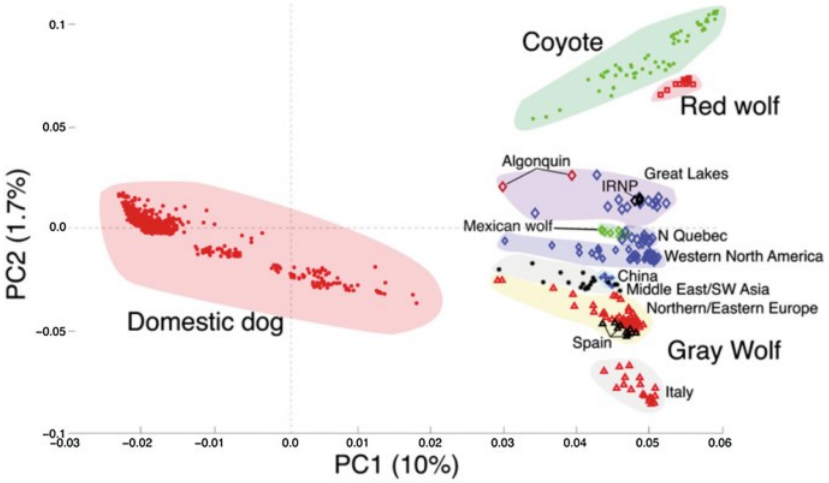
Introduction to Population Genomics:
The use of **Aristotelis HPC**

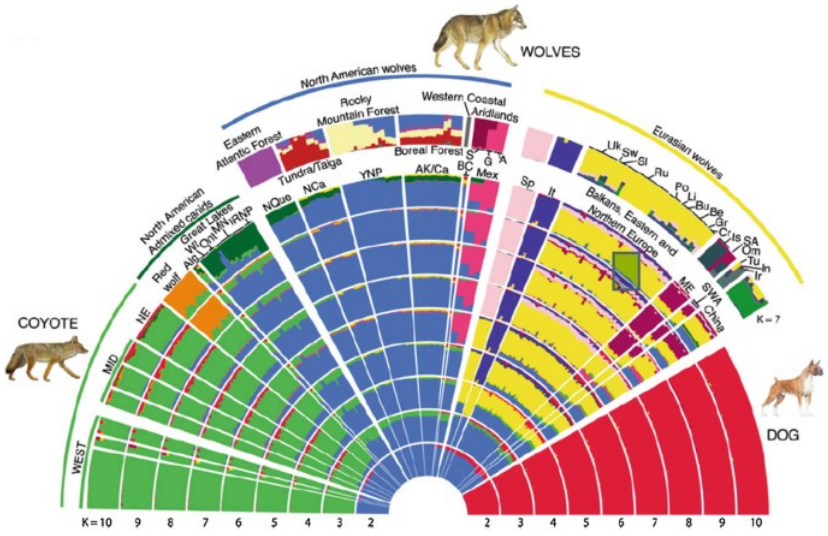
19-06-2025
Σαγώνας Κωνσταντίνος
https://www.sagonaslab.com/

Population Genomics

Main questions:

- **Footprints of natural and artificial selection**
- **Local adaptation to changing environments**
- **Population structure, gene flow, (adaptive) introgression**
- Evolution of genomic variation within and between species
- Deleterious mutations /Conservation Biology
- Speciation / Hybrid zones
- **Methods in population genomics (demographic inferences, genome scans/GWAS, …)**
- **Population Epigenetics**
- **Population Transcriptomics**





Population Genomics

General strategy

Raw Sequencing Data

Quality Control — *fastqc*, *multiqc*

Read trimming — *trimmomatic*, *cutadapt*, *trimgalore*

Read mapping — *Burrows-Wheeler Aligner*, *bowtie2*
--very-fast vs. --very-sensitive modes in **Bowtie2**
STAR and HISAT2 for RNAseq data

Filtering (e.g. deduplication) — *picard*, *samtools*

SNP caller for low-coverage data, *e.g.* ANGSD

Traditional SNP caller, *e.g.* GATK, Stacks, ipyRAD, Vcftools, Bcftools

Allele counts data, *e.g.* synchronized mpileup

Genomic Data

One of the first things we are interested in is: How many sequences are present in my fastq file?

