Package 'HaploCatcher'

June 9, 2026

Title A Predictive Haplotyping Package

Date 2026-06-01

Version 2.0.1

Description Used for predicting a genotype's allelic state at a specific locus/QTL/gene. This is accomplished by using both a genotype matrix and a separate file which has categorizations about loci/QTL/genes of interest for the individuals in the genotypic matrix. A training population can be created from a panel of individuals who have been previously screened for specific loci/QTL/genes, and this previous screening could be summarized into a category. Using the categorization of individuals which have been genotyped using a genome wide marker platform, a model can be trained to predict what category (haplotype) an individual belongs in based on their genetic sequence in the region associated with the locus/QTL/gene. These trained models can then be used to predict the haplotype of a locus/QTL/gene for individuals which have been genotyped with a genome wide platform yet not genotyped for the specific locus/QTL/gene. This package is based off work done by Winn et al 2021. For more specific information on this method, refer to <doi:10.1007/s00122-022-04178-w>.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/zjwinn/HaploCatcher

BugReports https://github.com/zjwinn/HaploCatcher/issues

Imports parallel, doParallel, foreach, caret, ggplot2, graphics, knitr, patchwork, randomForest, stats

Depends R (>= 2.10)

LazyData true

Suggests rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Zachary Winn [aut, cre] (ORCID: <https://orcid.org/0000-0003-1543-1527>)

Maintainer Zachary Winn <zwinn@outlook.com>

Repository CRAN

Date/Publication 2026-06-09 07:00:20 UTC

Contents

`auto_locus` *Auto Locus: An Automated Pipeline for Locus Prediction*

Description

Weaves the HaploCatcher functions into a single pipeline (Figure 1b of the package paper): permutation cross-validation, best-model selection by kappa or accuracy, then forward prediction either with one seeded model or by majority-rule voting over many random models.

Usage

```
auto_locus(
  geno_mat,
  gene_file,
  gene_name,
  marker_info,
  chromosome,
  training_genotypes,
  testing_genotypes,
  ncor_markers = 50,
  n_neighbors = 50,
  cv_percent_testing = 0.2,
  cv_percent_training = 0.8,
  n_perms = 30,
  model_selection_parameter = "kappa",
  n_votes = 30,
  set_seed = NULL,
  predict_by_vote = FALSE,
```

```
  include_hets = FALSE,
  include_models = FALSE,
  verbose = TRUE,
  parallel = FALSE,
  n_cores = NULL,
  plot_cv_results = TRUE,
  het_label = NULL,
  neg_label = NULL
)
```

Arguments

geno_mat	An imputed, number-coded genotypic matrix with n rows of individuals and m columns of markers. Row names are genotype IDs; column names are marker IDs. Missing data are not allowed. Numeric coding may vary as long as it is consistent across markers.
gene_file	A data frame with at least the columns 'Gene', 'FullSampleName', and 'Call'. 'Gene' is the gene each observation belongs to, 'FullSampleName' matches a column name in the genotypic matrix, and 'Call' is the marker call for that genotype.
gene_name	A character string matching a value in the 'Gene' column of `gene_file`.
marker_info	A data frame with the columns 'Marker', 'Chromosome', and 'BP_Position'. Every marker in the genotypic matrix must be listed. If positions are unavailable a numeric dummy (1..m) may be used.
chromosome	A character string matching a value in the 'Chromosome' column of `marker_info`.
training_genotypes	Character vector of FullSampleNames used for cross-validation and to train the prediction model.
testing_genotypes	Character vector of FullSampleNames to predict.
ncor_markers	Number of top correlated markers to retain for training. Default 50.
n_neighbors	Number of neighbors to consider in KNN. Default 50.
cv_percent_testing	Proportion reserved for validation during CV, strictly between 0 and 1. Default 0.20.
cv_percent_training	Proportion used for training during CV, strictly between 0 and 1. Default 0.80.
n_perms	Number of cross-validation permutations. Default 30.
model_selection_parameter	Metric for selecting the best model: "kappa" or "accuracy". Default "kappa".
n_votes	Number of models to train and predict with when voting. Default 30.
set_seed	Numeric seed for a single reproducible prediction (required when `predict_by_vote = FALSE`). Default NULL.
predict_by_vote	Logical; predict by majority rule over many random models. Default FALSE.

