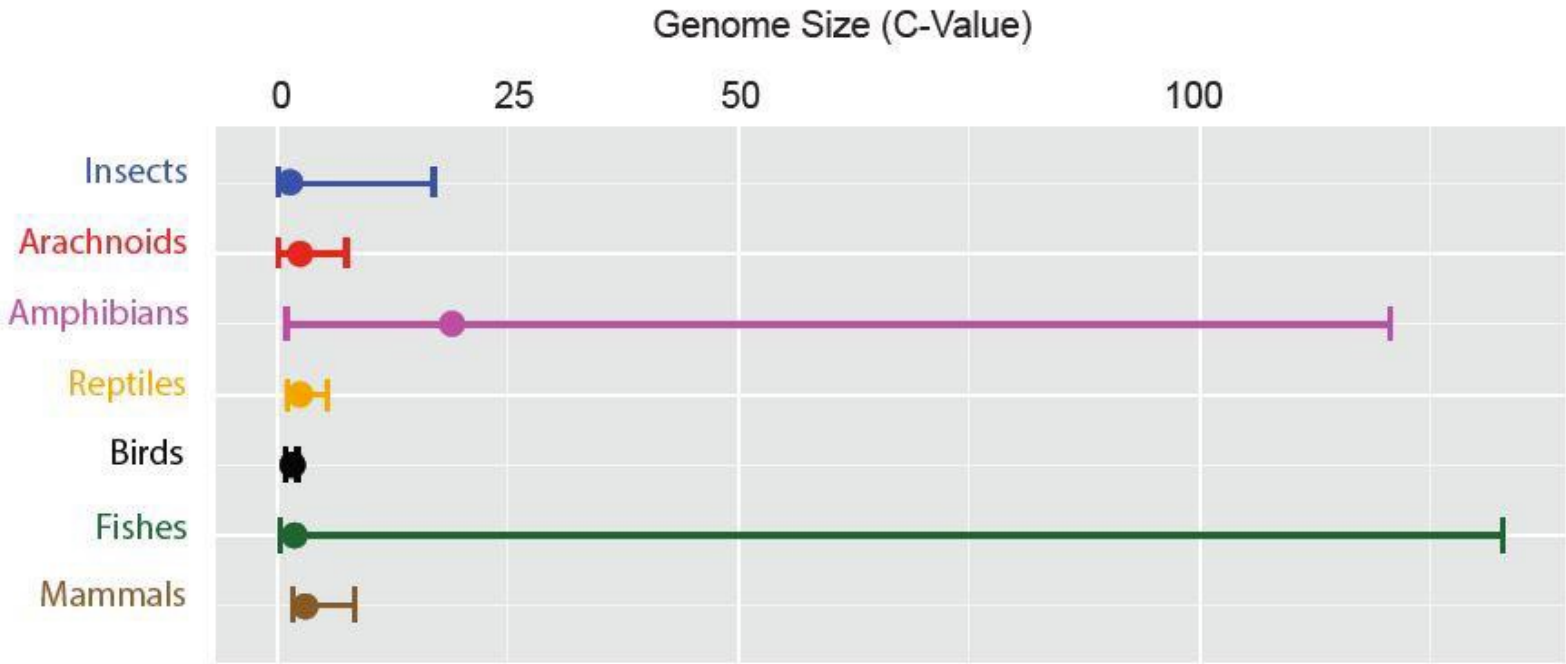
reads



Planning a WGS project:

Prior Information

Genome Size

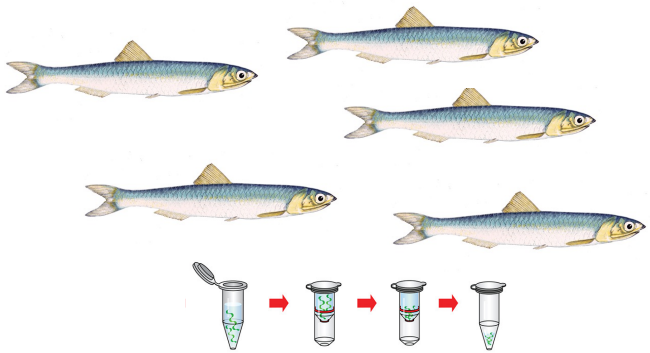


Planning a WGS project:

Prior Information

Sequence depth

COVERAGE?