Fastq.gz files:

zcat AE13_R1.fastq.gz | wc –l #(divide by 4)

Trimming reads has the objectives:

- Removing adapters
- Removing low quality bases
- Excluding short reads after quality trimming

Genomic Data

Mapping Reads

Read mapping corresponds to a balance between **speed and accuracy**:

- Faster algorithms find approximate positions (mapping)
- Slower, precise methods (alignment) match each base accurately

Software often allows fine-tuning of detection, e.g. **--very-fast** vs. **--very-sensitive** modes in Bowtie2

Different format used for the outputs: **SAM** (text-based) / **BAM** (binary) / **CRAM** (compressed)

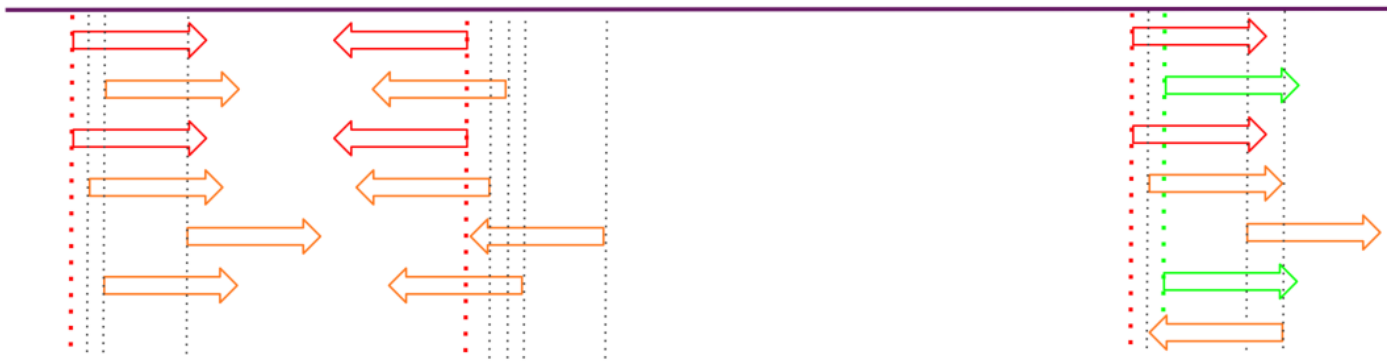
Summarizing the results of the mapping (e.g., **bamtools stats** / **samtools flagstat**)

Genomic Data

Mapping Reads

Mapping qualities (MAPQ; expressed in -10log10 probability that the mapping position is wrong)

Filtering low-confidence reads (e.g. MAPQ <5, <20) is generally performed since these mapping are more error-prone, excluding them therefore improve downstream analysis such as SNP calling



Mark (not remove) duplicates!

- **Picard** (Markduplicates)
- **Samtools** (markdup)

