Package: PONG2 (via r-universe)

June 25, 2026

Type Package

Title KIR Genotype Imputation and Model Training from SNP Array Data

Version 1.0.1

Date 2026-05-30

Description A scalable and accurate tool for Killer-cell Immunoglobulin-like Receptor (KIR) genotype imputation directly from SNP array data using supervised machine learning models trained across five continental ancestry groups. Uses attribute bagging and an ensemble classifier method with haplotype inference for SNPs and KIR types. Models are built from global populations in the 1000 Genomes Project and validated across diverse biobank cohorts. Methods are based on Zheng et al. (2014) <doi:10.1016/j.ajhg.2013.12.015> and Sadeeq et al. (2026) <`https://github.com/NormanLabUCD/PONG2`>.

Maintainer Suraju A. Sadeeq <`suraju.sadeeq@cuanschutz.edu`>

License GPL-3

URL `https://normanlabucd.github.io/PONG2/`, `https://github.com/NormanLabUCD/PONG2`

BugReports `https://github.com/NormanLabUCD/PONG2/issues`

Depends R (>= 4.0.0)

Imports parallel, graphics, stats, utils, tools

LinkingTo Rcpp, RcppParallel

Suggests HIBAG, knitr, rmarkdown, pkgdown, testthat

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements PLINK2 (>= 2.0), minimac4 (>= 4.1.6, optional)

Config/pak/sysreqs make

Repository https://normanlabucd.r-universe.dev

Date/Publication 2026-06-18 16:41:29 UTC

RemoteUrl https://github.com/normanlabucd/pong2

RemoteRef HEAD

RemoteSha 8d5abc2df6c2d49bda9d06cea64a0ec01c116279

Contents

`kirParallelAttrBagging` *Train KIR prediction models in parallel*

Description

Train KIR genotype prediction models using parallel attribute bagging across multiple CPU cores. This is the core training function used by the pong2 `train` CLI command.

Usage

```
kirParallelAttrBagging(
  cl,
  hla,
  snp,
  auto.save = "",
  nclassifier = 100,
  mtry = c("sqrt", "all", "one"),
  prune = TRUE,
  rm.na = TRUE,
  stop.cluster = FALSE,
  verbose = TRUE
)
```

Arguments

`cl`	a cluster object created by `parallel::makeCluster()` for parallel computation across multiple CPU cores
`hla`	a KIR allele table object of class `hlaAlleleClass` containing training allele calls
`snp`	a SNP genotype object of class `hlaSNPGenoClass` containing training SNP data

`auto.save`	character string; file path prefix for auto-saving classifiers during training. Use `""` (default) to disable
`nclassifier`	integer; number of individual ensemble classifiers to train (default: 100)
`mtry`	character; number of SNPs randomly selected at each node. One of `"sqrt"` (default), `"all"`, or `"one"`
`prune`	logical; if `TRUE` (default), prune classifiers
`rm.na`	logical; if `TRUE` (default), remove samples with missing KIR allele calls
`stop.cluster`	logical; if `TRUE`, stop the parallel cluster after training (default: `FALSE`)
`verbose`	logical; if `TRUE` (default), print progress

Value

An object of class `hlaAttrBagClass` representing the trained PONG2 KIR prediction model. The object contains:

n.samp integer; number of training samples

n.snp integer; number of SNP predictors used

hla.locus character; the KIR locus name

hla.allele character vector; KIR alleles in the model

classifiers list; individual ensemble classifiers

out.of.bag.acc numeric; out-of-bag accuracy estimate

Use `kirPredict()` to apply the model to new samples.

Examples

```
# Load example data
data(PONG2_example)

# Set up parallel cluster
cl <- parallel::makeCluster(2)

# Train a small model
model <- kirParallelAttrBagging(
  cl          = cl,
  hla         = example_kir,
  snp         = example_snp,
  nclassifier = 20,
  verbose     = FALSE
)

parallel::stopCluster(cl)

# View model summary
print(model)

# Clean up
hlaClose(model)
```

`kirPredict` *Predict KIR genotypes from SNP data*

Description

Predict KIR genotypes for a set of samples using a trained PONG2 attribute bagging model. This is the core prediction function used by the `pong2 impute` CLI command.

Usage

```
kirPredict(
  object,
  snp,
  cl = FALSE,
  type = c("response+dosage", "response", "prob", "response+prob"),
  vote = c("prob", "majority"),
  allele.check = TRUE,
  match.type = c("Position", "Pos+Allele", "RefSNP+Position", "RefSNP"),
  same.strand = FALSE,
  verbose = TRUE,
  verbose.match = TRUE
)
```

Arguments

`object`	a PONG2 model object of class `hlaAttrBagClass` as returned by `kirParallelAttrBagging()`
`snp`	a SNP genotype object of class `hlaSNPGenoClass` containing the target samples to impute
`cl`	a cluster object for parallel computation, or `FALSE` (default) for single-threaded prediction
`type`	character; type of prediction output. One of: `"response+dosage"` (default; predicted alleles + dosage scores), `"response"` (predicted alleles only), `"prob"` (posterior probabilities only), `"response+prob"` (predicted alleles + posterior probabilities)
`vote`	character; voting method for ensemble classifiers. One of `"prob"` (default; probability-weighted voting) or `"majority"` (majority vote)
`allele.check`	logical; if `TRUE` (default), check and validate allele names against the model
`match.type`	character; SNP matching method. One of `"Position"` (default), `"Pos+Allele"`, `"RefSNP+Position"`, or `"RefSNP"`
`same.strand`	logical; if `TRUE`, assume SNPs are on the same strand (default: `FALSE`)
`verbose`	logical; if `TRUE` (default), print progress messages
`verbose.match`	logical; if `TRUE` (default), print SNP matching summary

Value

An object of class `hlaAlleleClass` containing KIR imputation results. The object includes:

value data frame with columns `sample.id`, `allele1`, `allele2`, and `prob` (posterior probability of the best call)

dosage numeric matrix of allele dosage scores (samples x alleles); `NULL` if `type = "response"`

postprob numeric matrix of posterior probabilities (alleles x samples); `NULL` unless `type = "response+prob"` or `"prob"`

Samples with posterior probability below the call threshold (`CT`) are assigned `NA` for both alleles.

Examples

```
# Load example data
data(PONG2_example)

# Load model from object
model <- hlaModelFromObj(example_mobj)

# Predict KIR genotypes
pred <- kirPredict(
  object  = model,
  snp     = example_snp,
  type    = "response+prob",
  verbose = FALSE
)

# View results
head(pred$value)

# Clean up
hlaClose(model)
```

`PONG2_example` *PONG2 Example Dataset*

Description

A small example dataset for demonstrating PONG2 functions. Contains 50 samples and 200 SNPs in the KIR region (chr19), along with a pre-trained KIR3DL1 model with 10 classifiers.

Usage

```
data(PONG2_example)
```

Format

Three objects are loaded:

example_snp A `hlaSNPGenoClass` object with 50 samples and 200 SNPs in the KIR region (chr19, hg19 assembly)

example_kir A `hlaAlleleClass` object with KIR3DL1 allele calls for 50 samples

example_mobj A `hlaAttrBagObj` object — a pre-trained KIR3DL1 model with 10 ensemble classifiers

Examples

```
data(PONG2_example)
# SNP data
cat("Samples:", ncol(example_snp$genotype), "\n")
cat("SNPs:   ", nrow(example_snp$genotype), "\n")
# KIR allele table
cat("Locus:  ", example_kir$locus, "\n")
# Model
cat("Classifiers:", length(example_mobj$classifiers), "\n")
```

Index