15X coverage each sample

Reference Genome

1.55 Gb

~ 23 Gb data

Mapping Illumina short reads

- A **mapping algorithm** will try to find a **location in the reference** sequence that **matches the read**, while tolerating a certain amount of mismatch to allow subsequence variation detection.

- Practical challenge:
  - How quickly can we align millions of reads to a large reference genome (e.g. ~3 Gbp for human)?

- Strategic challenge:
  - How to **confidently map reads** originating from a **repetitive elements** in the reference?

Mapping workflow

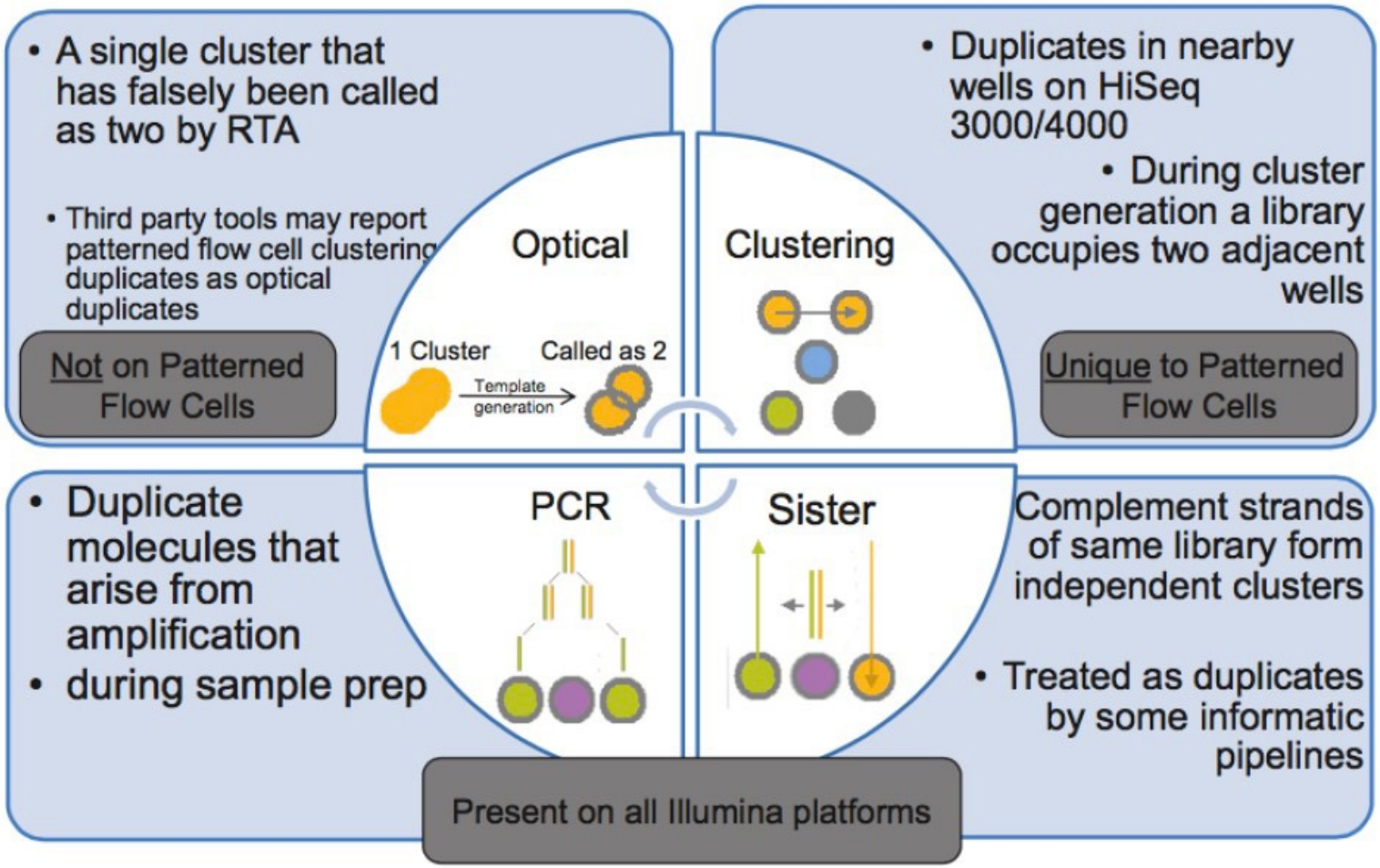
- **Indexing**: computational strategy to speed up algorithms.
  "Like the index at the end of a book, an index of a large DNA sequence allows one to rapidly find shorter sequences embedded within it."

