PRO-GRACE

PROMOTING A PLANT GENETIC RESOURCE COMMUNITY FOR EUROPE

DELIVERABLE 3.5

Demonstration of bioinformatic methods and services for kinship/population structure/pedigree determination, gap analyses, GWAS and QTL analyses

This deliverable has been submitted and is currently pending approval by the European Commission.

Call identifier: HORIZON-INFRA-2022-DEV-01-01
PRO-GRACE

Grant agreement no: 101094738

Promoting a plant genetic resource community for Europe

Deliverable No. 3.5

Demonstration of bioinformatic methods and services for kinship/population structure/pedigree determination, gap analyses, GWAS and QTL analyses

Contractual delivery date:

33

Actual delivery date:

34

Responsible partner:

UNITO

Contributing partners:

(Partners' short names; ENEA, INRAE)

This project has received funding from the European Union's Horizon Europe research and innovation programme

under grant agreement No 101094738.

Grant agreement no.	Horizon Europe – 101094738
Project full title	PRO-GRACE – Promoting a plant genetic resource community for Europe

Deliverable number	**D3.5**
Deliverable title	**Demonstration of bioinformatic methods and services for kinship/population structure/pedigree determination, gap analyses, GWAS and QTL analyses**
Type	R
Dissemination level	PU
Work package number	3
Author(s)	Lorenzo Barchi, Luciana Gaccione, Giuseppe Aprea, Maria Tiziana Sirangelo, Vèronique Lefebvre, Giovanni Giuliano
Keywords	

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101094738**.**

Table of Contents

Executive Summary

Deliverable D3.5 presents the implementation and demonstration of advanced bioinformatic workflows and services supporting the characterization and exploitation of plant genetic resources in Europe. The work focuses on developing and validating standardized pipelines for variant discovery, genotypic data filtering, population structure and kinship estimation, gap analyses of genetic diversity, and association mapping (GWAS and QTL).

The deliverable provides comprehensive protocols for SNP and structural variant calling using next-generation sequencing (NGS) data, leveraging established tools such as GATK, DeepVariant, and bcftools. Detailed filtering strategies are described to ensure high-quality variant datasets suitable for downstream analyses. Both hard filtering and machine-learning–based methods (VQSR) are evaluated to optimize accuracy across different crop datasets.

Population genetic analyses are demonstrated through the estimation of population structure and kinship matrices, employing approaches such as PCA, ADMIXTURE/sNMF, and VanRaden's method. These analyses enable robust control of confounding factors in genomic studies and the identification of duplicates or closely related accessions within genebank collections.

A gap analysis framework is presented to assess the representativeness of germplasm collections with respect to nucleotide diversity. Using the G2P-SOL tomato core collection as a case study, the analysis quantifies how genetic diversity saturates with increasing sample size, offering a practical approach for identifying redundancy or missing diversity within collections.

The deliverable further details a complete pipeline for Genome-Wide Association Studies (GWAS) and Quantitative Trait Locus (QTL) analyses. These pipelines integrate advanced statistical models implemented in GAPIT3 (including MLM, FarmCPU, and BLINK) and standard QTL mapping software (R/qtl, JoinMap). Demonstration activities using the Capsicum (pepper) G2P-SOL core collection showcase the identification of genomic regions linked to key agronomic and morphological traits across multiple environments. A total of 207 significant markers associated with 23 traits were identified, mapped to 112 genomic regions, and prioritized for candidate gene identification.

Overall, D3.5 delivers a harmonized suite of bioinformatic methods, open workflows, and demonstrative case studies that strengthen Europe's capacity for genomic characterization of plant genetic resources.

D3.5 finally points out that due to the high complexity and computational demands of bioinformatic analyses, it is more effective to centralize these activities within a few specialized hubs of the GRACE infrastructure. This coordinated model provides the

necessary expertise, standardized workflows, and computing capacity, ensuring data quality, reproducibility, and consistency while allowing genebanks to benefit from advanced analyses without developing local infrastructures.

SNPs calling

In modern genomics, the accurate interpretation of sequencing data relies heavily on two main steps: aligning sequencing reads to a reference genome and identifying genetic variants through variant calling.

Raw sequencing data (typically in FASTQ format) undergo quality control to remove low-quality sequences and technical artifacts. Tools such as FastQC[1], Trimmomatic[2], fastp[3] and Cutadapt[4] are used to assess base quality, detect adapter contamination, and trim unwanted sequences.

```
#example code for cleaning reads using fastp#
fastp -i ${READ}_1.fq.gz -I ${READ}__2.fq.gz -o
${READ}_clean_1.fq.gz -O ${READ}_clean_2.fq.gz
```

Subsequently, cleaned reads must be aligned against a reference genome of the species. To perform an efficient alignment, the genome is pre-processed into an indexed format. Most aligners rely on either:

- **Burrows-Wheeler Transform (BWT)**: Used in tools like **BWA-MEM**[5] and **Bowtie2**[6], this algorithm enables fast and memory-efficient exact and inexact string matching.
- **Hash-based indexing**: Employed by older or specialized aligners, using fixed-length k-mers for quick lookup.

During alignment, each read is compared to the reference genome to find the best possible match, allowing for mismatches, insertions, deletions, or sequencing errors. The aligner outputs a SAM (a **human-readable, plain-text** format)/BAM (the **binary, compressed** version of SAM, optimized for storage and computational efficiency) file that contains alignment positions, mapping quality scores, and other metadata. The BAM file is subsequently sorted and indexed for downstream analyses.

The most used aligners are:

- **BWA**[5]: The most widely used aligner for short-read DNA sequencing (e.g., WGS/WES). It provides accurate alignment around indels and handles paired-end data efficiently.
- **BWA-MEM2**[7] is an improved and faster version of the original **BWA-MEM** algorithm. It was developed by the Broad Institute as a drop-in replacement for BWA-MEM, designed to enhance performance while maintaining identical output.
- **BWA-MEME**[8] (BWA-MEM Enhanced) is a **variant of BWA-MEM** specifically designed to improve alignment accuracy around long insertions and deletions (indels).

- **Bowtie2**[6]: Optimized for speed and memory usage, suitable for small genomes and high-throughput applications, though less accurate for clinical variant detection.
- **Minimap2**[9]: Designed for long-read sequencing (e.g., PacBio, Oxford Nanopore), offering fast and accurate mapping for both genomic and transcriptomic reads.
- **HISAT2**[10] **and STAR**[11]: Splice-aware aligners primarily used for RNA-seq data to identify exon-exon junctions accurately.

```
#example code for bwa and short reads#
bwa index $REF
bwa mem -t ${THREADS} ${REF} ${READ}_clean_1.fq.gz
${READ}_clean_2.fq.gz | samtools view -@ ${THREADS} -b - |
samtools sort -@ ${THREADS} -o ${OUT}.sorted.bam
samtools index ${OUT}.sorted.bam
#example code for minimap2 and PacBio long reads#
minimap2 -ax map-pb -t ${THREADS} ${REF} ${READS}.fq.gz |
samtools view -@ ${THREADS} -b - | samtools sort -@ ${THREADS}
-o ${OUT}.sorted.bam -
samtools index ${OUT}.sorted.bam
```

After alignment, **marking PCR duplicates** with tools like **Picard**[12] or **Sambamba**[13] is typically performed to prepare data for variant calling. Marking PCR duplicates is mandatory for WGS, while for reduced representation sequencing methodologies like SPET and GBS, it should be avoided.

```
#example code for MarkDuplicates #
GATK MarkDuplicates --java-options -Xmx30g -I
${OUT}.sorted.bam -O ${OUT}.md.bam -M ${OUT}.metrics.txt
```

Once reads are aligned, the next step is to identify variants in the genome where the sample differs from the reference. These include single nucleotide polymorphisms (SNPs), insertions and deletions (indels), and structural variants (SVs).

The main Variant Calling Tools are:

- **GATK HaplotypeCaller**[14]: Performs local haplotype reassembly to accurately detect SNPs and indels. It is included in the GATK suite and supports joint genotyping of multiple samples and is the gold standard in many pipelines.
- **DeepVariant**[15]: A deep learning-based caller developed by Google, known for high accuracy in both germline and somatic variant calling.
- **FreeBayes**[16]: A Bayesian variant caller suitable for pooled and multisample datasets. It is robust in detecting complex alleles and indels.
- **bcftools**[17]: A lightweight tool built on Samtools for simple variant calling from pileups, useful in fast or low-resource settings.
- **Strelka2**[18] and **Octopus**[19]: High-performance callers that balance speed and accuracy, particularly strong in detecting somatic variants and low-frequency alleles.
- **Clair3**[20]: A neural network-based caller optimized for long-read sequencing technologies.

```
#example code for GATK with multiple samples#
GATK HaplotypeCaller -R ${REF} -I ${OUT}.md.bam -O
${OUT}.g.vcf.gz -ERC GVCF

GATK HaplotypeCaller -R ${REF} -I ${OUT2}.md.bam -O
${OUT2}.g.vcf.gz -ERC GVCF
GATK GenomicsDBImport --genomicsdb-workspace-path my_database
--sample-name-map sample_map.txt --intervals intervals.list --
batch-size 50
GATK GenotypeGVCFs -R ${REF} -V gendb://my_database -O
cohort.vcf.gz

#example code for DeepVariant with multiple samples with
glnexus for merging cohorts of gvcf#

$DEEPVARIANT_BIN --model_type=WGS -ref=${REF} -
reads=${OUT}.md.bam --output_vcf=${OUT}.vcf.gz --
output_gvcf=${OUT}.g.vcf.gz --num_shards=${THREADS}

$DEEPVARIANT_BIN --model_type=WGS -ref=${REF} -
reads=${OUT2}.md.bam --output_vcf=${OUT2}.vcf.gz --
output_gvcf=${OUT2}.g.vcf.gz --num_shards=${THREADS}
glnexus_cli --config DeepVariantWGS --threads ${THREADS} --
list "GVCF_LIST_FILE" > cohort_merged.vcf
```

Structural Variant Calling

Structural Variants (SVs) are large-scale genomic alterations that involve segments of DNA typically greater than 50 base pairs in length. They include a variety of genomic rearrangements such as deletions, duplications, insertions, inversions, and translocations. Structural variants can significantly impact gene function and regulation by disrupting coding sequences, altering gene dosage, or modifying regulatory regions. As a result, they play a crucial role in diversity, evolution, and disease susceptibility.