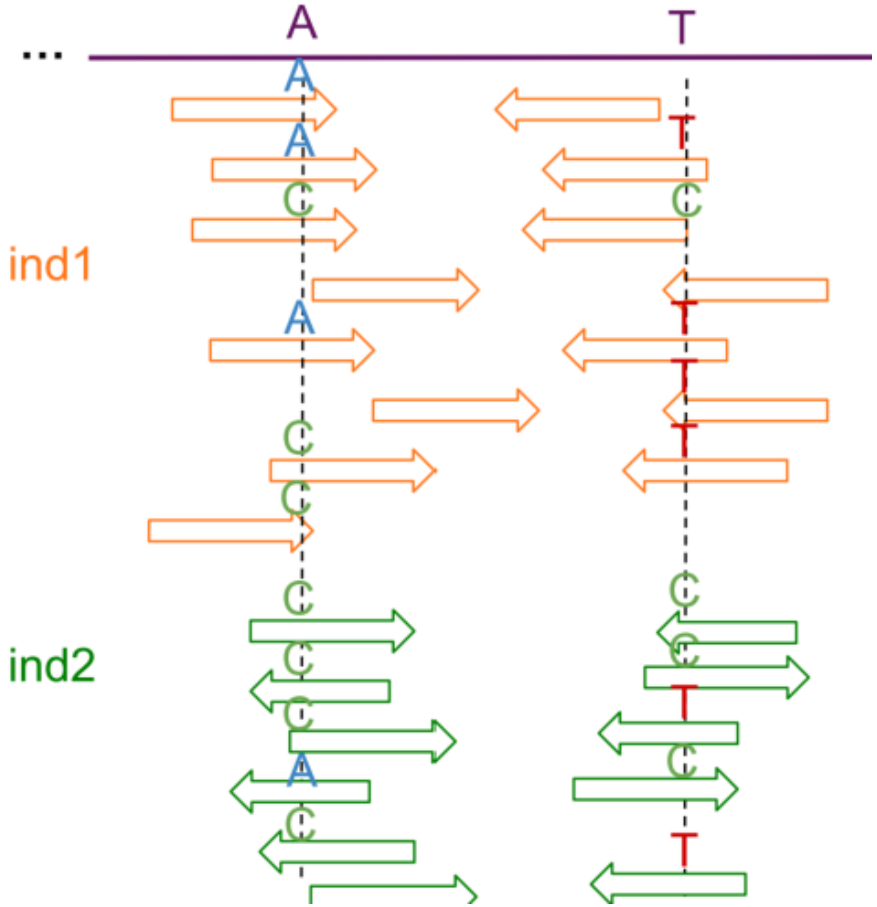
Genomic Data

Variant Calling

Multi-sample variant calling. By analyzing all individuals together, variant callers can:

- Increase sensitivity: Detect low-frequency variants that might be missed in single-sample analyses
- Improve accuracy: Use allele frequency data to distinguish true variants from errors
- Enable joint genotyping: Call genotypes across all samples consistently, which helps in downstream analyses like population genetics and GWAS

GATK, **FreeBayes**, **Vcftools**, **Bcftools**, **etc**



Genomic Data

Variant Calling

Selecting most reliable variants

```
136 ##contig=<ID=chrUn_gl000247,length=36422,assembly=hg19>
137 ##contig=<ID=chrUn_gl000248,length=39786,assembly=hg19>
138 ##contig=<ID=chrUn_gl000249,length=38502,assembly=hg19>
139 ##reference=file:///l3bioinfo/ucsc.hg19.fasta
140 ##source=SelectVariants
141 #CHROM  POS ID  REF ALT QUAL  FILTER  INFO  FORMAT  150017112327 57A
142 chr1  14464 . A T 810.77  PASS  AC=1;AF=0.500;AN=2;BaseQRankSum=0.152;ClippingRankSum=0.00;DP=56;Excess
143 chr1  15274 . A T 1084.77 PASS  AC=2;AF=1.00;AN=2;DP=43;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
144 chr1  28563 . A G 139.90  PASS  AC=2;AF=1.00;AN=2;DP=5;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=
145 chr1  49298 . T C 515.77  PASS  AC=2;AF=1.00;AN=2;DP=17;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
146 chr1  52238 . T G 716.77  PASS  AC=2;AF=1.00;AN=2;DP=22;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
147 chr1  55926 . T C 120.90  PASS  AC=2;AF=1.00;AN=2;DP=5;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=
148 chr1  61442 . A G 314.77  PASS  AC=2;AF=1.00;AN=2;DP=10;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
149 chr1  61947 . C T 397.77  PASS  AC=1;AF=0.500;AN=2;BaseQRankSum=3.01;ClippingRankSum=0.00;DP=33;ExcessH
150 chr1  61987 . A G 703.77  PASS  AC=1;AF=0.500;AN=2;BaseQRankSum=0.426;ClippingRankSum=0.00;DP=42;Excess
151 chr1  61989 . G C 703.77  PASS  AC=1;AF=0.500;AN=2;BaseQRankSum=0.125;ClippingRankSum=0.00;DP=41;Excess
152 chr1  69511 . A G 358.77  PASS  AC=2;AF=1.00;AN=2;DP=13;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
153 chr1  83084 . T A 204.80  PASS  AC=2;AF=1.00;AN=2;DP=7;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=
154 chr1  89946 . A T 613.77  PASS  AC=2;AF=1.00;AN=2;DP=19;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ
```