- **Aligning**: finding the best match, storing coordinates and quality information.

- **Sorting**: sorting mapped reads by their coordinate position in the reference.

- **Mark duplicates**: mark potential PCR/optical duplicates reads from mapped data.

Duplicates reads



Sources of read duplicates in Illumina data.

SAM / BAM format

- **S**equence **A**lignment/**M**ap format (SAM):
  TAB-delimited text format.
  Consists of a header section (optional) and an alignment section.
  Each alignment line has 11 mandatory fields and variable number of optional fields

- **B**inary **A**lignment/**M**ap format (BAM):
  Binary data.
  Not human readable.
  Compressed.
  Quick access for computers.

SAM / BAM fields

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Header section

Alignment section

Optional fields in the format of TAG:TYPE:VALUE

QUAL: read quality; * meaning such information is not available

SEQ: read sequence

TLEN: the number of bases covered by the reads from the same fragment. Plus/minus means the current read is the leftmost/rightmost read. E.g. compare first and last lines.

PNEXT: Position of the primary alignment of the NEXT read in the template. Set as 0 when the information is unavailable. It corresponds to POS column.

RNEXT: reference sequence name of the primary alignment of the NEXT read. For paired-end sequencing, NEXT read is the paired read, corresponding to the RNAME column.

CIGAR: summary of alignment, e.g. insertion, deletion

MAPQ: mapping quality

POS: 1-based position

RNAME: reference sequence name, e.g. chromosome/transcript id

FLAG: indicates alignment information about the read, e.g. paired, aligned, etc.

QNAME: query template name, aka. read ID

https://www.samformat.info/sam-format-flag

SAM / BAM flags

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Sequence Alignment / Map Format Specification

https://broadinstitute.github.io/picard/explain-flags.html

```
#!/bin/bash

#SBATCH --job-name=01_mapping_reads
#SBATCH --error %x-%j.err
#SBATCH --output %x-%j.out

#SBATCH --reservation=mer02
#SBATCH --partition=vfast
#SBATCH --mem=10G
#SBATCH --ntasks=2

module load BWA/0.7.17-iccifort-2020.1.217
```

```
asm=#Your genome
IN=gscratch/ikasleXX/02_Mapping_reads

# index the assembly
bwa index -a 'bwtsw' ${asm}

bwa mem -t 64 -a -c 10000 ${asm} \
 ${IN}/C19_0011_1.paired.fq ${IN}/C19_0011_2.paired.fq \
 | samtools view -1 -b - > ${IN}/${asm}.paired.bam
bwa mem -t 64 -a -c 10000 ${asm} \
 ${IN}/C19_0011_1.unpaired.fq \
 | samtools view -1 -b - > ${IN}/${asm}.unpaired.1.bam

samtools merge -@ 63 ${IN}/${asm}.bam ${IN}/${asm}.paired.bam
${IN}/${asm}.umpaired1.bam

samtools sort -l 9 -@ 63 -T ${IN}/${asm}.bam -o ${IN}/${asm}.sort.bam

rm ${IN}/${asm}.paired.bam ${IN}/${asm}.umpaired1.bam
```

Variant calling

What is variant calling?

