

# Package ‘hierGWAS’

December 18, 2025

**Title** Assessing statistical significance in predictive GWA studies

**Version** 1.41.0

**Author** Laura Buzdugan

**Maintainer** Laura Buzdugan <buzdugan@stat.math.ethz.ch>

**Description** Testing individual SNPs, as well as arbitrarily large groups of SNPs in GWA studies, using a joint model of all SNPs. The method controls the FWER, and provides an automatic, data-driven refinement of the SNP clusters to smaller groups or single markers.

**Depends** R (>= 3.2.0)

**License** GPL-3

**LazyData** true

**Imports** fastcluster,glmnet, fmsb

**Suggests** BiocGenerics, RUnit, MASS

**biocViews** SNP, LinkageDisequilibrium, Clustering

**Collate** 'cluster.snp.R' 'lasso.select.R' 'multisplit.R' 'MEL.R'  
'test.snp.R' 'adj.pval.R' 'comp.cluster.pval.R'  
'iterative.DFS.R' 'test.hierarchy.R' 'return.r2.R'  
'compute.r2.R'

**git\_url** <https://git.bioconductor.org/packages/hierGWAS>

**git\_branch** devel

**git\_last\_commit** 76af363

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-12-17

## Contents

|                          |   |
|--------------------------|---|
| cluster.snp . . . . .    | 2 |
| compute.r2 . . . . .     | 3 |
| hierGWAS . . . . .       | 4 |
| multisplit . . . . .     | 4 |
| simGWAS . . . . .        | 5 |
| test.hierarchy . . . . . | 6 |

## Index

8

---

|             |                                            |
|-------------|--------------------------------------------|
| cluster.snp | <i>Hierarchical Clustering of SNP Data</i> |
|-------------|--------------------------------------------|

---

**Description**

Clusters SNPs hierarchically.

**Usage**

```
cluster.snp(x = NULL, d = NULL, method = "average", SNP_index = NULL)
```

**Arguments**

|           |                                                                                                                                                                                                                                                                       |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x         | The SNP data matrix of size nobs x nvar. Default value is NULL                                                                                                                                                                                                        |
| d         | NULL or a dissimilarity matrix. See the 'Details' section.                                                                                                                                                                                                            |
| method    | The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). See <a href="#">hclust</a> for details. |
| SNP_index | NULL or the index vector of SNPs to be clustered. See the 'Details' section.                                                                                                                                                                                          |

**Details**

The SNPs are clustered using [hclust](#), which performs a hierarchical cluster analysis using a set of dissimilarities for the nvar objects being clustered. There are 3 possible scenarios.

If d = NULL, x is used to compute the dissimilarity matrix. The dissimilarity measure between two SNPs is 1 - LD (Linkage Disequilibrium), where LD is defined as the square of the Pearson correlation coefficient. If SNP\_index = NULL, all nvar SNPs will be clustered; otherwise only the SNPs with indices specified by SNP\_index will be considered.

If the user wishes to use a different dissimilarity measure, d needs to be provided. d must be either a square matrix of size nvar x nvar, or an object of class [dist](#). If d is provided, x and SNP\_index will be ignored.

**Value**

An object of class [dendrogram](#) which describes the tree produced by the clustering algorithm [hclust](#).

**Examples**

```
library(MASS)
x <- mvrnorm(60, mu = rep(0, 60), Sigma = diag(60))
clust.1 <- cluster.snp(x = x, method = "average")
SNP_index <- seq(1, 10)
clust.2 <- cluster.snp(x = x, method = "average", SNP_index = SNP_index)
d <- dist(x)
clust.3 <- cluster.snp(d = d, method = "single")
```

---

|            |                       |
|------------|-----------------------|
| compute.r2 | <i>R2 computation</i> |
|------------|-----------------------|

---

**Description**

Calculates the R2 of a cluster of SNPs.