Genomic Data

Variant Calling

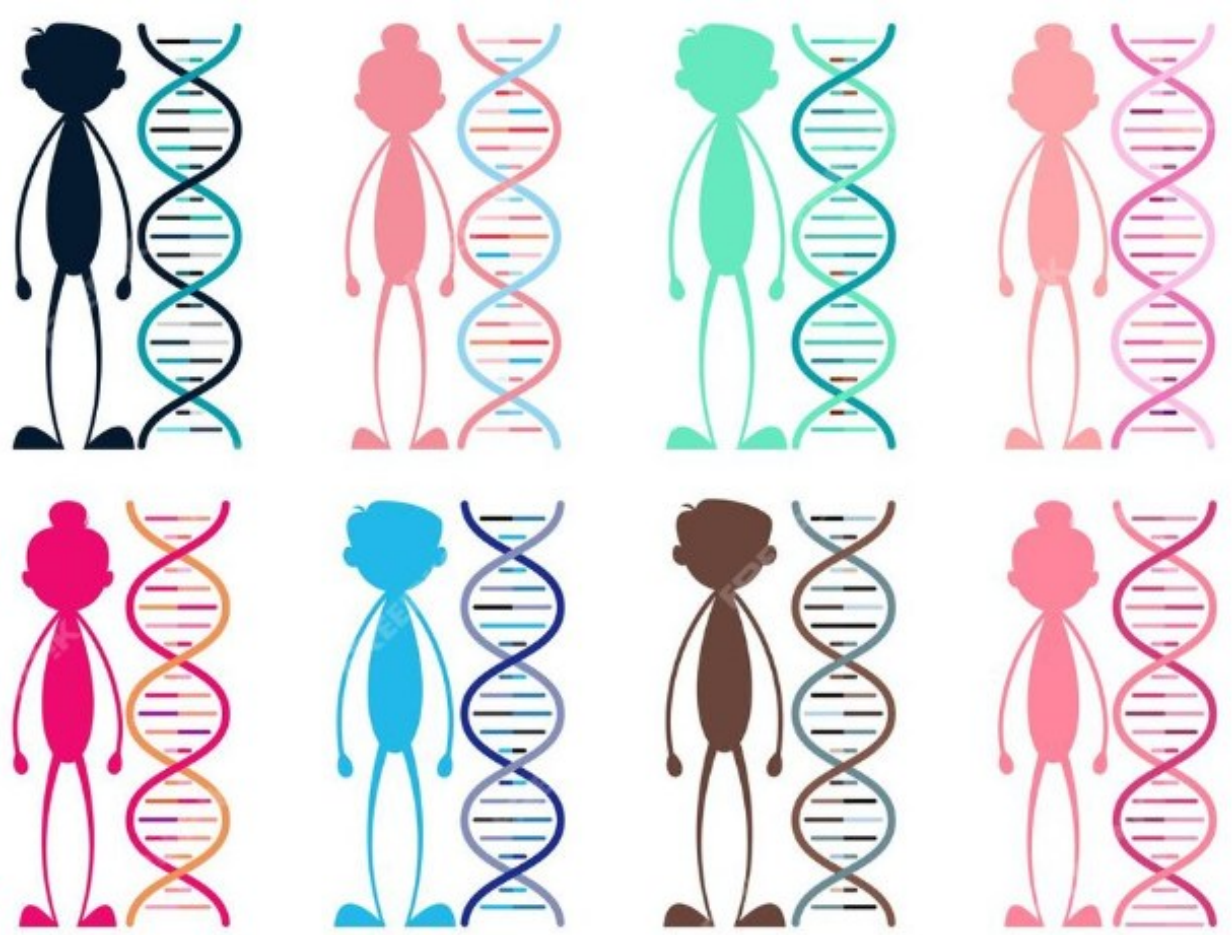
Selecting most reliable variants

Quality by Depth (QD): ensures that variants have sufficient supporting read depth
Mapping Quality (MQ): filtering out variants associated with mapping errors
Base Quality (BQ): average base quality of the reads spporting the variant, filtering out variants with low base quality
Minimum Allele Frequency (AF): filtering out variants with low allele frequency
Depth of Coverage (DP): ensures that the variant is supported by a large number of reads
Variant type (SNPs vs. indels): calling INDELs is more challenging than SNPs
Missing rate: filtering out variants with excessive missing data is a standard to reduce false positives (true also at the sample level)