Identifying single nucleotide polymorphisms (SNPs) and small insertions and deletion (indels) from high-throughput sequencing data.

Conceptually simple:

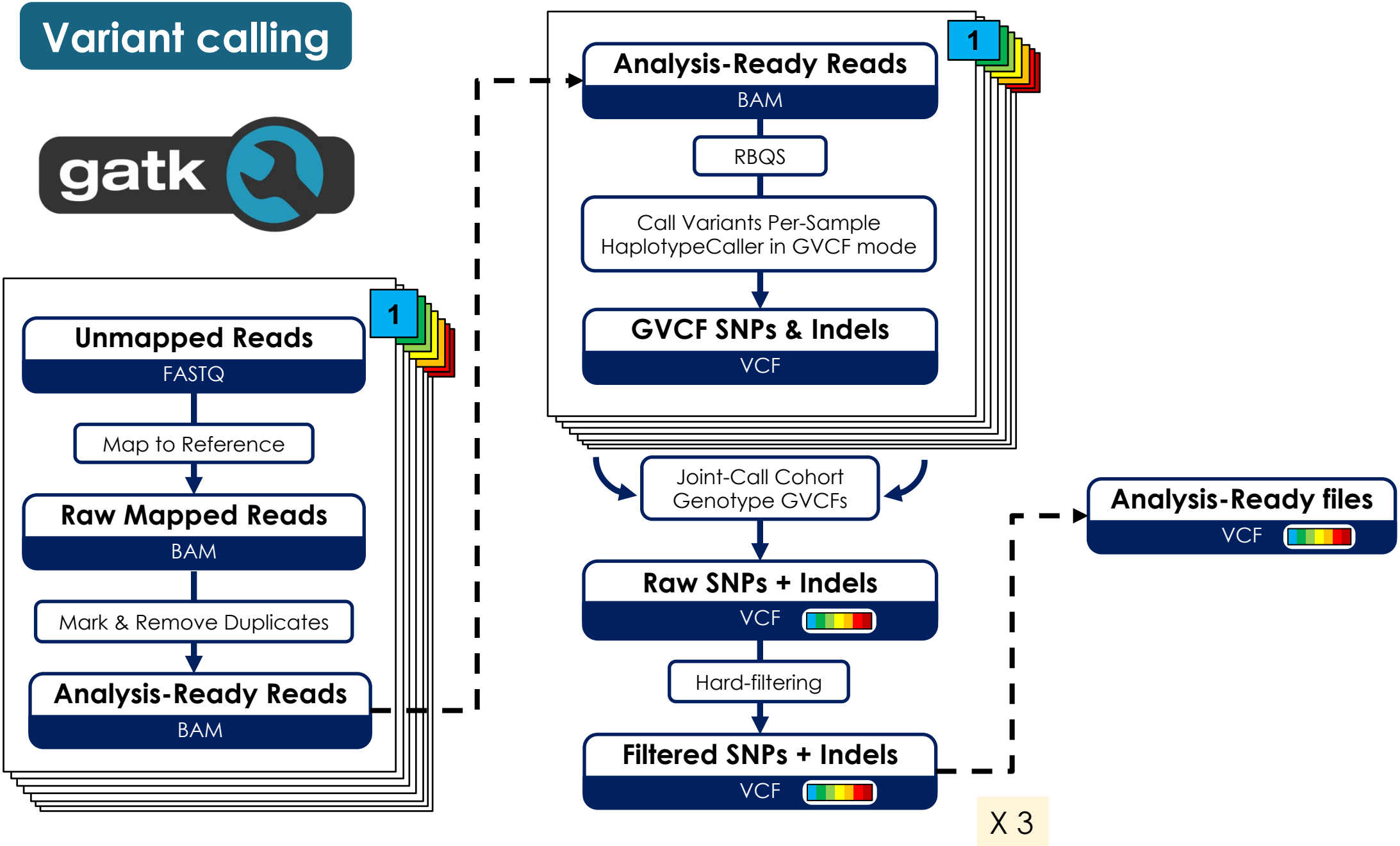
```
GGACGATGCTATCATAT
GGACGATGCTGTCATAT
```