include_hets	Logical; keep heterozygous calls. Default FALSE.
include_models	Logical; keep the trained models in the CV result (large). Default FALSE.
verbose	Logical; print progress and plots. Default TRUE.
parallel	Logical; run CV and voting in parallel. Default FALSE.
n_cores	Number of cores for parallel processing. If NULL and parallel = TRUE, uses all available cores minus one.
plot_cv_results	Logical; draw the cross-validation summary plot. Default TRUE.
het_label	Optional character vector of Call values to treat as heterozygous. When NULL (default), the package convention (calls containing "het_") is used.
neg_label	Optional character vector of Call values to treat as the negative/wild-type case (used only for plot facet labels). When NULL (default), the "non_" prefix is used.

Value

A list. When predict_by_vote = FALSE: method, cross_validation_results, prediction_model, and predictions. When predict_by_vote = TRUE: method, cross_validation_results, predictions (per-vote calls), and consensus_predictions (majority rule).

Examples

```
#refer to vignette for an in depth look at the auto_locus function
vignette("An_Intro_to_HaploCatcher", package = "HaploCatcher")
```

---

gene_comp *Model Gene Compendium Data Set*

Description

A data frame which contains information from 1345 unique wheat lines on the Sst1 solid stem locus.

Usage

```
gene_comp
```

Format

A data frame with 1345 rows and 7 columns:

Trait A short discription of the phenotype associated with the gene

Chromosome The chromosome where the gene resides

Gene The name of the gene

Nursery The program which produced the gene call for the genotype

Line A breeder assigned line designation

FullSampleName A designation unique to the line found in the genotypic matrix

Call A 'call' given for the allelic state. For this package, it is best to format the non desiarable allele as "non_gene" and the heterozygous state as "het_gene".

Source

Generated by Zachary James Winn for the CSU breeding program via USDA-ARS gene reports and in-house gene assays

Examples

```
data("gene_comp") #lazy loads the dataset for use in the package
```

---

geno_mat *Model Gene Compendium Data Set*

Description

A numeric matrix which contains molecular marker information on 1345 unique genotypes for 2271 SNP markers located on wheat chromosome 3B. This data set corresponds to the information found in the "gene_comp" and "marker_info" data sets.

Usage

```
geno_mat
```

Format

A numeric matrix with 1345 rows and 2271 columns:

Source

Generated by Zachary James Winn for the CSU breeding program via historical in-house GBS data

Examples

```
data("geno_mat") #lazy loads the dataset for use in the package
```

locus_cv *Haplotype Prediction: Cross Validation of KNN and RF Models*

Description

Performs one round of the cross-validation featured in Winn et al. (2022): a random partition of the training data trains KNN and RF models, and a reserved test partition validates them. This is a single permutation; use `locus_perm_cv()` to repeat it.

Usage

```
locus_cv(
  geno_mat,
  gene_file,
  gene_name,
  marker_info,
  chromosome,
  ncor_markers = 50,
  n_neighbors = 50,
  percent_testing = 0.2,
  percent_training = 0.8,
  include_hets = FALSE,
  include_models = FALSE,
  verbose = TRUE,
  graph = FALSE,
  het_label = NULL
)
```

Arguments

`geno_mat`	An imputed, number-coded genotypic matrix with n rows of individuals and m columns of markers. Row names are genotype IDs; column names are marker IDs. Missing data are not allowed. Numeric coding may vary as long as it is consistent across markers.
`gene_file`	A data frame with at least the columns 'Gene', 'FullSampleName', and 'Call'. 'Gene' is the gene each observation belongs to, 'FullSampleName' matches a column name in the genotypic matrix, and 'Call' is the marker call for that genotype.
`gene_name`	A character string matching a value in the 'Gene' column of `gene_file`.
`marker_info`	A data frame with the columns 'Marker', 'Chromosome', and 'BP_Position'. Every marker in the genotypic matrix must be listed. If positions are unavailable a numeric dummy (1..m) may be used.
`chromosome`	A character string matching a value in the 'Chromosome' column of `marker_info`.
`ncor_markers`	Number of top correlated markers to retain for training. Default 50.
`n_neighbors`	Number of neighbors to consider in KNN. Default 50.

`percent_testing`
: Proportion of data reserved for validation, strictly between 0 and 1. Default 0.20.

`percent_training`
: Proportion of data used for training, strictly between 0 and 1. Default 0.80.

