Inbred Line Genomic Prediction Pipeline

Technical Documentation

Project Type: Genomic Prediction & Cross-Validation for Inbred Lines
Language: R
Core Packages: `sommer`, `parallel`, `dplyr`, `tidyverse`, `arrow`, `paws`, `ggplot2`
Compute Model: Multi-core parallel processing with shared-memory clusters

---

1. Main overview

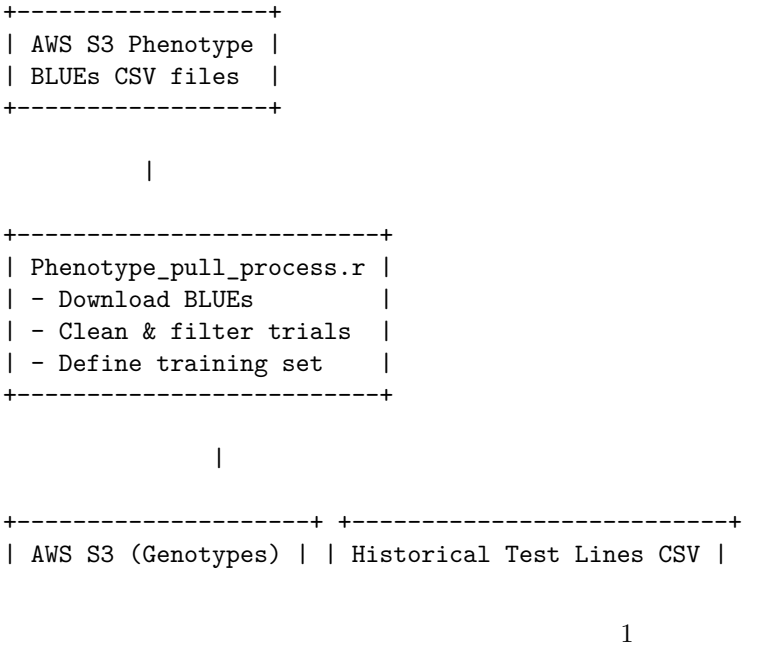
This pipeline performs **genomic prediction and model evaluation for inbred parental lines** using multi-environment phenotypic data and high-density genotype data. The workflow:

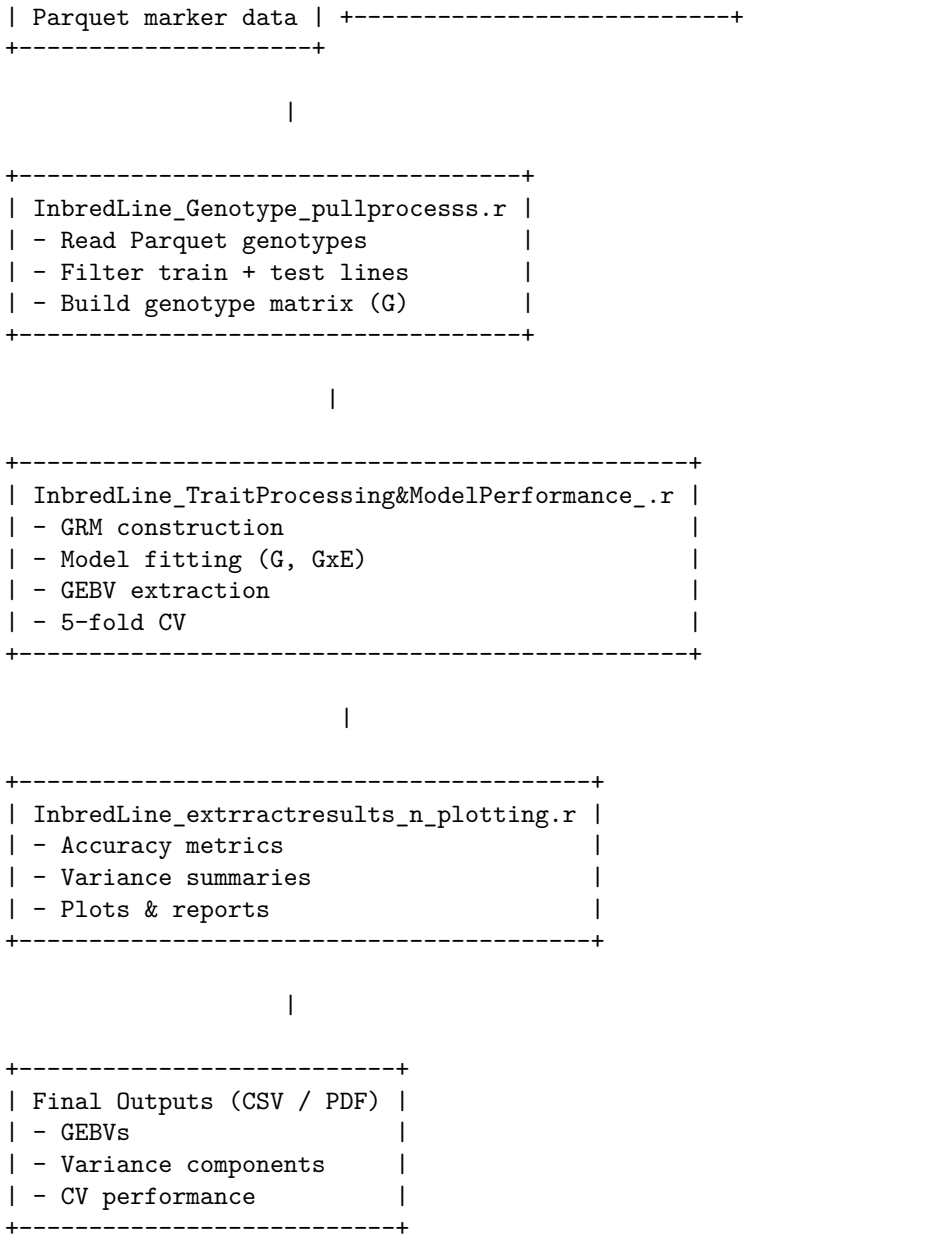
1. Pulls and processes phenotype data from AWS S3
2. Pulls and processes genotype data from AWS S3 (Parquet/VCF-derived)
3. Constructs Genomic Relationship Matrices (GRMs)
4. Fits genomic mixed models (G-only and G×E)
5. Generates Genomic Estimated Breeding Values (GEBVs)
6. Evaluates model performance using:
    - Training-set prediction accuracy
    - 5-fold cross-validation (mid-parent correlation)
7. Produces publication-ready summaries, tables, and plots

The pipeline is modular and scalable, designed for large breeding datasets.

---

2. Script-Workflow Diagram





---

3. Main Script

InbredLine prediction pipeline.R

Purpose

The **main script** mobilizes the entire analysis. It: - Loads required packages - Sources reuired helper scripts - Initializes parallel workers - Executes trait-wise model fitting and cross-validation - Aggregates results - Generates final outputs (CSV, PDF)

Script Framework

3.1 Package Loading Core statistical, genomic, and parallelization libraries are loaded: - `sommer` for mixed models - `parallel` for multi-core execution - `tidyverse` and `ggplot2` for data manipulation and plotting

3.2 Source Files The pipeline is split into logically independent source files:

File	Responsibility
`Phenotype_pull_process.r`	Pull and preprocess phenotypic data
`InbredLine_Genotype_pullprocesss.r`	Pull and preprocess genotype data
`InbredLine_TraitProcessing&ModelPerformance_Function.r`	GRM construction, modeling, CV
`InbredLine_extrractresults_n_plotting.r`	Result aggregation and visualization

This design improves maintainability and reproducibility.

3.3 Parallel Execution Two independent clusters are created:

- **CV Cluster (`cl_cv`)**: Used for 5-fold cross-validation
- **GEBV Cluster (`cl_gebv`)**: Used for full-data model fitting and GEBV extraction

Each cluster: - Loads required libraries - Receives shared objects via `clusterExport()`

Trait-level parallelism is achieved using `parLapply()`.

