Package ‘snpsettest’

Title A Set-Based Association Test using GWAS Summary Statistics

Version 0.1.1

Description The goal of ‘snpsettest’ is to provide simple tools that perform set-based association tests (e.g., gene-based association tests) using GWAS (genome-wide association study) summary statistics. A set-based association test in this package is based on the statistical model described in VEGAS (versatile gene-based association study), which combines the effects of a set of SNPs accounting for linkage disequilibrium between markers. This package uses a different approach from the original VEGAS implementation to compute set-level p values more efficiently, as described in <https://github.com/HimesGroup/snpsettest/wiki/Statistical-test-in-snpsettest>.

License GPL (>= 3)

Depends R (>= 3.1.0)

Imports gaston, data.table, Rcpp

Suggests tidyr, knitr, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

URL https://github.com/HimesGroup/snpsettest

BugReports https://github.com/HimesGroup/snpsettest/issues

NeedsCompilation yes

Author Jaehyun Joo [aut, cre], Blanca Himes [aut]

Maintainer Jaehyun Joo <jaehyunjoo@outlook.com>

Repository CRAN

Date/Publication 2022-01-18 03:40:02 UTC
R topics documented:

- exGWAS
- gene.curated.GRCh37
- gene.curated.GRCh38
- harmonize_sumstats
- map_snp_to_gene
- read_reference_bed
- snpset_test

Description

An example file of GWAS summary statistics

Usage

exGWAS

Format

Data frame with columns

- id  SNP ID.
- chr chromosome.
- pos base-pair position.
- A1, A2  allele codes.
- pvalue  p value.

Examples

head(exGWAS)
gene.curated.GRCh37  

*Human gene information from the GENCODE GRCh37 version*

**Description**

Human gene information was extracted from the GENCODE release 19. This data only contains 'KNOWN' status genes with the following gene biotypes: protein-coding, Immunoglobulin (Ig) variable chain and T-cell receptor (TcR) genes.

**Usage**

gene.curated.GRCh37

**Format**

Data frame with columns

- **gene.id**  SNP ID.
- **chr**  chromosome.
- **start**  genomic start location (1-based).
- **end**  genomic end location.
- **strand**  genomic strand.
- **gene.name**  gene symbols mapped to the GENCODE genes.
- **gene.type**  gene biotypes in the GENCODE genes.

**Source**


**Examples**

head(gene.curated.GRCh37)

---

gene.curated.GRCh38  

*Human gene information from the GENCODE GRCh38 version*

**Description**

Human gene information was extracted from the GENCODE release 37. This data only contains genes with the following gene biotypes: protein-coding, Immunoglobulin (Ig) variable chain and T-cell receptor (TcR) genes.

**Usage**

gene.curated.GRCh38
Format

Data frame with columns

- **gene.id**: SNP ID.
- **chr**: chromosome.
- **start**: genomic start location (1-based).
- **end**: genomic end location.
- **strand**: genomic strand.
- **gene.name**: gene symbols mapped to the GENCODE genes.
- **gene.type**: gene biotypes in the GENCODE genes.

Source

https://www.gencodegenes.org/human/release_37.html

Examples

`head(gene.curated.GRCh38)`

---

**harmonize_sumstats**  
*Harmonizing GWAS summary to reference data*

Description

Finds an intersection of variants between GWAS summary and reference data.

Usage

```r
harmonize_sumstats(
  sumstats,
  x,
  match_by_id = TRUE,
  check_strand_flip = FALSE,
  return_indice = FALSE
)
```

Arguments

- **sumstats**: A data frame with two columns: "id" and "pvalue".
  - id = SNP ID (e.g., rs numbers)
  - pvalue = SNP-level p value
  
  If `match_by_id = FALSE`, it requires additional columns: "chr", "pos", "A1" and "A2".
  - chr = chromosome
```
harmonize_sumstats

- pos = base-pair position (must be integer)
- A1, A2 = allele codes (allele order is not important)

x A bed.matrix object created using the reference data.

match_by_id If TRUE, SNP matching will be performed by SNP IDs instead of genomic position and allele codes. Default is TRUE.

check_strand_flip Only applies when match_by_id = FALSE. If TRUE, the function 1) removes ambiguous A/T and G/C SNPs for which the strand is not obvious, and 2) attempts to find additional matching entries by flipping allele codes (i.e., A->T, T->A, C->G, G->A). If the GWAS genotype data itself is used as the reference data, it would be safe to set FALSE. Default is FALSE.

return_indice Only applied when match_by_id = FALSE. If TRUE, the function provides an additional column indicating whether the match is with swapped alleles. If check_strand_flip = TRUE, the function also provides an additional column indicating whether the match is with flipped strand. Unnecessary for gene-based tests in this package, but may be useful for other purposes (e.g., harmonization for meta-analysis that needs to flip the sign of beta for a match with swapped alleles).

Details

Pre-processing of GWAS summary data is required because the sets of variants available in a particular GWAS might be poorly matched to the variants in reference data. SNP matching can be performed either 1) by SNP ID or 2) by chromosome code, base-pair position, and allele codes, while taking into account possible strand flips and reference allele swap. For matched entries, the SNP IDs in GWAS summary data are replaced with the ones in the reference data.

Value

A data frame with columns: "id", "chr", "pos", "A1", "A2" and "pvalue". If return_indice = TRUE, the data frame includes additional columns key_, swapped_, and flipped_. key_ is "chr_pos_A1_A2" in sumstat (the original input before harmonization). swapped_ contains a logical vector indicating reference allele swap. flipped_ contains a logical vector indicating strand flip.

Examples