**Usage**

```
compute.r2(x, y, res.multisplit, covar = NULL, SNP_index = NULL)
```

**Arguments**

|                |                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------|
| x              | The input matrix, of dimension nobs x nvar. Each row represents a subject, each column a SNP.         |
| y              | The response vector. It can be continuous or discrete.                                                |
| res.multisplit | The output of multisplit.                                                                             |
| covar          | NULL or the matrix of covariates one wishes to control for, of size nobs x ncovar.                    |
| SNP_index      | NULL or the index vector of the cluster of SNPs whose R2 will be computed. See the 'Details' section. |

**Details**

The R2 of a cluster of SNPs is computed on the second half-samples. The cluster members, are intersected with the SNPs selected by the lasso, and the R2 of this model is calculated. Thus, we obtain B R2 values. Finally, the mean of these values is taken. If the value of SNP\_index is NULL, the R2 of the full model with all the SNPs will be computed.

**Value**

The R2 value of the SNP cluster

**References**

Buzdugan, L. et al. (2015), Assessing statistical significance in predictive genome-wide association studies. (unpublished)

**Examples**

```
library(MASS)
x <- mvrnorm(60, mu = rep(0, 60), Sigma = diag(60))
beta <- rep(0, 60)
beta[c(5, 9, 3)] <- 1
y <- x %*% beta + rnorm(60)
SNP_index <- c(5, 9, 3)
res.multisplit <- multisplit(x, y)
r2 <- compute.r2(x, y, res.multisplit, SNP_index = SNP_index)
```

---

 hierGWAS

*Assessing statistical significance in predictive GWA studies*


---

### Description

Testing individual SNPs, as well as arbitrarily large groups of SNPs in GWA studies, using a joint model of all SNPs. The method controls the FWER, and provides an automatic, data-driven refinement of the SNP clusters to smaller groups or single markers.

### Details

hierGWAS is a package designed to assess statistical significance in GWA studies, using a hierarchical approach.

There are 4 functions provided: `cluster.snp`, `multisplit`, `test.hierarchy` and `compute.r2`. `cluster.snp` performs the hierarchical clustering of the SNPs, while `multisplit` runs multiple penalized regressions on repeated random subsamples. These 2 functions need to be executed before `test.hierarchy`, which does the hierarchical testing, though the order in which the 2 functions are executed does not matter. `test.hierarchy` provides the final output of the method: a list of SNP groups or individual SNPs, along with their corresponding p-values. Finally, `compute.r2` computes the explained variance of an arbitrary group of SNPs, of any size. This group can encompass all SNPs, SNPs belonging to a certain chromosome, or an individual SNP.

### Author(s)

Laura Buzdugan [laura.buzdugan@stat.math.ethz.ch](mailto:laura.buzdugan@stat.math.ethz.ch)

### References

Buzdugan, L. et al. (2015), Assessing statistical significance in predictive genome-wide association studies (unpublished)

---

 multisplit

*Variable Selection on Random Sample Splits.*


---

### Description

Performs repeated variable selection via the lasso on random sample splits.

### Usage

```
multisplit(x, y, covar = NULL, B = 50)
```

**Arguments**

|                    |                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>     | The SNP data matrix, of size <code>nobs</code> x <code>nvar</code> . Each row represents a subject, each column a SNP. |
| <code>y</code>     | The response vector. It can be continuous or discrete.                                                                 |
| <code>covar</code> | NULL or the matrix of covariates one wishes to control for, of size <code>nobs</code> x <code>ncovar</code> .          |
| <code>B</code>     | The number of random splits. Default value is 50.                                                                      |

**Details**

The samples are divided into two random splits of approximately equal size. The first subsample is used for variable selection, which is implemented using [glmnet](#). The first `[nobs/6]` variables which enter the lasso path are selected. The procedure is repeated `B` times.

If one or more covariates are specified, these will be added unpenalized to the regression.

**Value**

A data frame with 2 components. A matrix of size `B` x `[nobs/2]` containing the second subsample of each split, and a matrix of size `B` x `[nobs/6]` containing the selected variables in each split.

**References**

Meinshausen, N., Meier, L. and Buhlmann, P. (2009), P-values for high-dimensional regression, Journal of the American Statistical Association 104, 1671-1681.

