

# Package ‘genomicper’

May 8, 2026

**Type** Package

**Title** Circular Genomic Permutation using Genome Wide Association  
p-Values

**Version** 1.8

**Date** 2026-02-21

**Maintainer** Claudia P Cabrera <c.cabrera@qmul.ac.uk>

**Imports** stats,grDevices,utils,graphics

## Description

Circular genomic permutation approach uses genome wide association studies (GWAS) results to establish the significance of pathway/gene-set associations whilst accounting for genomic structure. All single nucleotide polymorphisms (SNPs) in the GWAS are placed in a 'circular genome' according to their location. Then the complete set of SNP association p-values are permuted by rotation with respect to the SNPs' genomic locations. Two testing frameworks are available: permutations at the gene level, and permutations at the SNP level. The permutation at the gene level uses Fisher's combination test to calculate a single gene p-value, followed by the hypergeometric test. The SNP count methodology maps each SNP to pathways/gene-sets and calculates the proportion of SNPs for the real and the permuted datasets above a pre-defined threshold. Genomicper requires a matrix of GWAS association p-values and SNPs annotation to genes. Pathways can be obtained from within the package or can be provided by the user. Cabrera et al (2012) <[doi:10.1534/g3.112.002618](https://doi.org/10.1534/g3.112.002618)> .

**License** GPL-2

**NeedsCompilation** no

**Author** Claudia P Cabrera [aut, cre],  
Pau Navarro [aut],  
Chris S Haley [aut]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2026-02-23 16:10:17 UTC

## Contents

genomicper-package . . . . . 2

demo . . . . .	4
genes_permutation . . . . .	5
genome_order . . . . .	7
get_results . . . . .	8
hyprbg . . . . .	10
plot_results . . . . .	11
read2_paths . . . . .	12
read_pvals . . . . .	14
RHSAXXXX . . . . .	16
SNPsAnnotation . . . . .	16
snps_permutation . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

genomicper-package	<i>Circular Genomic Permutations</i>
--------------------	--------------------------------------

---

## Description

Description: Circular genomic permutation approach uses genome wide association studies (GWAS) results to establish the significance of pathway/gene-set associations whilst accounting for genomic structure. All single nucleotide polymorphisms (SNPs) in the GWAS are placed in a 'circular genome' according to their location. Then the complete set of SNP association p-values are permuted by rotation with respect to the SNPs' genomic locations. Two testing frameworks are available: permutations at the gene level, and permutations at the SNP level. The permutation at the gene level uses Fisher's combination test to calculate a single gene p-value, followed by the hypergeometric test. The SNP count methodology maps each SNP to pathways/gene-sets and calculates the proportion of SNPs for the real and the permuted datasets above a pre-defined threshold. Genomicper requires a matrix of GWAS association p-values and SNPs annotation to genes. Pathways can be obtained from within the package or can be provided by the user.

## Details

Package: genomicper  
 Type: Package  
 Version: 1.8  
 Date: 2026-02-21  
 License: GPL-2

## Author(s)

Claudia P. Cabrera, Pau Navarro, Chris S. Haley  
 Maintainer: Claudia Cabrera <c.cabrera@qmul.ac.uk>

## References

SNP-level Permutations:

Genomicper: genome-wide association SNP-set analysis

Claudia P. Cabrera\*, Pau Navarro\*, Jennifer E. Huffman, Alan F. Wright, Caroline Hayward, Harry Campbell, James F. Wilson, Igor Rudan, Nicholas D. Hastie, Veronique Vitart, Chris S. Haley\*

Gene-level Permutations:

Uncovering Networks from Genome-Wide Association Studies via

Circular Genomic Permutation. G3: Genes|Genomes|Genetics 2, 1067-1075.

Claudia P. Cabrera\*, Pau Navarro\*, Jennifer E. Huffman, Alan F. Wright, Caroline Hayward, Harry Campbell, James F. Wilson, Igor Rudan, Nicholas D. Hastie, Veronique Vitart, Chris S. Haley\*

## See Also

Genomicper functions: 1) [read\\_pvals](#), 2) [genome\\_order](#), 3) [read2\\_paths](#), 4A) [snps\\_permutation](#), 4B) [genes\\_permutation](#), 5) [get\\_results](#), 6) [plot\\_results](#)