Detecting large-scale genomic rearrangements requires specialized tools, including:

- **Manta**[21], **Delly2**[22], **Lumpy**[23], **SvABA**[24], **Dysgu**[25], **Parliament2**[26]: Suitable for short-read SV detection.
- **Sniffles2**[27], **cuteSV**[28] and **SVIM**[29]: Tailored for long-read sequencing data.

These tools leverage discordant read pairs, split reads, and read depth signals to identify deletions, duplications, inversions, and translocations.

Filters for genotypic data

High-throughput technologies (see D3.2) are used to obtain genetic information, or **genotypes**, for each accession. The most common type of genetic variant analyzed is the **Single Nucleotide Polymorphism (SNP)**, a location in the genome where a single base of the DNA sequence varies among individuals. Millions of SNPs distributed across the entire genome are typically genotyped for each sample.

Genotyping providers typically apply initial **Quality Control (QC)** procedures that are specific to the technology used. For **Next-Generation Sequencing (NGS)**, standard protocols involve removing loci with low sequencing depth (i.e., supported by an insufficient number of reads) and loci with low PHRED-like quality scores (Q), where Q indicates the probability of an incorrect base call.

While necessary, these provider-level QC steps are insufficient on their own to prevent bias and spurious signals in genotype-trait association tests. Consequently, the investigator must implement a series of additional, more stringent QC measures. These supplementary procedures include filtering steps that are standard for any diversity, population structure and GWA analyses, as well as those adapted to the specific population structure of the study.

Basic filtering for GATK suite

These filters are exclusive of the GATK and must be applied only when using this pipeline

- **Base Quality Score Recalibration (BQSR)**

  With this method, implemented in GATK, systematic errors in sequencer-produced base quality scores are empirically characterized and corrected to

enhance variant calling accuracy. The process begins with the **BaseRecalibrator** module, which analyzes aligned sequencing reads (e.g., in BAM/CRAM format) alongside a database of known variant sites. By evaluating covariates such as read group, reported quality score, cycle (position in read), and sequence context (dinucleotide), the tool builds an empirical error model that reflects sequencing biases and machine-specific artifacts. Subsequently, **ApplyBQSR** uses the recalibration table generated by BaseRecalibrator to adjust base quality scores across the dataset, outputting a recalibrated BAM or CRAM file suitable for downstream analysis. For quality control, the **AnalyzeCovariates** tool can produce before-and-after plots to visualize the effects of recalibration and assess model performance.

BQSR is strongly recommended for workflows where base quality scores may be systematically biased, common in high-throughput sequencing platforms such as Illumina, PacBio, and others. In best-practice pipelines, it is considered an optional but highly advisable step, as it typically improves variant calling accuracy, especially in germline sequencing with standard or high coverage.

In the absence of known variant resources, such as experiments involving non-model organisms, BQSR may still be applied via a **bootstrapping strategy**, whereby an initial variant set is generated without BQSR, filtered for high confidence, and then used as a provisional known-sites resource for recalibration in subsequent rounds. It is noteworthy, however, that in certain contexts, such as when using variant callers like DeepVariant trained on raw data, or in low-coverage datasets, the benefits of BQSR may be minimal or context-dependent.

- **Variant Quality Score Recalibration (VQSR)**

  Variant Quality Score Recalibration (VQSR) is a refined, machine-learning–based method to filter variant calls by learning multi-dimensional annotation profiles from high confidence known variant sets and differentiating true variants from artifacts. Rather than applying rigid, unidimensional thresholds, VQSR constructs a **Gaussian mixture model (GMM)** to characterize the distribution of variant annotations, such as Quality by Depth (QD), Mapping Quality (MQ), Strand Odds Ratio (SOR), Fisher Strand (FS), ReadPosRankSum, among others, for both “true” and “false” variant classes. The tool **VariantRecalibrator** builds this model using overlapping sites between the callset and well-curated resources (e.g., HapMap, Omni, 1000 Genomes), and then assigns each variant a **VQSLOD score**, the log-odds ratio of being a true variant versus an artifact, based on its annotation profile. Users can then specify a **target sensitivity**, for instance aiming to retain 99% of known true positives, to set cutoffs that balance **sensitivity** and **specificity**, typically visualized via tranche plots. This enables nuanced filtering that captures

complex annotation interactions, akin to tracing contour lines around mountain-top clusters in a multi-dimensional annotation space.

VQSR is most effective for **large-scale germline variant callsets**, where ample variant data and high-quality training resources enable robust model fitting. It is the recommended best practice when performing **whole-exome or whole-genome analyses** with joint-called samples, as the substantial number of variants allows the Gaussian mixture model to distinguish true signals from noise reliably. However, for **small datasets**, such as single-sample or targeted panels with few variants, VQSR may fail to converge, and extensive manual tuning (e.g., reducing the number of Gaussians with --maxGaussians) might be required, though often hard-filtering remains the more practical choice. Crucially, VQSR requires **robust, well-validated truth resources**; in contexts lacking such resources, non-model organisms or novel experimental designs, hard-filtering or bootstrapped methods may still be preferable. When properly applicable, VQSR delivers a flexible, data-driven filtration framework that elegantly adapts to data characteristics and delivers maximized accuracy in variant classification.

GATK hard filter

The hard-filtering approach in GATK entails applying fixed numeric thresholds to one or more variant annotation metrics (e.g., QualByDepth, FisherStrand, StrandOddsRatio, RMSMappingQuality, MappingQualityRankSum, ReadPosRankSum) and rejecting any variant that fails to comply with these criteria. This method involves evaluating each annotation independently, resulting in variants being filtered out if even one metric exceeds (or falls below) the preset threshold; this unidimensional filtering can potentially exclude true positive variants or retain false positives to preserve others. GATK provides tools such as **VariantFiltration** to implement hard-filtering, which flags variants in the VCF by populating the FILTER field with appropriate labels instead of deleting them outright. Typical thresholds recommended by GATK, for instance, QD < 2.0, FS > 60, SOR > 3, MQRankSum < –12.5, ReadPosRankSum < –8.0, are often suggested as starting points, with adjustments made based on visualization of the annotation value distributions in the specific dataset.

Hard-filtering is particularly valuable when datasets lack sufficient size or high-quality known variant resources required for machine-learning-based strategies like VQSR (Variant Quality Score Recalibration). For example, targeted gene panels, exome subsets, non-model organisms, or small-scale sequencing experiments often yield too few variants to train reliable recalibration models, making hard-filtering the only practical alternative. In these contexts, custom thresholds, possibly informed through simulation or exploratory analysis, can enhance specificity without sacrificing sensitivity. Studies have recommended simulating variant datasets to derive optimized cutoffs, particularly for challenging variant types such as indels or in low-complexity regions, thereby tailoring

filtering parameters to the experimental design. While VQSR remains the recommended best practice for large, well-resourced datasets, hard-filtering serves as a robust fallback or manual refinement mechanism when VQSR is infeasible or when analyses require sample-level or cohort-specific customization.

```
#Hard filter SNPs#
GATK SelectVariants  -R ${REF} -V cohort.vcf.gz --select-type-
to-include SNP -O cohort_snps.vcf.gz
GATK VariantFiltration -R ${REF} -V cohort_snps.vcf.gz --
filter-expression "QD < 2.0 || QUAL <30 || FS > 60.0 || MQ <
40.0 || SOR > 4.0 || MQRankSum < -12.5 || ReadPosRankSum < -
8.0" --filter-name "SNP_FILTER" -O cohort_snps_tmp.vcf.gz
GATK SelectVariants -R ${REF} -V cohort_snps_tmp.vcf.gz -
exclude-filtered true -O cohort_snps_filtered.vcf.gz


#Hard filter Indels#
GATK SelectVariants -R ${REF} -V cohort.vcf.gz --select-type-
to-include INDEL -O cohort_indels.vcf.gz
GATK VariantFiltration  -R ${REF} -V cohort_indels.vcf.gz --
filter-expression "QD < 2.0 || QUAL <30 || FS > 200.0 ||
ReadPosRankSum < -20.0" --filter-name "INDEL_FILTER" -O
cohort_indels_tmp.vcf.gz
GATK SelectVariants -R ${REF} -V cohort_indels_tmp.vcf.gz -
exclude-filtered true -O cohort_indels_filtered.vcf.gz

#Merge SNPs and Indels vcf#
GATK GatherVcfsCloud --cloud-prefetch-buffer 0 -I
cohort_snps_filtered.vcf.gz -I cohort_indels_filtered.vcf.gz -
O cohort_snps_indels_filtered.tmp.vcf.gz
GATK SortVcf -I cohort_snps_indels_filtered.tmp.vcf.gz -O
cohort_snps_indels_filtered.final.vcf.gz
```

Population structure/pedigree determination

Understanding the population structure within a germplasm collection is essential to accurately interpret genetic diversity, assess redundancy, and control for confounding in downstream analyses such as GWAS or genomic prediction. Population structure refers to the presence of subgroups within the dataset that share a higher degree of relatedness due to geographic origin, domestication history, or selective breeding.

Population structure is typically inferred using several complementary approaches:

1. **Principal Component Analysis (PCA)**

PCA is a model-free, multivariate approach that summarizes genome-wide variation into orthogonal components (principal components, PCs). The first few PCs usually capture the main axes of genetic differentiation (e.g. between species, ecotypes, or breeding groups). For large genotypic datasets, the R package **SNPRelate**[30] offers an efficient implementation of PCA using the GDS (Genomic Data Structure) format, optimized for memory and computational speed.

```
Example code (R):
library(SNPRelate)

# Convert VCF to GDS format
vcf.fn <- "dataset_filtered.vcf.gz"
gds.fn <- "dataset_filtered.gds"
snpgdsVCF2GDS(vcf.fn, gds.fn, method="biallelic.only")

# Open GDS file
genofile <- snpgdsOpen(gds.fn)

# Perform LD pruning to reduce redundancy
set.seed(1000)
snpset <- snpgdsLDpruning(genofile, ld.threshold=0.2)
snpset.id <- unlist(snpset)

# Run PCA on pruned SNPs
pca <- snpgdsPCA(genofile, snp.id=snpset.id, num.thread=4)

# Extract eigenvectors and variance explained
pc.percent <- pca$varprop * 100
tab <- data.frame(sample.id = pca$sample.id,
                  EV1 = pca$eigenvect[,1],
                  EV2 = pca$eigenvect[,2],
                  EV3 = pca$eigenvect[,3],
                  stringsAsFactors = FALSE)

# Plot PCA
plot(tab$EV1, tab$EV2, col="dodgerblue", pch=19,
     xlab=paste0("PC1 (", round(pc.percent[1],1), "%)"),
     ylab=paste0("PC2 (", round(pc.percent[2],1), "%)"))
```

The resulting eigenvalues and eigenvectors can be visualized using R or Python to detect clustering patterns, which can then be included as covariates in GWAS to correct for stratification.