**Examples**

```
library(MASS)
x <- mvrnorm(60,mu = rep(0,200), Sigma = diag(200))
beta <- rep(1,200)
beta[c(5,9,3)] <- 3
y <- x %*% beta + rnorm(60)
res.multisplit <- multisplit(x, y)
```

---

simGWAS

---

*Simulated GWAS data*


---

**Description**

This data set was simulated using PLINK. Please refer to the vignette for more details.

**Usage**

```
simGWAS
```

**Format**

The dataset contains the following components:

SNP.1 The first SNP, of dimension 500 x 1. Each row represents a subject.

...

SNP.1000 The last SNP, of dimension 500 x 1. Each row represents a subject.

y The response vector. It can be continuous or discrete.

sex The first covariate, representing the sex of the subjects: 0 for men and 1 for women.

age The second covariate, representing the age of the subjects.

**Value**

data.frame

**Examples**

```
data(simGWAS)
```

---

test.hierarchy

*Hierarchical Testing of SNPs*


---

**Description**

Performs hierarchical testing of SNPs.

**Usage**

```
test.hierarchy(x, y, dendr, res.multisplit, covar = NULL, SNP_index = NULL,
  alpha = 0.05, global.test = TRUE, verbose = TRUE)
```

**Arguments**

|                |                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------|
| x              | The input matrix, of dimension nobs x nvar. Each row represents a subject, each column a SNP.                   |
| y              | The response vector. It can be continuous or discrete.                                                          |
| dendr          | The cluster tree obtained by hierchically clustering the SNPs using cluster.snp.                                |
| res.multisplit | The output of multisplit.                                                                                       |
| covar          | NULL or the matrix of covariates one wishes to control for, of size nobs x ncovar.                              |
| SNP_index      | NULL or the index vector of SNP to be tested. See the 'Details' section.                                        |
| alpha          | The significance level at which the FWER is controlled. Default value is 0.05.                                  |
| global.test    | Specifies wether the global null hypothesis should be tested. Default value is TRUE. See the 'Details' section. |
| verbose        | Report information on progress. Default value is TRUE                                                           |

## Details

The testing is performed on the cluster tree given by `dendr`. If the SNP data matrix was divided (e.g. by chromosome), and clustered separately, the user must provide the argument `SNP_index`, to specify which part of the data is being tested.

Testing starts at the highest level, which includes all variables specified by `SNP_index`, and moves down the cluster tree. It stops when a cluster's null hypothesis cannot be rejected anymore. The smallest, still significant clusters will be returned.

By default the parameter `global.test = TRUE`, which means that first the global null hypothesis is tested. If the data is divided (e.g. by chromosome), and clustered separately, this parameter can be set to `FALSE` once the global null has been rejected. This helps save time.

## Value

A list of significant SNP groups with the following components:

|                        |                                      |
|------------------------|--------------------------------------|
| <code>SNP_index</code> | The indices of the SNPs in the group |
| <code>pval</code>      | The p-value of the SNP group         |

## References

Buzdugan, L. et al. (2015), Assessing statistical significance in predictive genome-wide association studies

## Examples

```
library(MASS)
x <- mvrnorm(60, mu = rep(0, 60), Sigma = diag(60))
beta <- rep(0, 60)
beta[c(5, 9, 3)] <- 1
y <- x %*% beta + rnorm(60)
dendr <- cluster.snp(x = x, method = "average")
res.multisplit <- multisplit(x, y)
sign.clusters <- test.hierarchy(x, y, dendr, res.multisplit)
```

# Index

## \* **datasets**

simGWAS, [5](#)

cluster.snp, [2](#), [4](#)

compute.r2, [3](#), [4](#)

dendrogram, [2](#)

dist, [2](#)

glmnet, [5](#)

hclust, [2](#)

hierGWAS, [4](#)

hierGWAS-package (hierGWAS), [4](#)

multisplit, [4](#), [4](#)

simGWAS, [5](#)

test.hierarchy, [4](#), [6](#)