```r
## GWAS summary statistics
head(exGWAS)

## Load reference genotype data
bfile <- system.file("extdata", "example.bed", package = "snpsettest")
x <- read_reference_bed(path = bfile)

## Harmonize by SNP IDs
hsumstats1 <- harmonize_sumstats(exGWAS, x)

## Harmonize by genomic position and allele codes
## Reference allele swap will be taken into account
```
map_snp_to_gene

Map SNPs to genes

Description
Annotate SNPs onto their neighboring genes (or arbitrary genomic regions) to perform set-based
association tests.

Usage
map_snp_to_gene(
  info_snp,
  info_gene,
  extend_start = 20L,
  extend_end = 20L,
  only_sets = FALSE
)

Arguments
info_snp  A data frame with columns: "id", "chr", and "pos".
  • id = SNP ID (e.g., rs numbers)
  • chr = chromosome
  • pos = base-pair position
info_gene  A data frame with columns: "gene.id", "chr", "start", and "end".
  • gene.id = gene ID (or identifier for genomic regions)
  • chr = chromosome (must be the same chromosome coding scheme in info_snp)
  • start = genomic start position
  • end = genomic end position
If a gene has multiple intervals, SNPs mapped to any of them will be merged into a single set. Please assign unique IDs if you don’t want this behavior.
extend_start  A single non-negative integer, allowing for a certain kb window before the gene
to be included. Default is 20 (= 20kb).
extend_end  A single non-negative integer, allowing for a certain kb window after the gene
to be included. Default is 20 (= 20kb).
only_sets  If TRUE, only sets of SNPs for individual genes are returned. Otherwise, both
sets and mapping information are returned. Default is FALSE.
**Value**

A nested list containing following components:

- sets: a named list where each index represents a separate set of SNPs
- map: a data frame containing SNP mapping information

**Examples**

```r
## GWAS summary statistics
head(exGWAS)

## Gene information data
head(gene.curated.GRCh37)

## Map SNPs to genes
snp_sets <- map_snp_to_gene(exGWAS, gene.curated.GRCh37)

## Better to use harmonized GWAS data for gene mapping
bfile <- system.file("extdata", "example.bed", package = "snpsettest")
x <- read_reference_bed(path = bfile)
hsumstats <- harmonize_sumstats(exGWAS, x)
snp_sets <- map_snp_to_gene(hsumstats, gene.curated.GRCh37)
```

---

**Description**

Create a `bed.matrix` object from a .bed file. The function expects .fam and .bim files under the same directory. See `gaston::read.bed.matrix` for more details.

**Usage**

```r
read_reference_bed(path, ...)
```

**Arguments**

- `path`: A path to the .bed file
- `...`: Further arguments used in `gaston::read.bed.matrix`

**Value**

A `gaston::bed.matrix` object with a Z-standardized genotype matrix
Examples

```r
## Get a path to the example .bed file
bfile <- system.file("extdata", "example.bed", package = "snpsettest")

## Read a .bed file
x <- read_reference_bed(path = bfile)
```

snpset_test  |  Set-based association tests

Description

Perform set-based association tests between multiple sets of SNPs and a phenotype using GWAS summary statistics. If the function encounters missing genotypes in the reference data, they will be imputed with genotype means.

Usage

```r
snpset_test(hsumstats, x, snp_sets, method = c("saddle", "davies"))
```

Arguments

- `hsumstats`: A data frame processed by `harmonize_sumstats`.
- `x`: A `bed.matrix` object created from the reference data.
- `snp_sets`: A named list where each index represents a separate set of SNPs.
- `method`: A method to compute a set-level p value. "saddle" uses Kuonen's saddlepoint approximation (1999) and "davies" uses the algorithm of Davies (1980). When "davies" method failed to produce a meaningful result, "saddle" method is used as a fallback. Default is "saddle".

Value

A data.table with columns: "set.id", "pvalue", "n.snp", "top.snp.id" and "top.snp.pvalue"

- `set.id`: a name of SNP set
- `tstat`: a test statistic
- `pvalue`: a set-level p value
- `n.snp`: the number of SNPs used in a test
- `top.snp.id`: SNP ID with the smallest p-value within a set of SNPs
- `top.snp.pvalue`: The smallest p-value within a set of SNPs

References


Examples

```r
## GWAS summary statistics
head(exGWAS)

## Load reference genotype data
bfile <- system.file("extdata", "example.bed", package = "snpsettest")
x <- read_reference_bed(path = bfile)

## GWAS harmonization with reference data
hsumstats <- harmonize_sumstats(exGWAS, x)

## Perform a set-based test with an arbitrary SNP set
snpset_test(hsumstats, x, list(test = c("SNP_880", "SNP_1533", "SNP_4189")))

## Gene information data
head(gene.curated.GRCh37)

## Map SNPs to genes
snp_sets <- map_snp_to_gene(hsumstats, gene.curated.GRCh37)

## Perform gene-based association tests
out <- snpset_test(hsumstats, x, snp_sets$sets)
```
Index

* datasets
  - exGWAS, 2
  - gene.curated.GRCh37, 3
  - gene.curated.GRCh38, 3

  exGWAS, 2
  gaston::bed.matrix, 7
  gaston::read.bed.matrix, 7
  gene.curated.GRCh37, 3
  gene.curated.GRCh38, 3

  harmonize_sumstats, 4, 8
  map_snp_to_gene, 6
  read_reference_bed, 7
  snpset_test, 8