3.4 Result Collection and Cleanup

- Results are collected using `mccollect()`
- Clusters are explicitly stopped to free resources

3.5 Output Generation The main script: - Combines BLUPs and variance components - Writes GEBVs in wide format (one column per trait) - Produces model-wise CSV summaries - Generates PDF reports for trait correlations and CV performance

---

4. Source File Documentation

4.1 Phenotype Pull & Processing

File: Phenotype pull process.r‘

Purpose

This script: - Connects to AWS S3 - Downloads BLUEs (Best Linear Unbiased Estimates) - Cleans and formats phenotypic data - Defines the training population

Key Steps

AWS Connection

- Uses `paws::s3()` with credential-based authentication
- Pulls phenotype CSV files from structured S3 directories

Data Assembly

- Merges BLUEs
- Writes a local cached phenotype file

Phenotype Cleaning

- Converts trait columns to numeric
- Defines environments as `location × year`
- Filters:
  - Relevant trials
  - Trusted QC status
  - Target locations

Training Set Definition

- Extracts parental inbred lines from hybrid crosses
- Removes non-inbred or reference lines

Key Outputs: - `Pheno_final` - `TrainIDs`

4.2 Genotype Pull & Processing

File: InbredLine Genotype pullprocesss.r

Purpose

This module: - Reads genome-wide marker data from AWS S3 - Filters samples to training and test lines - Constructs a clean genotype matrix

Key Steps

Genotype Data Access

- Reads Parquet-partitioned genotype files using `arrow::open_dataset()`
- Efficiently collects only relevant samples

Sample Selection

- Combines:
  - Training parents
  - Historical test lines

Marker Cleaning

- Retains numeric marker columns only
- Removes samples with excessive missing data
- Deduplicates samples (least-missing rule)

Genotype Matrix Construction

- Rows = genotypes
- Columns = markers
- Output matrix is numeric and model-ready

Key Output: - `G` (genotype matrix)

4.3 GRM Construction & Model Functions

File: InbredLine TraitProcessingModelPerformance Function.r

Purpose

This is the **core analytical engine** of the pipeline. It: - Builds GRMs - Fits genomic mixed models - Performs GEBV prediction - Runs 5-fold cross-validation

Key Components

GRM Construction

- Marker-centered genotype matrix
- Uses `sommer::A.mat()`
- Separate GRMs for:
  - Full dataset
  - Training-only CV

Supported Models

Model	Description
G-only	Genetic main effects
G×E	Genetic + genotype-by-environment interaction

Trait Processing (`process_trait()`) For each trait:

1. Builds trait-specific phenotype table
2. Fits all models using `mmes()`
3. Extracts:
    - BLUPs
    - GEBVs plus SEs
    - Variance components
    - Heritability

Robust fitting includes fallback strategies for singularities.

Parallel Workers

- `process_trait_worker()` for full data
- `process_trait_worker_CV()` for 5-fold CV

Shared objects are attached/detached to avoid environment pollution.

Cross-Validation Logic

- Hybrid-level masking
- Mid-parent GEBV calculation
- Pearson correlation with observed hybrid means

Key Outputs: - Trait-wise BLUPs - GEBVs plus SEs - Variance components - Fold-level CV correlations

---

4.4 Result Aggregation & Visualization

File: InbredLine extrractresults n plotting.r

Purpose

This module summarizes and visualizes model performance.

Functions

`analyze_results()`

- Merges observed and predicted values
- Computes trait-wise correlations
- Produces faceted scatter plots

`analyze_5foldcv_results()`

- Aggregates fold-level CV results
- Generates:
  - Mean CV bar plots
  - Fold-wise comparison plots

Output Formats

- CSV summaries
- Publication-ready PDF plots

---

5. Outputs Summary

Output Type	Description
GEBV Tables	Predicted inbred line values
Variance CSVs	Genetic & residual variance
Heritability	Trait × model summaries
Correlation Plots	Observed vs predicted
CV Reports	Fold & mean CV accuracy

Env Correlations per trait

GxEmodel Inbredline5FoldCV results

Trait Correlations AcrossEnvironemtMeans

---

6. Intended Use

This pipeline is intended for: - Breeding program - generation advancemention selection decision support - Parental selection - Model benchmarking (G vs G×E) - Research and production-scale genomic prediction

---

Author: Vinay Kumar Reddy NANNURU **Status:** Production-ready **Maintenance:** Scalable, configurable, and expandable