Population Genomics

The study of changes in allele frequency over time and space

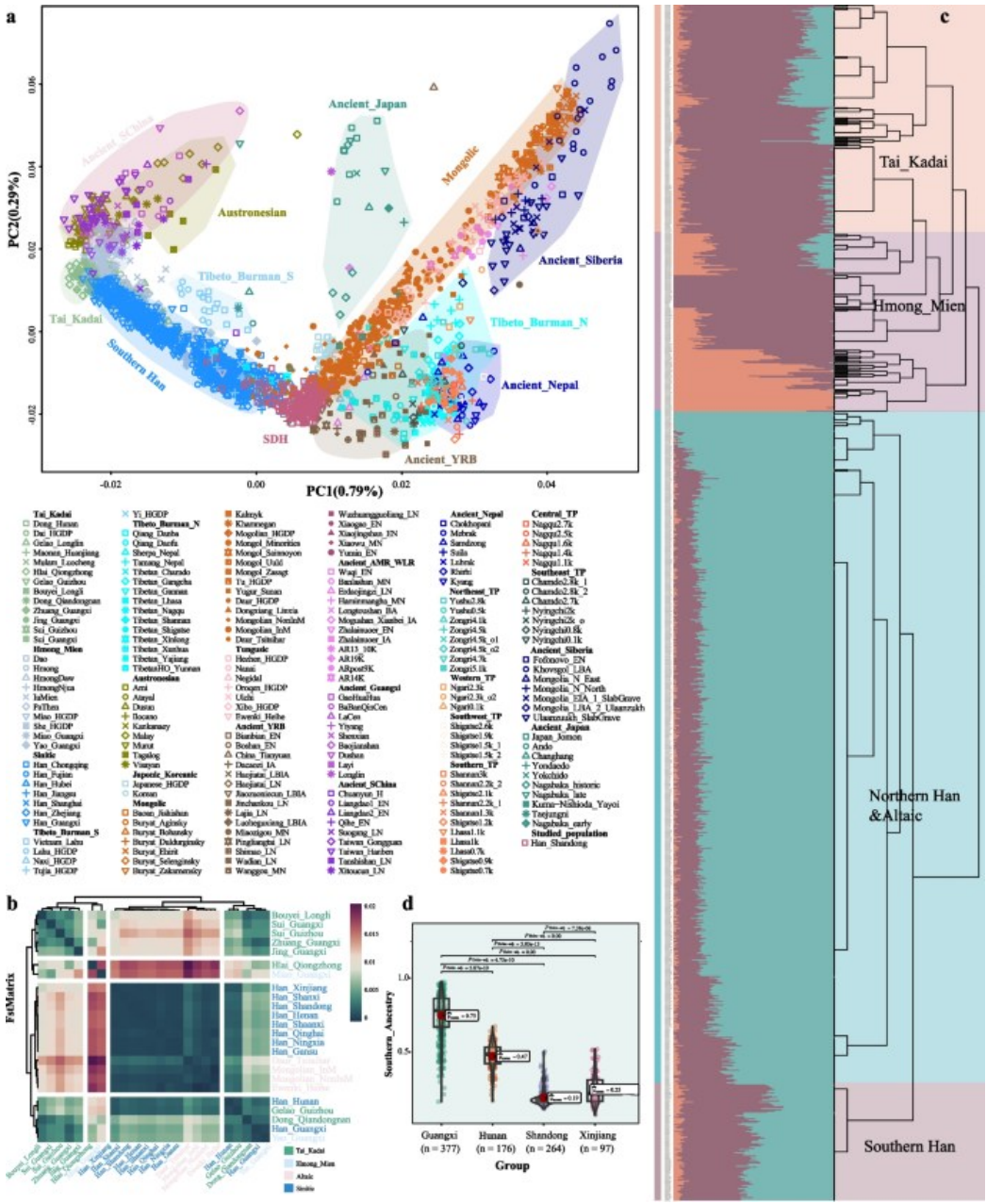
With new technologies we can examine millions of "independent" markers along the genome. We can further detect which polymorphisms segregate in a way that is not consistent with others (e.g. selection)



Population Genomics

How to deal with thousands or millions of markers?

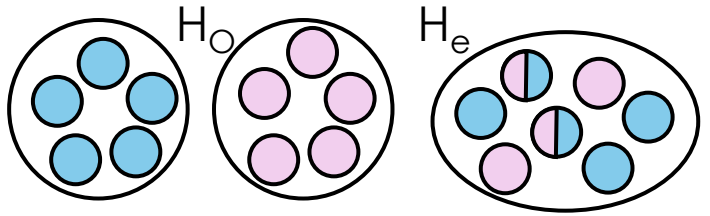
- Measures of pairwise relatedness, pedigrees, genetic distances, summary statistics (F-statistics, ABBA-BABA...)
- Dimensionality-reduction (e.g., DAPC, PCA, NMDS)
- Clustering of individuals (can be model-based; e.g., ADMIXTURE)
- Tree based methods (phylogenetics)