`include_hets`
: Logical; keep heterozygous calls. Default FALSE.

`include_models`
: Logical; keep the trained models in the result (large). Default FALSE.

`verbose`
: Logical; print progress and tables. Default TRUE.

`graph`
: Logical; draw the marker-correlation diagnostic. Default FALSE.

`het_label`
: Optional character vector of `Call` values to treat as heterozygous. When NULL (default), calls containing the prefix "het_" are used.

Value

A list with `data_frames` (training and test frames), `test_predictions` (per-model prediction frames), `confusion_matrices` (per-model confusion objects), and, when `include_models = TRUE`, `trained_models`.

Examples

```
#read in the genotypic data matrix
data("geno_mat")

#read in the marker information
data("marker_info")

#read in the gene compendium file
data("gene_comp")

#run the function without hets for a very limited number of markers and neighbors
#due to requirements by cran, this must be commented out
#to run, place this code in the console and remove comments
#fit<-locus_cv(geno_mat=geno_mat, #the genotypic matrix
#             gene_file=gene_comp, #the gene compendium file
#             gene_name="sst1_solid_stem", #the name of the gene
#             marker_info=marker_info, #the marker information file
#             chromosome="3B", #name of the chromosome
#             ncor_markers=2, #number of markers to retain
#             n_neighbors=1, #number of neighbors
#             percent_testing=0.2, #percentage of genotypes in the validation set
#             percent_training=0.8, #percentage of genotypes in the training set
#             include_hets=FALSE, #include hets in the model
#             include_models=TRUE, #include models in the final results
#             verbose=TRUE, #allows text output
#             graph=TRUE) #allows graph output
```

locus_perm_cv *Haplotype Prediction: Permutation Cross Validation of KNN and RF Models*

Description

Repeats locus_cv() over many random partitions (permutations) and summarizes the overall and by-class performance of the KNN and RF models. Can run sequentially or in parallel.

Usage

```
locus_perm_cv(
  n_perms = 30,
  geno_mat,
  gene_file,
  gene_name,
  marker_info,
  chromosome,
  ncor_markers = 50,
  n_neighbors = 50,
  percent_testing = 0.2,
  percent_training = 0.8,
  include_hets = FALSE,
  include_models = FALSE,
  verbose = FALSE,
  parallel = FALSE,
  n_cores = NULL,
  het_label = NULL
)
```

Arguments

n_perms	Number of permutations to perform. Default 30.
geno_mat	An imputed, number-coded genotypic matrix with n rows of individuals and m columns of markers. Row names are genotype IDs; column names are marker IDs. Missing data are not allowed. Numeric coding may vary as long as it is consistent across markers.
gene_file	A data frame with at least the columns 'Gene', 'FullSampleName', and 'Call'. 'Gene' is the gene each observation belongs to, 'FullSampleName' matches a column name in the genotypic matrix, and 'Call' is the marker call for that genotype.
gene_name	A character string matching a value in the 'Gene' column of gene_file.
marker_info	A data frame with the columns 'Marker', 'Chromosome', and 'BP_Position'. Every marker in the genotypic matrix must be listed. If positions are unavailable a numeric dummy (1..m) may be used.
chromosome	A character string matching a value in the 'Chromosome' column of marker_info.

ncor_markers	Number of top correlated markers to retain for training. Default 50.
n_neighbors	Number of neighbors to consider in KNN. Default 50.
percent_testing	Proportion of data reserved for validation, strictly between 0 and 1. Default 0.20.
percent_training	Proportion of data used for training, strictly between 0 and 1. Default 0.80.
include_hets	Logical; keep heterozygous calls. Default FALSE.
include_models	Logical; keep the trained models in each permutation (large). Default FALSE.
verbose	Logical; print per-permutation progress. Default FALSE.
parallel	Logical; run permutations in parallel. Default FALSE. When TRUE, textual/graphical feedback is suppressed.
n_cores	Number of cores for parallel processing. If NULL and `parallel = TRUE`, uses all available cores minus one.
het_label	Optional character vector of `Call` values to treat as heterozygous. When NULL (default), calls containing the prefix "het_" are used.

Value

A list with `Overall_Parameters`, `By_Class_Parameters`, `Overall_Summary`, `By_Class_Summary`, and `Raw_Permutation_Info`.