2. Model-based clustering methods aim to infer the ancestry composition of each individual by assuming a certain number of ancestral populations (*K*). Each individual's genome is represented as a mixture of contributions from these populations.

Traditionally, tools such as **STRUCTURE**[31] and **ADMIXTURE**[32] have been employed for this purpose. While **STRUCTURE** uses a Bayesian MCMC framework, **ADMIXTURE** relies on a maximum-likelihood approach, providing a faster estimation of ancestry proportions for large datasets.

In recent years, the **sNMF** algorithm implemented in the **R package LEA**[33] has become a preferred alternative for high-throughput genomic data, as it performs sparse non-negative matrix factorization (NMF) to estimate ancestry coefficients without the computational burden of MCMC sampling.

```
library(LEA)
# Convert VCF file to geno format (0, 1, 2 coding)
vcf2geno("dataset_filtered.vcf.gz", "dataset_filtered.geno")

# Run sNMF for a range of K values (number of ancestral
populations)
project <- snmf("dataset_filtered.geno", K = 1:8, entropy =
TRUE, repetitions = 3, project = "new")

# Cross-entropy criterion to identify the optimal K
plot(project, cex = 1.2, col = "dodgerblue4")

# Choose the best run for the optimal K (e.g., K=4)
best <- which.min(cross.entropy(project, K = 4))
qmatrix <- Q(project, K = 4, run = best)

# Plot ancestry coefficients
barplot(t(qmatrix),
        col = rainbow(4),
        border = NA,
        xlab = "Individuals",
        ylab = "Ancestry proportion")
```

The **cross-entropy criterion** identifies the optimal number of clusters (*K*) corresponding to the minimum entropy value. The resulting **Q-matrix** summarizes the proportion of ancestry components for each individual and can be visualized as stacked barplots.

This method provides accuracy comparable to ADMIXTURE but with substantially reduced runtime, making it particularly suitable for plant population genomics and for datasets derived from genebank accessions.

3. Discriminant Analysis of Principal Components (DAPC) Implemented in the R package *adegenet*[34], DAPC combines PCA for data reduction and discriminant analysis to maximize between-group variation. It is particularly effective in plant populations showing both clonal and sexual reproduction.

4. Phylogenetic and hierarchical clustering Tree-based methods (e.g. neighbor-joining, UPGMA) are often used to visualize relationships between accessions based on genetic distances (Nei, Identity-by-State, or IBS). These trees allow identification of misclassified accessions or close duplicates within genebank datasets.

Kinship

Kinship estimation quantifies the degree of relatedness between individuals, reflecting the proportion of alleles shared due to common ancestry. Accurate kinship estimation is essential to control confounding effects of relatedness in GWAS, genomic prediction, and pedigree reconstruction.

A kinship matrix (K) represents pairwise genomic relationships and can be computed using several methods:

- **VanRaden method**[35], implemented in GAPIT[36], TASSEL[37], and GEMMA[38]:

$$K = \frac{ZZ'}{2\sum p_i(1-p_i)}$$

where Z is the centered genotype matrix and p_i is the allele frequency at locus i.

A wide range of tools are available for estimating kinship or genomic relationship matrices, each optimized for different data sizes, formats, and analytical purposes:

Tool	Method	Input	Speed	Output format	Typical use
GAPIT3 (R)	VanRaden, EMMA	R data.frame	Fast	R matrix	GWAS, GBLUP
TASSEL	IBS, Centered	VCF / Hapmap	Fast	.txt / .csv	GWAS
GEMMA	GRM	PLINK .bed	Very fast	Binary matrix	Mixed models
GCTA	GRM (VanRaden)	PLINK .bed	Fast	.grm.bin	Heritability, genomic prediction
KING	IBD / IBS	PLINK .bed	Very fast	.kin0 table	Pedigree inference

Tool	Method	Input	Speed	Output format	Typical use
AGHmatrix (R)	VanRaden, Hybrid	R matrix	Fast	R object	Genomic selection
rrBLUP (R)	Additive G matrix	Numeric matrix	Fast	R matrix	BLUP / GBLUP models

```
Example PLINK command for IBS-based kinship:
plink2 --bfile INPUT --make-king-table --out kinship


GEMMA – efficient GRM calculation:
gemma -g genotypes.txt -p phenotype.txt -gk 1 -o kinship_gemma

GCTA – VanRaden-based GRM:

gcta64 --bfile genotypes --make-grm --out kinship_gcta

KING – pedigree and duplicate detection:
king -b genotypes.bed --kinship --prefix kinship_king
```

When explicit pedigree records are unavailable, as often in genebank accessions, kinship matrices can be used to infer putative parental or sibling relationships. High pairwise kinship coefficients (>0.45) typically indicate duplicates or clonally propagated material, whereas values around 0.25 suggest half-sibs or parent–offspring relationships.

In GWAS and genomic prediction, the kinship matrix is incorporated as a random effect to account for covariance among individuals. This reduces false positives caused by cryptic relatedness. In *GAPIT3*, the kinship matrix can be automatically computed and included in the mixed linear model (MLM) or compressed MLM (CMLM).

```
Example in R:
myGAPIT <- GAPIT(
  Y = myY,
  GD = myGD,
  GM = myGM,
  PCA.total = 4,
  kinship.algorithm = "VanRaden",
  model = "MLM"
)
```

Finally, heatmaps of kinship matrix allow visual inspection of genetic relatedness patterns across accessions. Clusters of high relatedness may correspond to specific breeding lines, geographic groups, or recent selection bottlenecks.

Gap analyses

Genebanks and curated germplasm collections play a crucial role in safeguarding the genetic richness of crop species. However, the degree to which existing collections represent the full spectrum of a species' genetic diversity remains uncertain. To address this, we developed an empirical framework to quantify the completeness of genetic collections with respect to nucleotide diversity (π). This parameter provides a direct, quantitative measure of average pairwise sequence differences among accessions. It is robust to missing data, scalable across genomic regions, and can be readily computed from standard SNP-level VCF files. As such, π represents a practical and biologically meaningful indicator for assessing genetic diversity completeness.

Demonstration activity: gap analysis in the tomato G2P-SOL core collection

The approach builds upon high-throughput genotyping data derived from whole-genome sequencing (WGS). Using tomato (*Solanum lycopersicum*) as a test case, we analyzed approximately 350 accessions sequenced at high coverage (~20×).

To assess diversity completeness, we generated random subsets of increasing size (ranging from 20 to 300 accessions) and calculated global nucleotide diversity (π) using the software *Pixy*[39]. For each sampling size, we computed the mean and standard deviation of π across 20 independent replicates.

Overall, the potential advantages of this approach are:

- **Assessment of collection completeness**, identifying whether a core set adequately represents total species diversity.
- **Comparison across genebanks**, to evaluate the relative comprehensiveness of their holdings.
- **Detection of redundancy or conservation gaps**, guiding future acquisition and curation efforts.

The resulting saturation curve should describe how genetic diversity increases with sample size. When the curve approaches a plateau, the collection can be considered to capture most of the species' nucleotide diversity. The preliminary results (**Fig. 1**) suggest i) that the π reaches a plateau at a relatively low number of accessions (60); ii) that the method needs a finer tuning of markers used (in terms of MAF filter, missing data, SNPs in coding or non-coding regions, etc.) to be carried out.

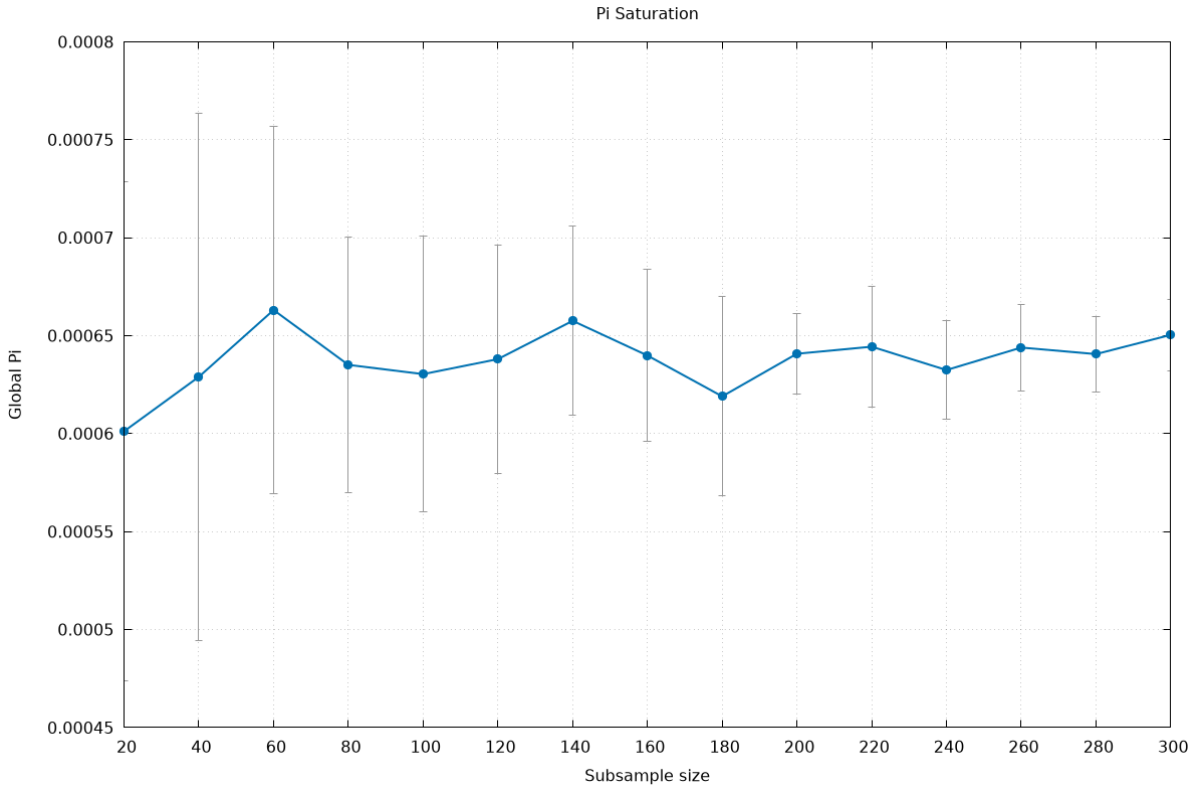


Figure 1. *Rarefaction-based assessment of nucleotide diversity (π) in tomato accessions. Each point represents the mean nucleotide diversity estimated from multiple random subsamplings of increasing collection size. The curve illustrates how genetic diversity accumulates with sample size, approaching a plateau that indicates the completeness of the collection in capturing species-wide variation.*

Genome wide association (GWA) study

Genome-Wide Association Study (GWAS) is a powerful, hypothesis-free approach used to identify associations between specific genetic variants and a particular trait or phenotype. This method systematically scans the entire genome of a large number of individuals to identify genetic markers that are statistically correlated with the trait of interest. While initially used in human genetics, the principles of GWAS are universally applicable across diverse species, including plants, animals, and microbes, to investigate the genetic basis of complex traits.