The key challenge with NGS data is distinguishing which mismatches represent real mutations and which are just noise?

Many tools:

- **GATK**
- **SAMtools**
- **FreeBayes**
- **ANGSD**

Variant calling



Variant calling

FreeBayes

```
#!/bin/bash

#SBATCH --job-name=01_Freebayes
#SBATCH --error %x-%j.err
#SBATCH --output %x-%j.out

#SBATCH --reservation=mer02
#SBATCH --partition=vfast
#SBATCH --mem=1G
#SBATCH --cpus-per-task=5

module load freebayes/1.3.5-GCC-9.3.0-Python-3.8.2 VCFtools/0.1.16-GCC-9.3.0

REF="/02.1_Mapped_data/Salsal_genome.fa"
ls *_removed_duplicates.bam > bam.fofn

freebayes-parallel \
 <(fasta_generate_regions.py ${REF}.fai 100000) 10 \
 --fasta-reference ${REF} \
 --bam-list bam.fofn \
 --min-alternate-count 5 > /Salsal_FB_output.vcf
```

VCF files

```
ikasle01@kalk2020:~/gscratch/Anchovy_data$ bcftools view -H
snps.filtered.bcf | wc -l
738833
```

Filtering and handling VCFs

Randomly subsampling a VCF

```
ikasle01@kalk2020:~/gscratch/Anchovy_data$ bcftools view All_SNPs.vcf |
vcfrandomsample  -r 0.001 > Subset_SNPs.vcf
```

Generating statistics from a VCF

- **Depth**
- **Quality**
- **Minor allele frequency**
- **Missing data**

Filtering and handling VCFs

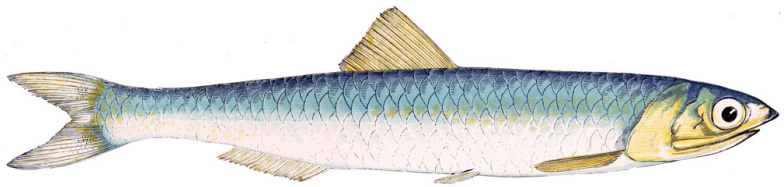
```
#!/bin/bash

#SBATCH --job-name=01_filtering_SNPs.sh
#SBATCH --error %x-%j.err
#SBATCH --output %x-%j.out

#SBATCH --reservation=mer02
#SBATCH --partition=vfast
#SBATCH --mem=2G
#SBATCH --ntasks=5

#Generating statistics from the subset VCFs

module load vcftools/0.1.17

# we will declare to variables to save us some typing below
SUBSET_VCF=/anchovy_subset.vcf.gz
OUT=/anchovy_snps_selected_subset
```

Filtering and handling VCFs

```
# Calculate allele frequency
vcftools --gzvcf $SUBSET_VCF --freq2 --out $OUT --max-alleles 2
# Calculate mean depth per individual
vcftools --gzvcf $SUBSET_VCF --depth --out $OUT
# Calculate mean depth per variant
vcftools --gzvcf $SUBSET_VCF --site-mean-depth --out $OUT
# Calculate site quality
vcftools --gzvcf $SUBSET_VCF --site-quality --out $OUT
# Calculate proportion of missing data per individual
vcftools --gzvcf $SUBSET_VCF --missing-indv --out $OUT
# Calculate proportion of missing data per site
vcftools --gzvcf $SUBSET_VCF --missing-site --out $OUT
# Calculate heterozygosity and inbreeding coefficient per individual
vcftools --gzvcf $SUBSET_VCF --het --out $OUT
```