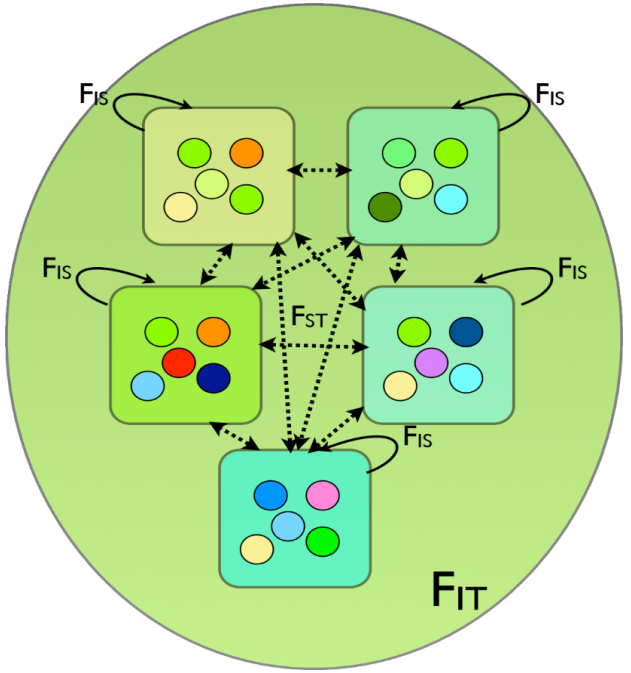


Describing population structure using allele frequencies

F-statistics measure the reduction in heterozygosity relative to HW expectations in specific populations (subpopulation).

$F = 1 - (H_O / H_E)$ **or** $F = (H_E - H_O)/H_E$





FIS: probability that two alleles in an individual are identical by descent (≈ *F* averaged across all individuals) - intra-population

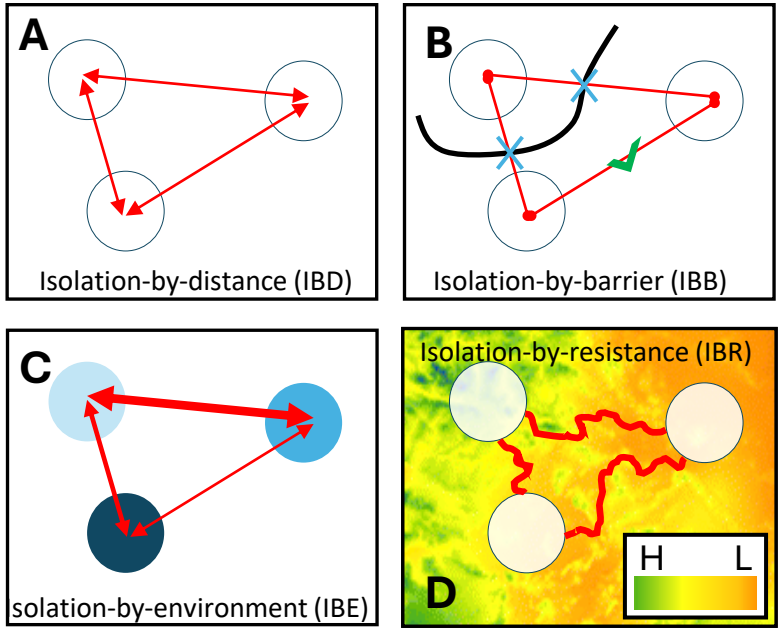
FST: fixation index - probability that two alleles from two populations are identical by descent

FIT: general genetic structure

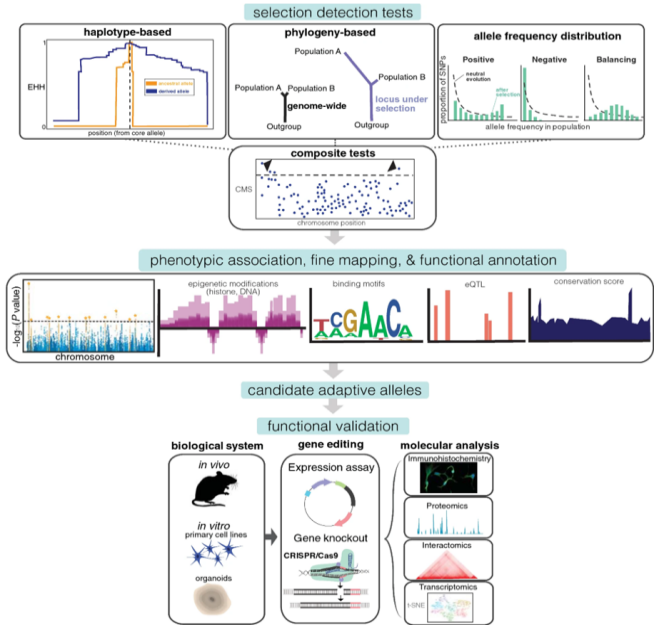
In real life, we use different estimators (Weir an Cockerham, Bhatia, GST…)