The heart of a GWAS is the statistical test for association. For each SNP, a test is performed to determine if there is a statistically significant difference in its allele frequencies between phenotypic groups. For a binary trait, this might be a chi-squared test, while for a quantitative trait, linear regression is commonly used. The model tests if the presence of a particular allele at a given SNP is predictive of the trait value. The fundamental model can be expressed as:

$$\text{Phenotype} \sim \beta \cdot \text{SNP} + \text{Covariates} + \epsilon$$

where β represents the effect size of the SNP and the model often includes covariates to control for confounding factors.

Correction for Confounding Factors: To avoid spurious associations, GWAS must account for population structure and cryptic relatedness. **Population structure** refers to systematic differences in allele frequencies between subpopulations, which can correlate with both genotype and phenotype, leading to false positives. Statistical methods, such as principal component analysis (PCA), are employed to correct these effects.

Significance Thresholds and Visualization: Since a GWAS involves performing millions of simultaneous statistical tests, a stringent significance threshold is required to correct for multiple testing. The standard threshold is typically set at **p-value < 5×10−8**, known as the genome-wide significance level. The results are commonly visualized using a **Manhattan plot**, which displays the negative logarithm of the p-value for each SNP across each chromosome. Significant associations appear as "skyscrapers" that rise above the significance threshold.

Several tools are currently available for Genome-Wide Association Analysis:

PLINK[40]: A high-performance toolkit enabling efficient management of large-scale genotypic datasets. It provides modules for stringent quality control, univariate association tests, population structure correction, and basic visualization. Its robustness and scalability make it a baseline tool for GWA workflows.

TASSEL[37]: Initially developed with a focus on plant genetics, TASSEL incorporates both single-locus and mixed-model association approaches. It allows the integration of genotypic, phenotypic, and environmental covariates, which is particularly valuable for agrigenomics and multi-environment trials.

GEMMA[38]: A software package implementing linear mixed models (LMMs) for both univariate and multivariate association analyses. GEMMA is optimized for variance component estimation, thereby efficiently accounting for relatedness and population stratification.

EMMAX[41]: A computationally efficient implementation of variance component models that leverages restricted maximum likelihood (REML) estimation. Its design allows for rapid analyses in very large datasets without compromising model accuracy.

Of particular relevance is GAPIT3 (Genome Association and Prediction Integrated Tool, version 3)[36], an R package providing a comprehensive and integrative framework for both association mapping and genomic prediction. GAPIT3 extends traditional models by incorporating state-of-the-art multi-locus and multi-trait approaches, thereby addressing the limitations of single-locus methods in terms of statistical power and false discovery control. Specifically, GAPIT3 integrates:

- General Linear Model (GLM) and Mixed Linear Model (MLM) as basic approaches.
- Compressed Mixed Linear Model (CMLM) and Enriched CMLM (ECMLM) to improve computational efficiency and statistical power in high-dimensional data.
- FarmCPU[42] (Fixed and random model Circulating Probability Unification), a multi-locus algorithm that iteratively incorporates significant markers as covariates, reducing confounding and enhancing detection power.
- BLINK[43] (Bayesian-information and Linkage-disequilibrium Iteratively Nested Keyway), a high-performance multi-locus model that leverages Bayesian information criteria and LD pruning for superior speed and precision in locus detection.

The combination of these models within GAPIT3 provides a flexible and extensible analytical environment, enabling researchers to tailor the methodological framework to the genetic architecture of the traits under study and to the specific characteristics of their datasets. Consequently, GAPIT3 has become a reference platform for advanced GWA studies across plant, animal and human genetics.

Additional filters on genotypic data for GWA

Prior to conducting a genome-wide association analysis, it is essential to apply stringent quality control filters to SNPs and SVs to minimize the risk of spurious associations and ensure robust statistical inference. Standard filters typically include the removal of multiallelic variants, as well as those with a **low call rate** (e.g., <20%; take into account your sequencing depth, since low sequencing depth requires a higher call rate filter), which may indicate poor genotyping quality, and exclusion of variants with a **minor allele frequency (MAF)** below a chosen threshold (commonly 0.01 or 0.05), as rare variants often lack sufficient power in GWAS. Furthermore, markers that deviate significantly from **Hardy–Weinberg equilibrium (HWE)** in the population (e.g., $p < 1e\text{-}6$) could be excluded, since such deviations may suggest genotyping errors or population stratification. However, numerous other factors can lead to deviations from HWE, particularly because its assumptions are rarely fulfilled in natural populations[44]. As a result, excluding loci that deviate from HWE can significantly influence population genetic inferences. Marked deviations from HWE heterozygosity expectations may also result from repetitive genomic elements[45]. Additional factors frequently responsible for departures from HWE in natural populations include overlapping generations, non-panmictic mating, deviations from diploidy, and very small effective population sizes. Moreover, genotype and single nucleotide polymorphism (SNP) calling methods can further contribute to such departures: genotype inference is often influenced by sequencing depth and by the mismatch thresholds used to call variants, both of which may reduce observed heterozygosity and consequently lead to deviations from HWE[46].

Generally, removing markers according to HWE deviation can be done in open pollinating crops, although in crops the HWE assumption of random mating is not respected in presence of stratification (i.e. structure). Furthermore, loci under selection are not in HWE, so you might remove them for GWAS. This leads to be careful with this filter. A possible solution is then to remove SNPs showing an excess of heterozygosity. It is very important to apply this filter based on the level of expected heterozygosity in the species you are working with.

Another critical aspect of variant-level quality control prior to GWAS concerns the **depth of sequencing (DP)**. Variants with **extremely low coverage** are prone to high genotype uncertainty and elevated error rates, while those with **excessively high coverage** may reflect mapping artifacts or repetitive regions. To mitigate these issues, SNPs are typically filtered by enforcing **minimum and maximum depth thresholds** (e.g., excluding genotypes with DP <10 to avoid false heterozygotes, and removing variants with mean DP exceeding 2/3 times the cohort average, which may indicate alignment artifacts).

Depth-based filtering can be performed both at the **per-genotype level** (removing low-confidence genotype calls within individuals) and at the **per-variant level** (excluding SNPs systematically under- or over-covered across samples). This ensures that retained variants are supported by sufficient, but not inflated, read evidence, thereby improving downstream genotype accuracy and association testing power.

In addition to SNP-level filtering, rigorous **sample-level quality control** is required to ensure the integrity of the dataset prior to genome-wide association studies. A common first step is the exclusion of individuals with an **excessive missing genotype rate** (e.g., >5%), which may indicate poor DNA quality or technical artifacts. Further, **heterozygosity outliers** are excluded, as elevated or reduced genome-wide heterozygosity may signal contamination, inbreeding, or technical errors.

Finally, since GWAS cannot deal with missing values, missing marker values need to be imputed before we can regress phenotypes on genotypes. Various options are available, including the mean imputation or using Beagle[47] (based on Hidden Markov Model (HMM)). The former replaces missing marker calls by the mean value of the marker across the population. It is the fastest and simplest way to impute missing markers. An adequate method if the fraction of missing markers is very low and the marker density is high (i.e., genomic regions are represented by many markers).