Examples

```
#read in the genotypic data matrix
data("geno_mat")

#read in the marker information
data("marker_info")

#read in the gene compendium file
data("gene_comp")

#run permutational analysis - commented out for package specifications
#to run, copy and paste without '#' into the console

#fit<-locus_perm_cv(n_perms = 10, #the number of permutations
#                   geno_mat=geno_mat, #the genotypic matrix
#                   gene_file=gene_comp, #the gene compendium file
#                   gene_name="sst1_solid_stem", #the name of the gene
#                   marker_info=marker_info, #the marker information file
#                   chromosome="3B", #name of the chromosome
#                   ncor_markers= 25, #number of markers to retain
#                   n_neighbors = 25, #number of nearest-neighbors
#                   percent_testing=0.2, #percentage of genotypes in the validation set
#                   percent_training=0.8, #percentage of genotypes in the training set
#                   include_hets=FALSE, #excludes hets in the model
#                   include_models=FALSE, #excludes models in results object
#                   verbose = FALSE) #excludes text
```

locus_pred *Haplotype Prediction: Using Trained Models to Make Predictions*

Description

Applies the models from locus_train() to forward-predict the haplotype of genotypes that have genome-wide marker data but no locus record.

Usage

```
locus_pred(locus_train_results, geno_mat, genotypes_to_predict)
```

Arguments

locus_train_results	The list returned by locus_train().
geno_mat	A genotypic matrix containing the genotypes to predict. The genome-wide markers must be shared with the training population.
genotypes_to_predict	A character vector of genotype names (rows of geno_mat) to predict. Names that were in the training data are dropped to avoid bias.

Value

A data frame with FullSampleName and one prediction column per trained model (Prediction_KNN and/or Prediction_RF).

Examples

```
#set seed for reproducible sampling
set.seed(022294)

#read in the genotypic data matrix
data("geno_mat")

#read in the marker information
data("marker_info")

#read in the gene compendium file
data("gene_comp")

#Note: in practice you would have something like a gene file
#that does not contain any lines you are trying to predict.
#However, this is for illustrative purposes on how to run the function

#sample data in the gene_comp file to make a traning population
train<-gene_comp[gene_comp$FullSampleName %in%
                   sample(gene_comp$FullSampleName,
                          round(length(gene_comp$FullSampleName)*0.8),0),]
```

```
#pull vector of names, not in the train, for forward prediction
test<-gene_comp[!gene_comp$FullSampleName
                %in% train$FullSampleName,
                "FullSampleName"]

#run the function with hets
fit<-locus_train(geno_mat=geno_mat, #the genotypic matrix
                 gene_file=train, #the gene compendium file
                 gene_name="sst1_solid_stem", #the name of the gene
                 marker_info=marker_info, #the marker information file
                 chromosome="3B", #name of the chromosome
                 ncor_markers=2, #number of markers to retain
                 n_neighbors=3, #number of neighbors
                 include_hets=FALSE, #include hets in the model
                 verbose = FALSE, #allows for text and graph output
                 set_seed = 022294, #sets a seed for reproduction of results
                 models_request = "knn") #sets what models are requested

#predict the lines in the test population
pred<-locus_pred(locus_train_results=fit,
                 geno_mat=geno_mat,
                 genotypes_to_predict=test)

#see predictions
head(pred)
```

`locus_train` *Haplotype Prediction: Training Models for Forward Prediction*

Description

Trains KNN and/or RF models on the full training data for use in forward prediction of lines that have no locus record. Shares all data preparation and model-fitting logic with `locus_cv()`.

Usage

```
locus_train(
  geno_mat,
  gene_file,
  gene_name,
  marker_info,
  chromosome,
  ncor_markers = 50,
  n_neighbors = 50,
  include_hets = FALSE,
  verbose = FALSE,
  set_seed = NULL,
```

```
  models_request = "all",
  graph = FALSE,
  het_label = NULL
)
```