Integrate other data on population genomics

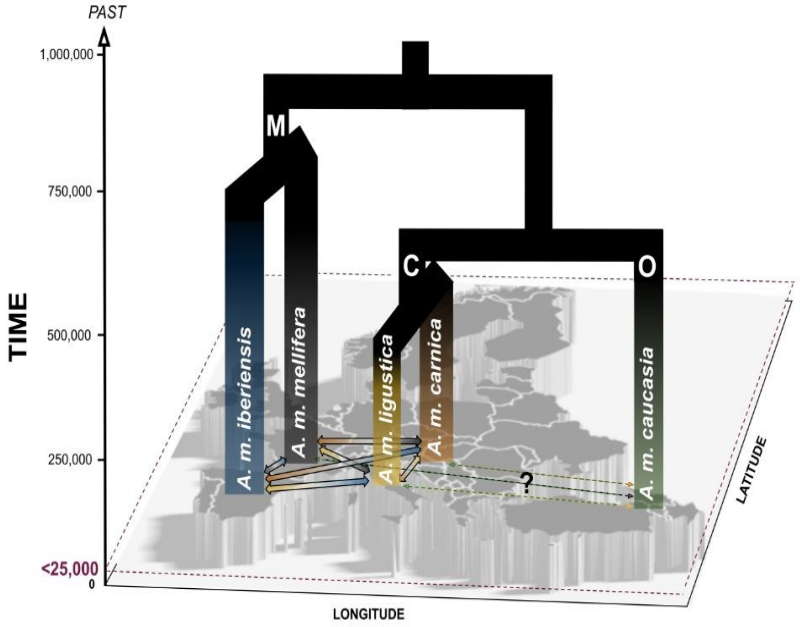
Landscape genomics



GWAS



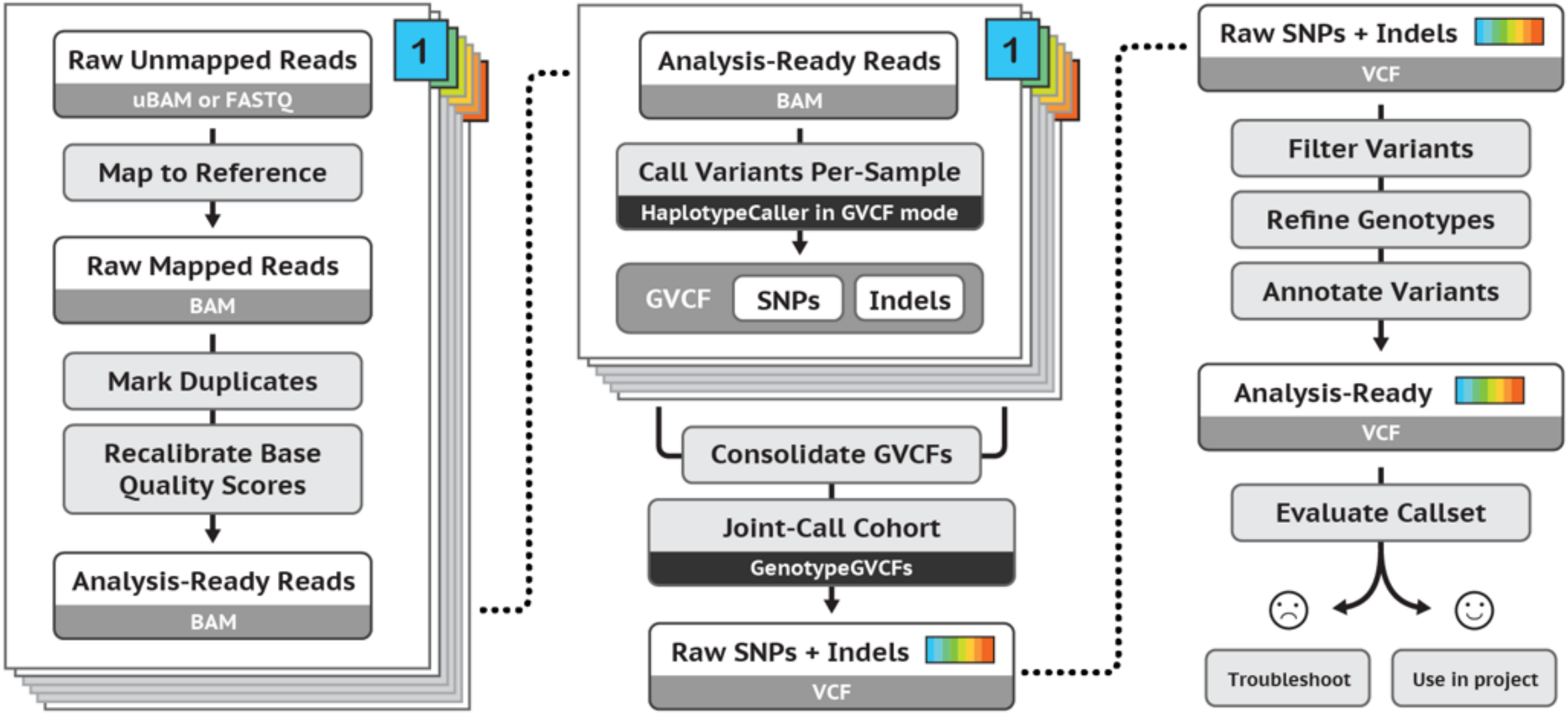
Demographic modeling



GATK

Genome Analysis Toolkit

Variant Discovery in High-Throughput Sequencing Data


Raw Unmapped Reads
uBAM or FASTQ
1
Map to Reference
Raw Mapped Reads
BAM
Mark Duplicates
Recalibrate Base Quality Scores
Analysis-Ready Reads
BAM
Analysis-Ready Reads
BAM
1
Call Variants Per-Sample
HaplotypeCaller in GVCF mode
GVCF
SNPs
Indels
Consolidate GVCFs
Joint-Call Cohort
GenotypeGVCFs
Raw SNPs + Indels
VCF
Raw SNPs + Indels
VCF
Filter Variants
Refine Genotypes
Annotate Variants
Analysis-Ready
VCF
Evaluate Callset
Troubleshoot
Use in project

Step 1: HaplotypeCaller

- Command: gatk HaplotypeCaller
- Generates a GVCF per-sample for joint genotyping.
- Important options:
- -R: Reference genome (FASTA)
- -I: Input BAM file
- -O: Output GVCF file
- -ERC GVCF: Emit reference confidence mode

Step 2: GenomicsDBImport

- Command: gatk GenomicsDBImport
- Converts per-sample GVCFs into a database format for efficient joint calling.
- Important options:
- --genomicsdb-workspace-path: Output database path
- --batch-size: Number of samples per batch
- --intervals: List of genomic intervals (chromosomes)
- -V: Input GVCFs

Step 3: GenotypeGVCFs

- Command: gatk GenotypeGVCFs
- Performs joint genotyping on the database produced in previous step.
- Important options:
- -R: Reference genome
- -V: gendb:// path to GenomicsDB
- -O: Output VCF file

Step 4: Variant Filtration

- Two main options: Hard Filtering or Variant Quality Score Recalibration (VQSR)
- Command (Hard Filtering): gatk VariantFiltration
- Important expressions:
- - QD < 2.0
- - FS > 60.0
- - MQ < 40.0
- - SOR > 3.0
- Use SelectVariants before this to filter SNPs or INDELs specifically