Beagle uses a localized haplotype clustering imputation algorithm. It makes use of a Hidden Markov Model (HMM) to find the most likely haplotype pair given the genotype data for that individual and the haplotype frequency in the population. Beagle is relatively user-friendly, accurate under default settings, well-supported, and widely used. It takes a vcf (also compressed).

```
#example code for additional filters on SNPs and indels vcf
for GWA purpose#
#left align variants and remove SNPs near indels
bcftools norm -f "genome.fa" -O z -o OUTPUT_norm.vcf.gz --
threads 50 INPUT.vcf.gz
bcftools view OUTPUT_norm.vcf.gz | bcftools filter -e 'AC==0
|| AC==AN' --SnpGap 5 -O z -o OUTPUT_norm_no_5bpSNPS.vcf.gz --
threads 50
# keep only biallelic SNPs and remove indels
bcftools view -O z -o OUTPUT_norm_no_5bpSNPS_biallelic.vcf.gz
-c 1 -m2 -M2 --types snps --threads 50
OUTPUT_norm_no_5bpSNPS.vcf.gz
#calculate depth of coverage on a per-basis level
vcftools --gzvcf OUTPUT_norm_no_5bpSNPS_biallelic.vcf.gz --
site-mean-depth --out mean_depth
```

```
#In R plot and get statistics of the depth at each site
library(tidyverse)
var_depth <- read_delim("mean_depth.ldepth.mean", delim =
"\t", col_names = c("chr", "pos", "mean_depth", "var_depth"),
skip = 1)
summary(var_depth$mean_depth)
png(filename="DP.png", width =300, height=300, units="mm", res
= 300)
ggplot(var_depth, aes(mean_depth)) + geom_density(fill =
"dodgerblue1", colour = "black", alpha = 0.3)+
  theme_light()+ xlim(0, 50)
+geom_vline(xintercept=mean(var_depth$mean_depth),color="red")
dev.off()
```

```
#Filter based on the depth of coverage and also based GQ
parameter
bcftools filter  -e 'MEAN(FMT/DP)<12'
OUTPUT_norm_no_5bpSNPS_biallelic.vcf.gz | bcftools filter -e
'MEAN(FMT/DP)>30' | bcftools filter  -S . -e 'FMT/GQ<20'  |
bcftools filter -e 'AC==0 || AC==AN' -O z  -o
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ.vcf.gz
#calculate missingness at SNP level
plink2  --vcf
```

```
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ.vcf.gz  --allow-
extra-chr -missing
```

```
#In R plot and get statistics of the missing data at SNP level
snpmiss<-read.table(file="plink2.vmiss",
header=TRUE,comment.char = "")####This fle contains missing at
SNP level, to USE
summary(snpmiss$F_MISS)
png(filename="missing.png", width =300, height=300,
units="mm", res = 300)
ggplot(snpmiss, aes(F_MISS)) + geom_density(fill =
"dodgerblue1", colour = "black", alpha = 0.3)+xlim(0,0.3)
dev.off()
```

```
#filter at 5% of missing data at SNP level
bcftools filter
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ.vcf.gz  -e
'N_MISSING>11' --threads 40 | bcftools filter -e 'AC==0 ||
AC==AN' -O z --threads 50 -o
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5.vcf.gz

#filter accessions with high percentage of missing data

plink2  --vcf
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5.vcf.gz --
allow-extra-chr --missing
```

```
#In R plot and get statistics of the missing data at
individual level
library(tidyverse)
indmiss<-read.table(file="plink2.smiss",
header=TRUE,comment.char = "")
ggplot(indmiss, aes(F_MISS)) + geom_density(fill =
"dodgerblue1", colour = "black", alpha = 0.3)+xlim(0,0.6)
```

```
#As an example, keep accessions with no more than 20% of
missing data
plink2  --vcf
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5.vcf.gz --
allow-extra-chr  --set-missing-var-ids @:# --mind 0.2  --make-
bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20
```

```
#Filter out markers with a MAF < 0.05, exclude chromosome 0
(not very useful for GWA) and save output as plink bed file
plink2  --bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5 --allow-
extra-chr --not-chr 0 -maf 0.05 -make-bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5

#Filter highly heterozygous SNPs based on mean heterozygosity
plink2  --bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 --geno-counts
```

```
In R plot and get statistics of the heterozygosity per site
het_site <- read.table("plink2.gcount", head=TRUE, sep="\t",
comment.char = "")
het_site$HET_RATE =
het_site$"HET_REF_ALT_CTS"/(het_site$"HOM_REF_CT" +
het_site$"TWO_ALT_GENO_CTS"+het_site$"HET_REF_ALT_CTS")
ggplot(het_site, aes(HET_RATE))+geom_density()+theme_bw()
mean(het_site$HET_RATE)
sd(het_site$HET_RATE)

#####get sites with a heterozygosity higher than the mean plus
2 SD
het_fail_site = subset(het_site, (het_site$HET_RATE >
mean(het_site$HET_RATE)+2*sd(het_site$HET_RATE)));
write.table(het_fail_site, "fail-het_site-qc.txt",
row.names=FALSE)
```

```
# In bash:
sed 's/"// g' fail-het_site-qc.txt | awk '{print$1, $2}'>
het_fail_site.txt
plink2 -bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 --allow-extra-chr --exclude  het_fail_site.txt --
make-bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 _het_site
```

```
#check the heterozygosity of your accessions. Keep in mind the
species you are working with. Calculate heterozygosity per
```

```
individual by pruning SNPs matrix. Better to use independent
SNPs
plink2 -bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 _het_site  --allow-extra-chr --indep-pairwise 50 5
0.1
#extract pruned SNPs and calculate heterozygosity on
individual bases
plink2 -bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 --extract plink2.prune.in --het --make-bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_pruned
```

```
In R, plot and get statistics of the heterozygosity per
individual
het -<
read.table("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSIN
G5_missing_ind20_MAF5_pruned.het", head=TRUE, comment.char =""
)
het$HET_RATE = (het$"OBS_CT" - het$"O.HOM.")/het$"OBS_CT"
ggplot(het, aes(HET_RATE))+geom_density()+theme_bw()
mean(het$HET_RATE)
sd(het$HET_RATE)

Remove individuals having, as example a mean + 3 SD het
(contamination?) and -3SD (inbreeding?)
het_fail = subset(het, (het$HET_RATE < mean(het$HET_RATE)-
3*sd(het$HET_RATE)) | (het$HET_RATE >
mean(het$HET_RATE)+3*sd(het$HET_RATE)));
het_fail$HET_DST = (het_fail$HET_RATE-
mean(het$HET_RATE))/sd(het$HET_RATE)##add deviation from the
mean
write.table(het_fail, "fail-het-qc.txt", row.names=FALSE)
```

```
#In bash:
sed 's/"// g' fail-het-qc.txt | awk '{print$1, $2}'>
het_fail_ind.txt
plink2 --bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5 --remove het_fail_ind.txt --make-bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
```

```
nd20_MAF5_NO_het_ind
# Save as vcf al well
plink2 -bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind --allow-extra-chr  --export vcf bgz --out
VCF_for_LD
```

An important step is to calculate linkage disequilibrium (LD) decay. It shows the relationship between R^2 or Dprime on the y-axis and the distance between marker pairs on the x-axis to understand the pattern of LD, which will be used later for QTL identification. One can use PopLDdecay in bash (https://github.com/BGI-shenzhen/PopLDdecay) with a -MaxDist: 1 million bp.

GWA analysis using GAPIT3

Molecular markers

```
#SNPs Imputation with Beagle
plink2 --bfile
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind --allow-extra-chr --export vcf bgz --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_BEAGLE
#Imputation
java -jar -Xmx30G beagle gt=
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_BEAGLE.vcf.gz out=
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED

# Filter again for maf after imputation and save as
plink2 --vcf
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED.vcf.gz  --allow-extra-chr --
export vcf bgz --maf 0.05 --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED _MAF
# Write out in dosage format, i.e. 012 and setup files for
GAPIT
plink2 --vcf
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF.vcf.gz --export A --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF
```

```
# Write out in bed format
plink2 --vcf
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF.vcf.gz --make-bed --out
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF
```

```
#In R read in genotype file which we imputed before using
Beagle in numeric format####
test=fread("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSIN
G5_missing_ind20_MAF5_NO_het_ind_IMPUTED_MAF.raw", head=T)
test1=test[, -c(1,3:6)] #remove unwanted columns
fwrite(test1, file="
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF.raw", quote=FALSE, sep="\t",
nThread = 8, dec=".")
###Prepare SNP information for GAPIT with ID SNPs
myGD=read.table("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_M
ISSING5_missing_ind20_MAF5_NO_het_ind_IMPUTED_MAF.bim")
myGD_final=myGD%>%select(V2,V1,V4)%>%rename(Name=V2,Chromosome
=V1,Position=V4 )
write.table(myGD_final, file="
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF_snp_information",
sep="\t",row.names = F, quote=F)
```

Phenotypic data Processing for GWA

The reliability and interpretability of Genome-Wide Association (GWA) results strongly depend on the quality and statistical properties of the phenotypic data used as input. In plant genetics, phenotypic datasets are often derived from multi-environment trials, replicated experimental designs, and high-throughput phenotyping platforms. This complexity requires careful data preprocessing and standardization before association analysis.

Phenotypic data must first undergo rigorous quality control procedures. Outlier detection is performed to identify measurements that deviate markedly from the distribution of the trait within a given environment or replicate, which may result from measurement errors, environmental stress events, or recording inconsistencies. Outliers can be removed or, when appropriate, replaced using best linear unbiased estimates (BLUEs) or predictors (BLUPs) derived from mixed models.

Many GWA methods assume that phenotypic values approximate a normal distribution. Traits exhibiting skewness, kurtosis, or bounded distributions (e.g., percentage traits,

counts, or scores) may violate this assumption. In such cases, statistical transformations (e.g., logarithmic, square-root, Box-Cox, or rank-based inverse normal transformation) are applied to improve normality and stabilize variance. The choice of transformation depends on the underlying biological meaning of the trait and must preserve interpretability of results.

Plant phenotypes are typically collected across multiple blocks, environments, and years. To account for these sources of variation, linear mixed models are used to partition phenotypic variance into genetic and environmental components. This adjustment improves trait heritability estimates and reduces noise from uncontrolled environmental heterogeneity. The resulting BLUEs or BLUPs provide standardized trait values that can be directly used in downstream GWA analyses.

```
#R code for normalization and BLUPs for a single trait with
multiple environments using inti and BestNormalize[48] packages
feno=read.delim(file = "Pheno_3_ENVs_outliers.txt", header=T)
feno$reps<-as.factor(feno$reps)
feno$Env<-as.factor(feno$Env)
feno_ENV=feno %>% mutate_at(c(2), as.numeric)
feno_ENV$names<-rownames(feno_ENV)

ggplot(feno_ENV, aes(x=Trait4,color=Env)) +
geom_density()+theme_bw()
ggplot(feno_ENV, aes(y=Trait4,color=Env)) +
geom_boxplot()+theme_bw()
#Check and remove outliers
rmout <- outliers_remove(data = feno_ENV, trait ="Trait4" ,
model = "1 + (1|reps:Env) + (1|ID)+(1|Env) +  (1|ID:Env)")
rmout$outliers
outlier_4=rmout$outliers
outlier_4$names <- rownames(outlier_4)
feno4_no_outliers=feno_ENV%>%anti_join(outlier_4,
by="names")#remove outliers based on rownames
#Normalization after outliers removal
(BNobject <- bestNormalize(feno4_no_outliers$Trait4))
orderNorm <- orderNorm(feno4_no_outliers$Trait4)
orderNorm
p <- predict(orderNorm)
x2 <- predict(orderNorm, newdata = p, inverse = TRUE)
pdf=data.frame(p)
pdf_renamed=rename(pdf, trait4_norma =p)
feno4_no_outliers_norma <- cbind(feno4_no_outliers,
```

```
pdf_renamed)
hr <- H2cal(data = feno4_no_outliers_norma, trait =
"trait4_norma", gen.name = "ID", rep.n = 3,env.n=3,env.name =
"Env", fixed.model = "0 + (1|reps) + ID+(1|Env)   (1|ID:Env)"
, random.model = "1 +  (1|ID)+(1|Env/reps) +  (1|ID:Env)" ,
emmeans = F, plot_diag = TRUE , outliers.rm = FALSE)
hr$model %>% summary()
hr$tabsmr
BLUPs=hr$BLUPs
write.table(BLUPs,file="Trait4_MET.txt", row.names = F,
quote=F, sep="\t")
```

GWA with GAPIT3 and Blink

To mitigate confounding due to population structure or cryptic relatedness, individuals with unexpected **kinship coefficients** (e.g., duplicate or closely related samples) are identified and pruned, typically using identity-by-descent (IBD) analysis. Additionally, **principal component analysis (PCA)** is routinely applied to detect population outliers that deviate substantially from the main study cohort, as these may inflate false-positive associations if left unaddressed.

