Package 'CHAPGWAS'

May 7, 2026

Type Package

Title CHAP-GWAS: Leveraging Chromosomal Haplotypes to Improve Genome-Wide Association Studies

Version 0.1.3

Description CHAP-GWAS (Chromosomal Haplotype-Integrated Genome-Wide Association Study) provides a dynamically adaptive framework for genome-wide association studies (GWAS) that integrates chromosome-scale haplotypes with single nucleotide polymorphism (SNP) analysis. The method identifies and extends haplotype variants based on their phenotypic associations rather than predefined linkage blocks, enabling high-resolution detection of quantitative trait loci (QTL). By leveraging long-range phased haplotype information, CHAP-GWAS improves statistical power and offers a more comprehensive view of the genetic architecture underlying complex traits.

Depends R (>= 4.0.0)

Imports MASS, plyr

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Shibo Wang [aut, cre],
Qiong Jia [aut],
Zhenyu Jia [aut, ctb]

Maintainer Shibo Wang <shibow@ucr.edu>

Repository CRAN

Date/Publication 2026-01-08 04:10:02 UTC

Contents

RANDOM *RANDOM: Mixed model and haplotype random effect test*

Description

Internal core function used in SEL.HAP() to fit mixed models and test the random effect of a haplotype design matrix z.

Usage

```
RANDOM(z, YFIX, KIN, PAR)
```

Arguments

z	Haplotype design matrix (n x r). When PAR is NULL, z can be NULL and the function only estimates variance components in the mixed model.
YFIX	A data.frame or matrix with phenotype in the first column and fixed-effect covariates in the remaining columns.
KIN	A list of kinship matrices.
PAR	Optional vector of (log) variance components. If NULL, they are estimated by the internal mixed model.

Value

If PAR is NULL, returns the estimated parameters (vector). Otherwise returns a list with likelihood ratio test statistics and parameter estimates.

SEL.HAP *SEL.HAP: Haplotype selection and extension along the genome*

Description

Perform genome-wide haplotype selection and extension using the CHAP-GWAS framework. The function scans along each chromosome, builds local haplotype segments, and adaptively extends them based on association evidence with the phenotype.

Usage

```
SEL.HAP(GEN, YFIX, KIN, nHap, p.threshold, PAR)
```

Arguments

GEN	Genotype matrix with rows corresponding to markers and columns corresponding to individuals. The first two columns give chromosome (chr) and physical position (pos); the remaining columns contain alleles for each individual (e.g. "A", "C", "G", "T"), one allele per haplotype copy.
YFIX	A matrix or data.frame with phenotype in the first column and fixed-effect covariates (e.g. intercept, PCs) in the remaining columns, one row per individual.
KIN	A list of kinship matrices, each of dimension $n \times n$, where n is the number of individuals.
nHap	Initial haplotype window size (number of consecutive markers).
p.threshold	P-value threshold for haplotype extension.
PAR	Optional variance component parameters passed to RANDOM(). If NULL, they are estimated internally.

Value

A list of three matrices summarizing:

- **FINAL[[1]]**: initial haplotype segments
- **FINAL[[2]]**: extended haplotype segments
- **FINAL[[3]]**: final selected segments after extension

Examples

```
## Minimal example with small simulated data (alleles encoded as A/C/G/T)
set.seed(1)

## Number of individuals and markers
n_ind  <- 200
n_mark <- 50

## Construct a simple GEN matrix:
## first two columns: chromosome and position
## each individual is represented by two allele columns (A1/A2)
chr <- rep(1, n_mark)
pos <- seq_len(n_mark) * 100
alleles <- c("A", "C", "G", "T")

geno <- matrix(NA_character_, nrow = n_mark, ncol = 2 * n_ind)
for (m in seq_len(n_mark)) {
  a <- sample(alleles, 2, replace = FALSE)   # biallelic per marker
  geno[m, ] <- sample(a, 2 * n_ind, replace = TRUE)
}

colnames(geno) <- as.vector(rbind(
  paste0("id", seq_len(n_ind), "_A1"),
  paste0("id", seq_len(n_ind), "_A2")
))
```

```
GEN <- cbind(chr, pos, geno)

## Phenotype + intercept as fixed effect
y <- rnorm(n_ind)
X <- cbind(1, rnorm(n_ind))  # intercept + one covariate
YFIX <- cbind(y, X)

## Simple kinship: identity matrix
KIN <- list(diag(n_ind))

## Run SEL.HAP with a small initial window and mild threshold
res <- SEL.HAP(GEN, YFIX, KIN,
               nHap = 2,
               p.threshold = 0.05,
               PAR = NULL)

## Inspect the structure of the result (three matrices)
str(res)
```

`test.HAP` *test.HAP: Test haplotype effects for a local region*

Description

test.HAP: Test haplotype effects for a local region

Usage

```
test.HAP(HAP.X, YFIX, KIN, PAR)
```

Arguments

`HAP.X`	Haplotype genotype matrix for a candidate genomic region. Each column corresponds to a SNP within the region. Every two consecutive rows correspond to the two homologous haplotypes of the same individual (rows 1–2 = individual 1, rows 3–4 = individual 2, etc.).
`YFIX`	Phenotype and fixed-effect covariates.
`KIN`	List of kinship matrices.
`PAR`	Optional vector of variance components; if NULL, estimated internally.

Value

A list containing: * my.scan – scan statistics * z – haplotype design matrix (n x number of haplotypes) * ELEMENT – labels of distinct haplotype categories

Index