Exercise

Population structure of the European anchovy



Exercise

Population structure of the European anchovy

a) Conduct a PCA from of unlinked SNPs from 50 anchovy specimens

Compute a covariance matrix with PCAngsd

(https://github.com/Rosemeis/pcangsd)

Exercise

Population structure of the European anchovy

a) Conduct a PCA from of unlinked SNPs from 50 anchovy specimens

Compute a covariance matrix with PCAngsd

(https://github.com/Rosemeis/pcangsd)

```
ikasle01@kalk2020:~/gscratch$ ./pcangsd_hwe.sh Anchovy_data Anchovy_PCA 4
```

bash script

input directory

output directory

number of cpus

Exercise

Population structure of the European anchovy

a) Conduct a PCA from of unlinked SNPs from 50 anchovy specimens

Check SNPs number after pruning and covariance matrix

```
ikasle01@kalk2020:~/gscratch/Anchovy_PCA$ head snps.ld_pruned.pcangsd.log
PCAngsd v.1.03
Using 10 thread(s).

Parsing Beagle file.
Loaded 193073 sites and 50 individuals.
Estimating minor allele frequencies.
EM (MAF) converged at iteration: 4
Number of sites after MAF filtering (0.05): 193073
```

Exercise

Population structure of the European anchovy

a) Conduct a PCA from of unlinked SNPs from 50 anchovy specimens

Check SNPs number after pruning and covariance matrix

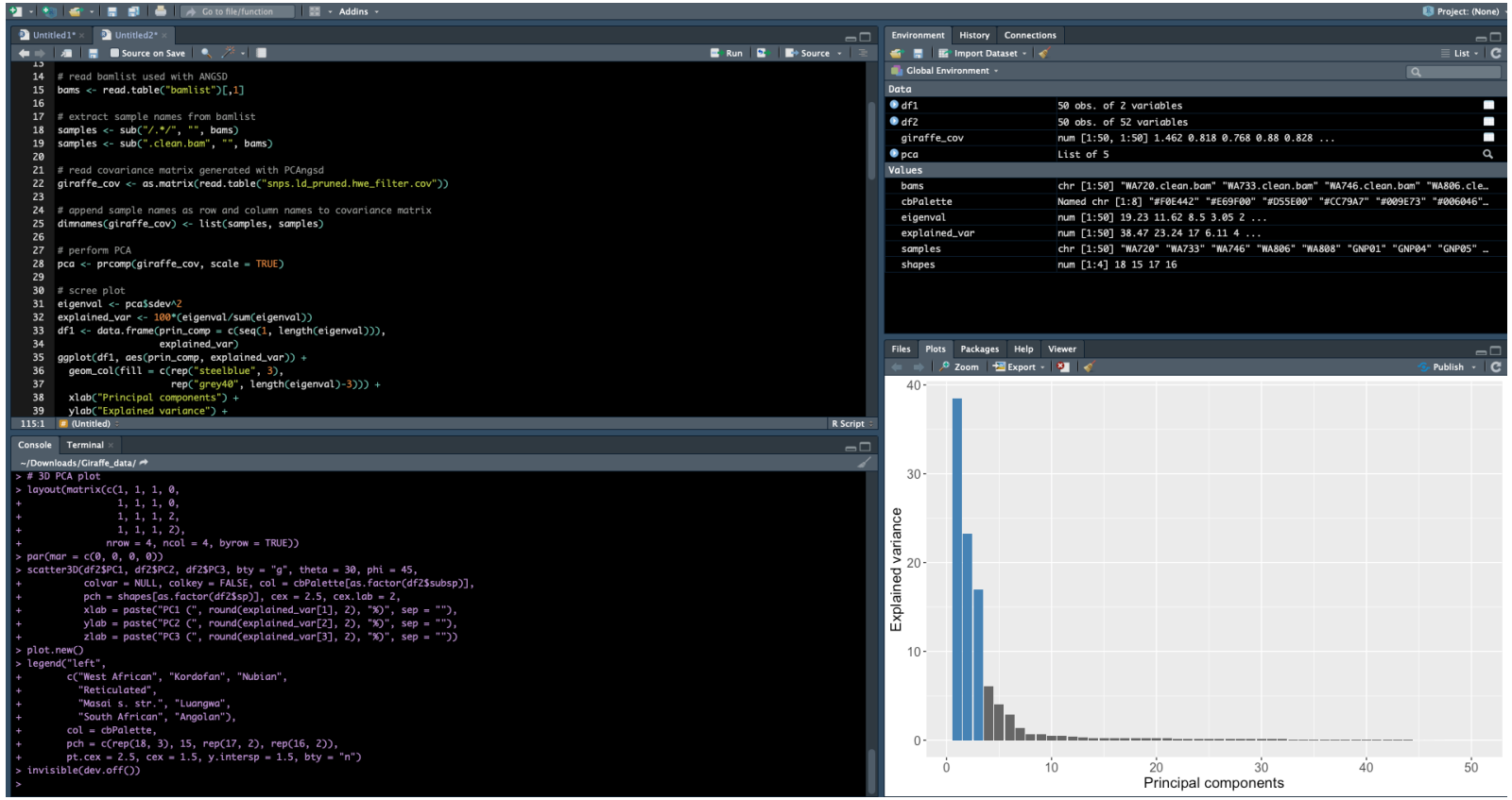
```
ikasle01@kalk2020:~/gscratch/Anchovy_PCA$ less snps.ld_pruned.hwe_filter.cov
02  -0.2200116
0.1346814   0.1432341   0.1272488   0.1358504   0.1431709   0.6824088
0.6813440   1.3195672   0.3790502   0.5609132   0.2032463   0.2082322