GAPIT3 calculates both kinship matrix and PCA according to the model used. For a detailed description of the procedures see “Kinship/population structure” section. As a rule of thumb, one can prepare a PCA on SNPs data and check the appropriate numbers of components to use in the GAPIT analysis.

```
#R code for GWA analysis using the raw SNPs matrix previously
obtained as well as the BLUPs for a trait measured in multiple
environments
# Read in genotype file which we imputed before using Beagle
in numeric format
test=fread("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSIN
G5_missing_ind20_MAF5_NO_het_ind_IMPUTED_MAF.raw", head=T)
test1=test[, -c(1,3:6)] #remove unwanted columns
fwrite(test1, file="
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF_OK.raw", quote=FALSE,
sep="\t", nThread = 8, dec=".")
###Prepare SNP information for GAPIT with ID SNPs
myGD=read.table("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_M
```

```
ISSING5_missing_ind20_MAF5_NO_het_ind_IMPUTED_MAF_OK.bim")
myGD_final=myGD%>%select(V2,V1,V4)%>%rename(Name=V2,Chromosome
=V1,Position=V4 )
write.table(myGD_final, file="
OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_missing_i
nd20_MAF5_NO_het_ind_IMPUTED_MAF_OK_snp_information",
sep="\t",row.names = F, quote=F)
myY=read.delim("Trait4_MET.txt", head=T)
myGD <-
fread("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSING5_mi
ssing_ind20_MAF5_NO_het_ind_IMPUTED_MAF_OK.raw", head = T)
myGM<-
read.table("OUTPUT_norm_no_5bpSNPS_biallelic_DP12_30_GQ_MISSIN
G5_missing_ind20_MAF5_NO_het_ind_IMPUTED_MAF_OK_snp_informatio
n", head=T)
#myGM$Chromosome<-as.numeric(myGM$Chromosome)
#Running GAPIT3 with Blink with 0 PCA components as covariates
myGAPIT <- GAPIT( Y=myY,  GD=myGD, GM=myGM,
SNP.fraction=0.2,  SNP.test = T,  PCA.total=0,  PCA.3d=FALSE,
Inter.Plot=TRUE,   Multiple_analysis=FALSE,   model=c("Blink")
,   file.output = TRUE)
```

Statistical Procedures for Determining Significance in GWA

Genome-wide association (GWA) studies typically test a very large number of genetic markers (from several hundred thousand to millions) for association with phenotypic traits. This massive multiple-testing burden makes it essential to apply appropriate statistical procedures to distinguish true associations from spurious results. Several complementary approaches are employed to determine statistical significance, each with advantages and limitations.

1. Bonferroni and Šidák Corrections (Family-wise Error Rate Control)

The Bonferroni[49] method sets the genome-wide significance threshold by dividing the nominal α (e.g. 0.05) by the number of independent tests.

- **Example**: with 1,000,000 SNPs, the Bonferroni threshold is p < 0.05 / 1,000,000 = **5 × 10^−8**.
- This threshold has become a de facto standard in human GWAS, corresponding approximately to the number of independent common variants in European populations.

- The Šidák[50] correction ($\alpha_i = 1 - (1 - \alpha)^{(1/m)}$) is slightly less conservative, but both assume independence among tests, which is not fully satisfied due to linkage disequilibrium (LD).

2. False Discovery Rate (FDR) Control

The False Discovery Rate (FDR) approach, most commonly via the Benjamini-Hochberg procedure[51], controls the expected proportion of false positives among the declared significant associations.

- **Example**: setting FDR q = 0.05 means that, on average, 5% of the reported significant SNPs are expected to be false positives.
- This approach is more powerful than Bonferroni, particularly for traits controlled by many loci with small effects, as often observed in plants.
- FDR thresholds are data-dependent: for instance, in a maize GWAS with ~500,000 SNPs, the cut-off may fall around $p < 10^{-5}$ depending on the observed distribution of test statistics.

3. Permutation-Based Thresholds

Permutation testing provides empirical thresholds by randomly permuting phenotypes relative to genotypes, recalculating test statistics, and estimating the null distribution[52].

- **Example**: in a plant GWAS with 100,000 SNPs and 1,000 permutations, the 5% empirical genome-wide significance threshold might correspond to the **minimum p-value observed across permutations**, often lying around $p \approx 10^{-5}$ rather than the much more stringent Bonferroni threshold of 5×10^{-7}.
- This method accounts for marker correlation due to LD and can be less conservative than Bonferroni, but it is computationally intensive.

4. Effective Number of Independent Tests

Because of LD, the number of independent tests is smaller than the raw number of SNPs. Estimating the **effective number of tests (M_{eff})** provides a less stringent, but still rigorous, threshold.

- **Example**: in rice GWAS with 400,000 SNPs, the effective number of independent markers may be ~100,000. The Bonferroni-adjusted threshold would then be 0.05 / 100,000 = **5×10^{-7}**, less conservative than the naïve 0.05 / 400,000 = 1.25×10^{-7}.

5. Hybrid and Weighted Procedures

More advanced procedures integrate prior knowledge (e.g. genomic annotation, minor allele frequency, functional relevance) to adaptively weight tests.

- **Example**: SNPs in coding or regulatory regions may be given higher weight, leading to a less stringent threshold (e.g. p < 10^−6) for these variants compared to intergenic SNPs.
- Such methods can substantially increase the power to detect biologically relevant loci while controlling type I error rates.

6. q-value approach for FDR control in GWA

The **q-value** procedure[53] provides an extension of the False Discovery Rate (FDR) framework. While the Benjamini–Hochberg[51] (BH) method controls FDR at a predefined threshold *q* (e.g. 0.05), the q-value method estimates, for each test, the minimum FDR at which the test may be called significant.

- **Interpretation**: the q-value of a SNP can be interpreted as the expected proportion of false positives among all associations at least as significant as that SNP.
- **Practical use**: instead of reporting a fixed cut-off (e.g. p < 1 × 10^−5), researchers can report all SNPs with q-value < 0.05, ensuring that, on average, no more than 5% of these associations are false discoveries.
- **Advantages**:
    - Provides SNP-specific FDR estimates rather than a single global threshold.
    - More powerful than conservative FWER methods (Bonferroni, Šidák).
    - Particularly suitable for **polygenic traits in plants**, where numerous small-effect loci may be detected.
- **Limitations**:
    - Requires accurate estimation of the proportion of true null hypotheses (π0), which may be challenging in some datasets.
    - Results can be influenced by p-value distributional properties, especially in structured populations or under strong LD.

```
#R code for threshold determination using gapit output
results_t4=fread(file =
"GAPIT.Blink.trait4_norma.GWAS.Results.csv", header=T,
sep=",")
####Bonferroni#####
number_comparison=nrow(results_t1)
bonferroni_threshold=0.05/number_comparison
bonferroni_threshold_LOG=-log10(bonferroni_threshold)
```

```
results_t4_bonferroni<-
results_t4%>%filter(P.value<=bonferroni_threshold)
###qvalue
p_2 <- results_t4$P.value
alpha <- 0.05
qobj <- qvalue(p_2, fdr.level = alpha)
summary(qobj)####see how many significant at p 0.05
qobj$significant
# find the significant pvalues at FDR level equal to alpha
sp <- sort(qobj$pvalues)
numbs <- sum(qobj$significant)
s <- sp[numbs]
ns <- sp[numbs+1]
thr_nolog <- mean(c(s,ns))
thr <- mean(c(-log10(s),-log10(ns)))
results_t4_FDR<-results_t4%>%filter(P.value<=thr_nolog)
```

Demonstration activity: GWA in the pepper G2P-SOL core collection

Core collection resequencing and SNP calling

The G2P-SOL core collection of *Capsicum* spp., consisting of 423 accessions representing the genetic variability of a panel of 10,083 accessions, contains 393 *C. annuum* accessions and 32 accessions from other cultivated species; these include 16 *C. chinense*, 4 *C. frutescens*, 7 *C. baccatum*, 1 *C. chacoense*, 1 *C. praetermissum*, and one unclassified accession. This collection underwent resequencing using the MGI platform at 20X coverage depth. Subsequently, the raw reads obtained from the sequencing step were aligned to the *C. annuum cultivar* Zhangshugang[54] pepper reference genome using the BWA-MEM tool[5]. Following this, SNP and small indel calling was conducted using GATK[14], yielding more than 300 million unfiltered variants. The raw variants were then filtered in accordance with the GATK Best Practices pipeline and bcftools, harvesting 31.328.757 high quality SNPs[14,17].

Analysis of the post-filtering marker set revealed a mean minor allele frequency (MAF) of 20,7% and an average density of 10,36 markers/Kb. At the marker level, the dataset exhibited 2% missing data and a 2,1% rate of heterozygosity, the accessions displayed on average 2% missing data and a heterozygosity rate of 7,5%.