Arguments

geno_mat	An imputed, number-coded genotypic matrix with n rows of individuals and m columns of markers. Row names are genotype IDs; column names are marker IDs. Missing data are not allowed. Numeric coding may vary as long as it is consistent across markers.
gene_file	A data frame with at least the columns 'Gene', 'FullSampleName', and 'Call'. 'Gene' is the gene each observation belongs to, 'FullSampleName' matches a column name in the genotypic matrix, and 'Call' is the marker call for that genotype.
gene_name	A character string matching a value in the 'Gene' column of gene_file.
marker_info	A data frame with the columns 'Marker', 'Chromosome', and 'BP_Position'. Every marker in the genotypic matrix must be listed. If positions are unavailable a numeric dummy (1..m) may be used.
chromosome	A character string matching a value in the 'Chromosome' column of marker_info.
ncor_markers	Number of top correlated markers to retain for training. Default 50.
n_neighbors	Number of neighbors to consider in KNN. Default 50.
include_hets	Logical; keep heterozygous calls. Default FALSE.
verbose	Logical; print progress and tables. Default FALSE.
set_seed	Numeric seed for reproducibility, or NULL. Default NULL.
models_request	Which models to train: "knn", "rf", or "all". Default "all".
graph	Logical; draw the marker-correlation diagnostic. Default FALSE.
het_label	Optional character vector of Call values to treat as heterozygous. When NULL (default), calls containing the prefix "het_" are used.

Value

A list with seed, models_request, trained_models, and data (the training frame). trained_models is a single caret model when one model was requested, or a list with knn and rf when "all".

Examples

```
#set seed for reproducible sampling
set.seed(022294)

#read in the genotypic data matrix
data("geno_mat")

#read in the marker information
data("marker_info")
```

```
#read in the gene compendium file
data("gene_comp")

#Note: in practice you would have something like a gene file
#that does not contain any lines you are trying to predict.
#However, this is for illustrative purposes on how to run the function

#sample data in the gene_comp file to make a traning population
train<-gene_comp[gene_comp$FullSampleName %in%
                   sample(gene_comp$FullSampleName,
                          round(length(gene_comp$FullSampleName)*0.8),0),]

#pull vector of names, not in the train, for forward prediction
test<-gene_comp[!gene_comp$FullSampleName
                %in% train$FullSampleName,
                "FullSampleName"]

#run the function with hets
fit<-locus_train(geno_mat=geno_mat, #the genotypic matrix
                 gene_file=train, #the gene compendium file
                 gene_name="sst1_solid_stem", #the name of the gene
                 marker_info=marker_info, #the marker information file
                 chromosome="3B", #name of the chromosome
                 ncor_markers=2, #number of markers to retain
                 n_neighbors=3, #number of neighbors
                 include_hets=FALSE, #include hets in the model
                 verbose = FALSE, #allows for text and graph output
                 set_seed = 022294, #sets a seed for reproduction of results
                 models_request = "knn") #sets what models are requested

#predict the lines in the test population
pred<-locus_pred(locus_train_results=fit,
                 geno_mat=geno_mat,
                 genotypes_to_predict=test)

#see predictions
head(pred)
```

`marker_info` *Model Gene Compendium Data Set*

Description

A data frame which contains marker information of GBS markers found on wheat chromosome 3B. This data pairs with the markers found in "geno_mat" data file associated with the HaploCatcher package.

Usage

```
marker_info
```

Format

A data frame with 2271 rows and 3 columns:

Marker The designation of the markers which are found in the genotypic matrix

Chromosome The chromosome where each marker resides

BP_Position The position of each marker in basepairs

Source

Generated by Zachary James Winn for the CSU breeding program via historical in-house GBS data

Examples

```
data("marker_info") #lazy loads the dataset for use in the package
```

---

plot_locus_perm_cv *Visualize Permutation CV Results*

Description

Takes the result of `locus_perm_cv()` and draws a composite of accuracy, kappa, sensitivity, and specificity across permutations. When more than one call class is present (heterozygotes retained), sensitivity and specificity are faceted by class.

Usage

```
plot_locus_perm_cv(
  results,
  individual_images = FALSE,
  het_label = NULL,
  neg_label = NULL
)
```

Arguments

`results`	A list produced by `locus_perm_cv()`.
`individual_images`	Logical; also print each panel on its own. Default FALSE.
`het_label`	Optional character vector of class labels to treat as heterozygous when relabeling facets. When NULL (default), the "het_" prefix is used.
`neg_label`	Optional character vector of class labels to treat as the negative/wild-type case when relabeling facets. When NULL (default), the "non_" prefix is used.

Value

Invisibly returns NULL; called for its plotting side effect.

Examples

```
#refer to vignette for an in depth look at the plot_locus_perm_cv function
vignette("An_Intro_to_HaploCatcher", package = "HaploCatcher")
```

Index