0.2052714   0.1867319   0.1891698   0.1902663   -0.1013756  0.0409134
-0.0772971  -0.0748142  -0.0787681  -0.0712066  -0.0663602  -0.0817849
-0.0849438  -0.0817924  -0.2020987  -0.2259787  -0.2233962  -0.2235594
-0.2244449  -0.2246796  -0.1969602  -0.2123237  -0.2153756  -0.2131487
-0.2111289  -0.2101994  -0.2061230  -0.2037562  -0.2020375  -0.2018170
-0.1933281  -0.2055342  -0.2162730  -0.2201347  -0.2194902  -0.2160828
-0.2188473  -0.2198513
```

Exercise

Population structure of the European anchovy

a) Conduct a PCA from of unlinked SNPs from 50 anchovy specimens

Perform the PCA and plot it in R



Exercise

Population structure of the European anchovy

b) Estimate Admixture proportions with NGSAdmix

(http://www.popgen.dk/software/index.php/NgsAdmix)

```
ikasle01@kalk2020:~/gscratch$ screen -S NGSadmix
ikasle01@kalk2020:~/gscratch$ ./ngsadmix.sh Anchovy_data Anchovy_Admixture
2 10 4
```

k min

k max

number of cpus

bash script

input directory

output directory

Exercise

Population structure of the European anchovy

b) Estimate Admixture proportions with NGSAdmix

(http://www.popgen.dk/software/index.php/NgsAdmix)

```
ikasle01@kalk2020:~/gscratch$ screen -S NGSadmix
ikasle01@kalk2020:~/gscratch$ ./ngsadmix.sh Anchovy_data Anchovy_Admixture
2 10 4 1>./Anchovy_Admixture/ngsadmix.err
2>./Anchovy_Admixture/ngsadmix.log
```

Exercise

Population structure of the European anchovy

b) Estimate Admixture proportions with NGSAdmix

Plot Admixture results in R