Hard Filtering vs. VQSR

- Hard Filtering: Uses fixed thresholds for filtering variants.
- Pros: Simple, no external data required.
- Cons: Less flexible, risk of over/under-filtering.

- VQSR: Builds statistical models using known, high-quality variant datasets.
- Pros: More accurate, adaptive.
- Cons: Requires large datasets and known resources (truth sets).

Post-Variant Calling Analyses in R

📊 Population Genetic Indices

- adegenet – diversity, clustering, PCA
- hierfstat – F-statistics
- pegas – haplotype and allele frequency analysis
- poppr – clone correction, AMOVA

🌍 Landscape Genomics

- LEA – genotype-environment associations (LFMM)
- ResistanceGA – isolation-by-resistance modeling
- gdm – generalized dissimilarity modeling
- raster / terra – spatial data handling

👥 Demographic Inference

- fastsimcoalR – simulate demographic scenarios
- abc – approximate Bayesian computation
- msprime, SLiM – demographic simulations

🧬 Structure & Relatedness

- SNPRelate – PCA, IBS, LD analysis
- adegenet – DAPC for population structure
- pcadapt – outlier detection for structure
- starmie – STRUCTURE plot visualization

📦 Visualization

- ggplot2 / ggpubr – elegant data plotting
- ggmap / sf – spatial visualizations
- plotly – interactive PCA, STRUCTURE plots

Registration Open for GBCC2025:
Joint Galaxy/Bioconductor Conference!

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

About Learn Packages Developers Search Get Started

Open source software for Bioinformatics

The Bioconductor project aims to develop and share open source software for precise and repeatable analysis of biological data.
We foster an inclusive and collaborative community of developers and data scientists.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(version = "3.21")

BiocManager::install("packagename")
```

Get started