Field trials and phenotyping

Six independent field trials were conducted across different years and locations. One trial was performed by ARO in Israel (2020), one by BATEM in Türkiye (2020–2021), one by CREA in Italy (2019), and two by INRAE in France (2019–2020). WorldVeg conducted an additional trial in Taiwan (2020–2021). Seedlings were transplanted 5–7 weeks after

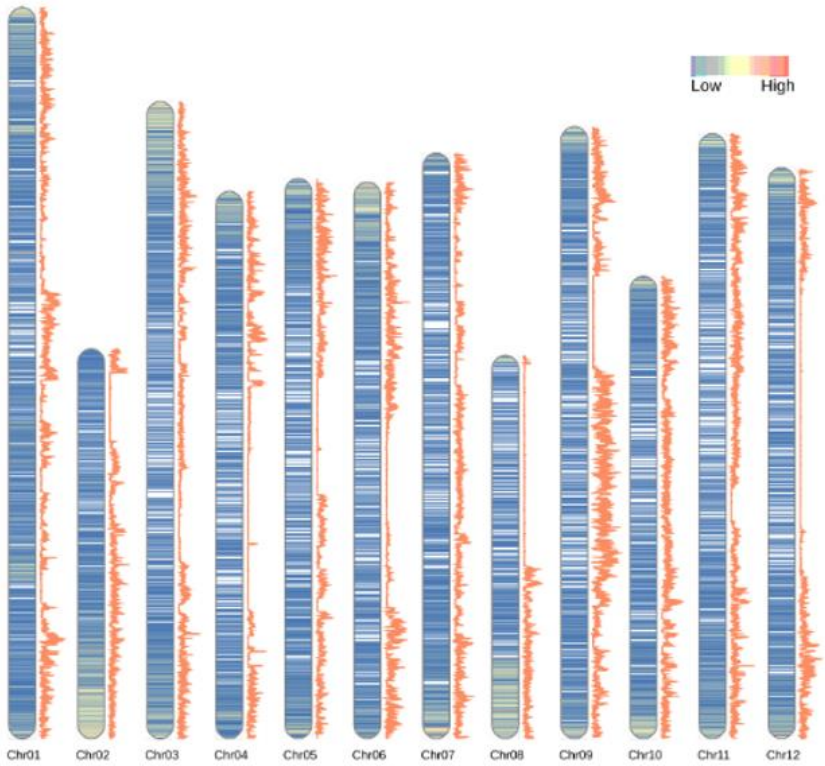
sowing in a randomized complete block design with two to three blocks and two to four plants per accession per block. Across trials, 23 agronomic traits based on pepper descriptors from the International Plant Genetic Resources Institute (IPGRI) were assessed. Of the 18 quantitative traits, 15 were recorded as continuous variables (e.g., axis length, Brix, fruit dimensions, plant height, pericarp thickness, total fruit weight) and three as discrete (flowering time, locule number, total fruit number). Among the five qualitative traits, three were ordinal (fruit fasciation, fruit load, immature fruit color) and two were binary (fruit pungency, predominant oblate fruit shape). The resulting field-trial data were then used to estimate best linear unbiased predictors (BLUPs), which served as input for the GWAS, using the R package inti[55].

C. annuum accessions and markers selected for GWAS

The initial core collection, encompassing 423 accessions representative of the *Capsicum* genus, underwent a series of refinement steps. First, the collection was narrowed to 393 accessions specifically belonging to *C. annuum*. Subsequently, accessions exhibiting excessive heterozygosity, defined as those deviating by more than two standard deviations from the mean individual heterozygosity, were excluded. Further filtration removed accessions with over 20% missing markers. Additionally, accessions displaying multiple phenotypes in field trials or lacking phenotypic data were eliminated. This comprehensive selection process yielded a final set of 362 accessions, which will serve as the foundation for genome-wide association studies (GWAS).

The final marker set composed of 17.404.615 SNPs, exhibited a mean density of 5,75 markers per Kb, but not homogeneously distributed across the different chromosomes (**Fig. 2**); with an average minor allele frequency (MAF) of 21%. Heterozygosity rates were observed at 1.36%, while missing data at the marker level accounted for 1.3%, the accessions displayed on average 1.3% missing data and a heterozygosity rate of 3.4%.

Figure 2 *Marker density along the 12 chromosomes. The 12 pepper chromosomes are depicted as vertical bars. In each chromosome, the horizontal bars represent the gene density, while the red lines represent the SNP density.*



GWAS on 23 different traits

Genome-wide association studies (GWAS) were performed on both single-environment and multi-environment BLUPs using the R package Genomic Association and Prediction Integrated Tool (GAPIT, Version 3)[36]. The BLINK model[43] was applied to each trait, and results were corrected for relatedness using a kinship matrix calculated with the VanRaden formula, as well as for population structure by including the first four principal components estimated by GAPIT3. Circular Manhattan plots were generated with CMplot[56], applying a Bonferroni threshold to control the false discovery rate (**Fig. 3**). Quantitative trait nucleotides (QTNs) detected across up to six environments were then merged, considering the previously estimated linkage disequilibrium (LD) decay for each chromosome (0.3–0.4 Mb).

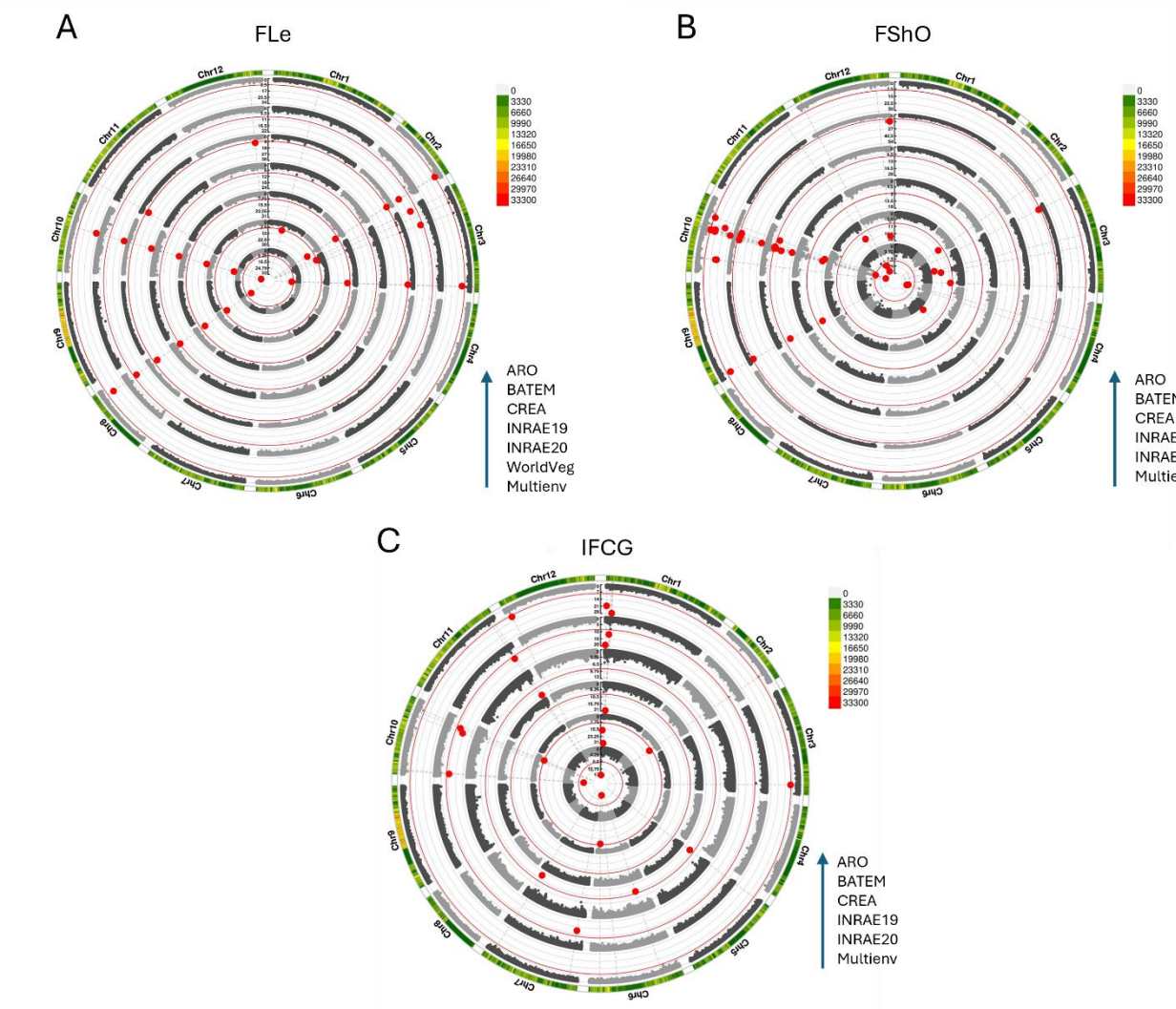


Figure 3 *Examples of highly robust QTLs identified in this study: (A) FLe (fruit length), with robust QTLs on chromosomes 2, 8, and 10; (B) FShO (fruit predominant shape oblate), with robust QTLs on chromosomes 9 and 10; and (C) IFCG (external immature fruit color green), with a robust QTL on chromosome 1. In the circular Manhattan plots, each concentric ring represents a different environment, and in every ring the red line indicates the Bonferroni threshold.*

From the GWAS of the 23 traits, 207 significant markers were identified and mapped to 112 genomic regions. Among these, the most promising regions, based on gene annotations from the Zhangshugang reference genome[54], will be further investigated to identify candidate genes and to elucidate the complex mechanisms underlying these traits.

QTL analyses

Quantitative Trait Locus (QTL) analysis in biparental populations represents a classical yet powerful approach for dissecting the genetic architecture of complex traits. Unlike genome-wide association studies (GWAS), which exploit natural diversity, QTL mapping relies on controlled crosses between two parental lines differing for one or more target traits. The resulting segregating population allows the detection of genomic regions associated with phenotypic variation under a defined genetic background.

Common biparental populations used in plants include F_2, backcross (BC), recombinant inbred lines (RILs), doubled haploids (DH), and near-isogenic lines (NILs). Each design offers distinct advantages:

- **F_2 and BC**: quick to develop and suitable for detecting major-effect loci.
- **RILs and DH**: provide stable, immortal populations ideal for multi-environment evaluation and fine mapping.
- **NILs**: allow validation of individual QTLs in uniform backgrounds.

High-density genetic linkage maps are constructed using molecular markers (SNPs, SSRs, DArTseq). Recombination fractions between markers are converted into genetic distances (in centiMorgans, cM) using mapping functions such as Kosambi or Haldane. Popular software for linkage map construction include:

- **JoinMap**[57], **R/qtl**[58], and **MSTMap**[59].

```
Example R code for linkage map construction (R/qtl):
library(qtl)
cross <- read.cross(format="csv", file="cross_data.csv",
genotypes=c("AA","AB","BB"))
cross <- est.rf(cross)
cross <- est.map(cross)
plotMap(cross)
```

QTL mapping models the relationship between marker genotype and trait phenotype across the genome.

Several statistical approaches are widely used:

- **Single-marker analysis (SMA)**: tests each marker individually, simple but limited in power.
- **Simple Interval Mapping (SIM)**: estimates QTL position between adjacent markers using likelihood ratios or LOD scores.
- **Composite Interval Mapping (CIM)** and **Multiple QTL Mapping (MQM)**: improve resolution by incorporating background markers as cofactors.
- **Mixed models (MLM/QTL-seq)**: integrate kinship or population structure corrections, applicable to complex pedigrees or bulk-segregant sequencing data.

```
Example CIM command in R/qtl:
cim_results <- cim(cross, n.marcovar=5, window=10)
summary(cim_results)
plot(cim_results)
```

Permutation tests (usually 1,000–10,000 iterations) are used to define genome-wide LOD thresholds at a given significance level (typically $\alpha = 0.05$). Confidence intervals for QTL positions are estimated using 1- or 2-LOD drop methods, corresponding approximately to 95% confidence regions.

Detected QTLs are characterized by additive, dominant, and epistatic effects, as well as by the percentage of phenotypic variance explained (PVE). Co-localization with annotated genes or functional variants from genome assemblies enables candidate gene identification and biological interpretation.

Conclusion

Deliverable **D3.5** successfully demonstrates the implementation and integration of bioinformatic methods and analytical pipelines essential for the characterization and exploitation of plant genetic resources within the PRO-GRACE framework. The activities carried out confirm the feasibility and robustness of standardized workflows for variant discovery, population structure and kinship assessment, gap analysis, and genome-wide association and QTL mapping.

Through the integration of established software tools and reproducible pipelines, the deliverable provides a practical framework to harmonize data processing and analysis across different crops, genebanks, and research institutions. The developed methods ensure data quality, interoperability, and comparability, supporting the long-term goal of building a cohesive European plant genetic resource community.

The demonstration activities conducted on tomato and pepper core collections validated the pipelines' performance on real datasets. They provided results for genetic diversity assessment, identification of population structure and kinship patterns, and discovery of loci associated with key agronomic traits. These examples illustrate how the proposed computational approaches can support breeding programs, germplasm management, and conservation strategies.

D3.5 also shows that given the complexity of bioinformatic analyses and the computational resources required, it is neither practical nor efficient for individual European genebanks to perform such tasks independently. Centralizing these activities within two or more dedicated hubs of the GRACE infrastructure in Europe ensures the availability of specialized expertise, standardized environments, and high-performance computing capacity. This distributed-yet-coordinated model allows for scalability and redundancy, while maintaining harmonized analytical standards across sites. By concentrating bioinformatic efforts in a limited number of well-equipped centers, the infrastructure can guarantee data quality, reproducibility, and methodological

consistency, ultimately enabling genebanks to fully benefit from advanced genomic analyses without the need for local technical infrastructures.

Deviations

None.

References

1. Andrews, S. FastQC: a quality control tool for high throughput sequence data. (2010).
2. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014).
3. Chen, S., Zhou, Y., Chen, Y. & Gu, J. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* **34**, i884–i890 (2018).
4. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnetjournal Vol 17 No 1 Gener. Seq. Data Anal.* https://doi.org/10.14806/ej.17.1.200 (2011) doi:10.14806/ej.17.1.200.
5. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. http://arxiv.org/abs/1303.3997 (2013).
6. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).
7. Vasimuddin, Md., Misra, S., Li, H. & Aluru, S. Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* 314–324 (2019). doi:10.1109/IPDPS.2019.00041.
8. Jung, Y. & Han, D. BWA-MEME: BWA-MEM emulated with a machine learning approach. *Bioinformatics* **38**, 2404–2413 (2022).
9. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
10. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods* **12**, 357–360 (2015).

11. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).

12. Picard Tools - By Broad Institute. https://broadinstitute.github.io/picard/.

13. Tarasov, A., Vilella, A. J., Cuppen, E., Nijman, I. J. & Prins, P. Sambamba: fast processing of NGS alignment formats. *Bioinformatics* **31**, 2032–2034 (2015).

14. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491–498 (2011).

15. Poplin, R. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983–987 (2018).

16. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. Preprint at https://doi.org/10.48550/arXiv.1207.3907 (2012).

17. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *GigaScience* **10**, giab008 (2021).

18. Kim, S. *et al.* Strelka2: fast and accurate calling of germline and somatic variants. *Nat. Methods* **15**, 591–594 (2018).

19. Cooke, D. P., Wedge, D. C. & Lunter, G. A unified haplotype-based method for accurate and comprehensive variant calling. *Nat. Biotechnol.* **39**, 885–892 (2021).

20. Zheng, Z. *et al.* Symphonizing pileup and full-alignment for deep learning-based long-read variant calling. *Nat. Comput. Sci.* **2**, 797–803 (2022).

21. Chen, X. *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32**, 1220–1222 (2016).

22. Rausch, T. *et al.* DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28**, i333–i339 (2012).

23. Layer, R. M., Chiang, C., Quinlan, A. R. & Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.* **15**, R84 (2014).

24. Wala, J. A. *et al.* SvABA: genome-wide detection of structural variants and indels by local assembly. *Genome Res.* **28**, 581–591 (2018).

25. Cleal, K. & Baird, D. M. Dysgu: efficient structural variant calling using short or long reads. *Nucleic Acids Res.* **50**, e53 (2022).

26. Zarate, S. *et al.* Parliament2: Accurate structural variant calling at scale. *GigaScience* **9**, giaa145 (2020).

27. Smolka, M. *et al.* Detection of mosaic and population-level structural variants with Sniffles2. *Nat. Biotechnol.* **42**, 1571–1580 (2024).

28. Jiang, T. *et al.* Long-read-based human genomic structural variation detection with cuteSV. *Genome Biol.* **21**, 189 (2020).

29. Heller, D. & Vingron, M. SVIM: structural variant identification using mapped long reads. *Bioinformatics* **35**, 2907–2915 (2019).

30. Zheng, X. *et al.* A high-performance computing toolset for relatedness and principal component analysis of SNP data. *Bioinformatics* **28**, 3326–3328 (2012).

31. Falush, D., Stephens, M. & Pritchard, J. K. Inference of Population Structure Using Multilocus Genotype Data: Linked Loci and Correlated Allele Frequencies. *Genetics* **164**, 1567–1587 (2003).

32. Alexander, D. H., Novembre, J. & Lange, K. Fast model-based estimation of ancestry in unrelated individuals. *Genome Res.* **19**, 1655–1664 (2009).

33. Frichot, E. & François, O. LEA: An R package for landscape and ecological association studies. *Methods Ecol. Evol.* **6**, 925–929 (2015).

34. Jombart, T. adegenet: a R package for the multivariate analysis of genetic markers. *Bioinformatics* **24**, 1403–1405 (2008).

35. VanRaden, P. M. Efficient Methods to Compute Genomic Predictions. *J. Dairy Sci.* **91**, 4414–4423 (2008).

36. Wang, J. & Zhang, Z. GAPIT Version 3: Boosting Power and Accuracy for Genomic Association and Prediction. *Genomics Proteomics Bioinformatics* https://doi.org/10.1016/j.gpb.2021.08.005 (2021) doi:10.1016/j.gpb.2021.08.005.

37. Bradbury, P. J. *et al.* TASSEL: software for association mapping of complex traits in diverse samples. *Bioinformatics* **23**, 2633–2635 (2007).

38. Genome-wide efficient mixed-model analysis for association studies | Nature Genetics. https://www.nature.com/articles/ng.2310.

39. Korunes, K. L. & Samuk, K. pixy: Unbiased estimation of nucleotide diversity and divergence in the presence of missing data. *Mol. Ecol. Resour.* **21**, 1359–1368 (2021).

40. Chang, C. C. *et al.* Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience* **4**, s13742-015-0047–8 (2015).

41. Kang, H. M. *et al.* Variance component model to account for sample structure in genome-wide association studies. *Nat. Genet.* **42**, 348–354 (2010).

42. Iterative Usage of Fixed and Random Effect Models for Powerful and Efficient Genome-Wide Association Studies | PLOS Genetics. https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1005767.

43. Huang, M., Liu, X., Zhou, Y., Summers, R. M. & Zhang, Z. BLINK: a package for the next level of genome-wide association studies with both individuals and markers in the millions. *GigaScience* **8**, giy154 (2019).

44. Waples, R. S., Waples, R. K. & Ward, E. J. Pseudoreplication in genomic-scale data sets. *Mol. Ecol. Resour.* **22**, 503–518 (2022).

45. Hohenlohe, P. A., Amish, S. J., Catchen, J. M., Allendorf, F. W. & Luikart, G. Next-generation RAD sequencing identifies thousands of SNPs for assessing hybridization between rainbow and westslope cutthroat trout. *Mol. Ecol. Resour.* **11**, 117–122 (2011).

46. Double-digest RAD-sequencing: do pre- and post-sequencing protocol parameters impact biological results? | Molecular Genetics and Genomics. https://link.springer.com/article/10.1007/s00438-020-01756-9.

47. Browning, B. L., Zhou, Y. & Browning, S. R. A One-Penny Imputed Genome from Next-Generation Reference Panels. *Am. J. Hum. Genet.* **103**, 338–348 (2018).

48. Peterson, R. A. Finding Optimal Normalizing Transformations via bestNormalize. *R J.* **13**, 294–313 (2021).

49. Bonferroni, C. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni R Ist. Super. Sci. Econ. E Commericiali Firenze* **8**, 3–62 (1936).

50. Šidák, Z. Rectangular Confidence Regions for the Means of Multivariate Normal Distributions. *J. Am. Stat. Assoc.* **62**, 626–633 (1967).

51. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B Methodol.* **57**, 289–300 (1995).

52. Duggal, P., Gillanders, E. M., Holmes, T. N. & Bailey-Wilson, J. E. Establishing an adjusted p-value threshold to control the family-wide type 1 error in genome wide association studies. *BMC Genomics* **9**, 516 (2008).

53. Storey, J. D. & Tibshirani, R. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci.* **100**, 9440–9445 (2003).

54. Liu, F. *et al.* Genomes of cultivated and wild Capsicum species provide insights into pepper domestication and population differentiation. *Nat. Commun.* **14**, 5487 (2023).

55. Lozano-Isla, F. *Inti: Tools and Statistical Procedures in Plant Science*. (2025).

56. Yin, L. *et al.* rMVP: A Memory-efficient, Visualization-enhanced, and Parallel-accelerated Tool for Genome-wide Association Study. *Genomics Proteomics Bioinformatics* **19**, 619–628 (2021).

57. Stam, P. Construction of integrated genetic linkage maps by means of a new computer package JoinMap. *Plant J* **3**, 739–744 (1993).

58. Broman, K. W. *et al.* R/qtl: QTL mapping in experimental crosses. *Bioinforma. Oxf. Engl.* **19**, 889–90 (2003).

59. Mohseni, A. & Lonardi, S. MSTmap Online: enhanced usability, visualization, and accessibility. *Nucleic Acids Res.* **53**, W427–W430 (2025).