## Examples

```
#####
# Genomicper functions #####
# 1) read_pvals(data_name="", snps_ann="")
# 2) genome_order(all_data="")
# 3) read2_paths(ordered_alldata="", gs_locs="", sets_from="", sets_prefix="RHSA", level="")
# 4A) snps_permutation(ordered_alldata="", pers_ids="", ntraits="", nper="", saveto="",
# threshold="", gs_locs=gs_locs, envir = "")
# 4B) genes_permutation(ordered_alldata="", pers_ids="", pathways="",
# ntraits="", nper="", threshold="", saveto="", gs_locs=gs_locs, envir = "")
# 5) get_results(res_pattern="Perm", level="snp", from="workspace",
# threshold=0.05, envir = "")
# 6) plot_results(results = "", by = "", plot_all = TRUE, var = "", save_plot = TRUE,
# plot_name = "", bf = FALSE, save_qq = TRUE)
#####
##### DEMO: #####

#### SNP-level #####
# SNPs annotation and Pathways provided by user
# all data stored at the WORKSPACE

### Load files for analysis
data(demo, SNPsAnnotation)

# Read & format GWAS pvalues
all_data <- read_pvals(data_name=demo, snps_ann=SNPsAnnotation)

# Order data according to the genome
genome_results <- genome_order(all_data=all_data)
# Results from genome_order
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs
```

```

# Create new environment to save variables (e.g. pathways, permutations):
gper.env <- new.env()

# Load example pathways into the new environment.
data(RHSA164843,RHSA446343,RHSA8876384,RHSA8964572,RHSA109582,RHSA1474244,envir=gper.env)

# Map SNPs to pathways
paths_res <- read2_paths(ordered_alldata=ordered_alldata,
gs_locs=gs_locs,sets_from="workspace",sets_prefix="RHSA",
level="snp",envir=gper.env)
# Results from read2_paths:
pers_ids <- paths_res$per_ors
pathways<- paths_res$pathways

# Perform permutations:
snps_permutation(ordered_alldata=ordered_alldata,
pers_ids=pers_ids,ntraits=c(7:13),nper=10,saveto="workspace",
threshold=0.05,gs_locs=gs_locs,envir = gper.env)

# Get results
results <- get_results(res_pattern="Permus",level="snp",
from="workspace",threshold=0.05,envir = gper.env)

# Plot results
## Not run:
#saves plots to working directory
qq <- plot_results(results=results,by="set",plot_all=TRUE)
qq <- plot_results(results=results,by="trait",
plot_all=FALSE,var="trait1")
# Displays interactive plot. Select a trait/set to plot and
# set arguments save_plot=FALSE, plot_all = FALSE
# IMPORTANT: to EXIT interactive plot, RIGHT CLICK on the
# plot and STOP.
qq <- plot_results(results=results,by="set",plot_all=FALSE,
var="RHSA109582",save_plot=FALSE)

## End(Not run)
# -- END OF DEMO
#####

```

---

demo

*GWAS p\_values demo data*

---

## Description

GWAS p-values (tab delimited file). First Column must contain the SNP ids and the column name = "name"

## Usage

data(demo)

**Format**

A data frame with SNPs identifiers and gwas p-values of association

name a character vector

Trait1 a numeric vector

Trait2 a numeric vector

Trait3 a numeric vector

Trait4 a numeric vector

Trait5 a numeric vector

Trait6 a numeric vector

Trait7 a numeric vector

Trait8 a numeric vector

Trait9 a numeric vector

name	Trait1	Trait2	Trait3	Trait4	Trait5	Trait6
rs10000010	0.9122360	0.30088096	0.2332038	0.5193068	0.1255104	0.07253145
rs10000023	0.8642906	0.52064064	0.9243443	0.7177759	0.9512171	0.81716250
rs10000030	0.2832705	0.99021664	0.8359339	0.9662707	0.8491221	0.50208681

**Examples**

```
#Read input demo file for "read_pvals" function
data(demo)
```

---

genes\_permutation      *Gene-level Permutations*

---

**Description**

Performs gene-level circular genomic permutations. In each permutation, the complete set of SNP association p-values are permuted by rotation with respect to the SNPs' genomic locations. Once these 'simulated' p-values are assigned, the joint gene p-values are calculated using Fisher's combination test, and pathways' association tested using the hypergeometric test

**Usage**

```
genes_permutation(ordered_alldata = "", pers_ids = "", pathways = "",
ntraits = "", nper = 100, threshold = 0.05, seed=10, saveto = "workspace",
gs_locs="", envir = "")
```

**Arguments**

ordered_alldata	Return variable from "genome_order". Ordered genome and trait p-values
gs_locs	Return variable from "genome_order". SNP indexes
pers_ids	Return variable "per_ors" from "read2_paths". Gene indexes
pathways	Return variable "pathways" from "read2_paths"
ntraits	Traits INDEX to be analysed. Index according to "ordered_alldata". Trait Columns index must start at 7. Example: ntraits=c(7:9), ntraits=7
nper	Number of permutations.Example: nper=1000
threshold	Threshold to be set by the hypergeometric test. threshold=0.05
seed	Set a number for random sampling
saveto	Save permutation results to "workspace" OR "directory"
envir	R environment to save the data to when saveto is set to "workspace"

**Value**

Returns "Permus\_trait" variables or files (permutation datasets).

**References**

Imports phyper (from stats)

**See Also**

[snps\\_permutation](#)

**Examples**

```
#load data
data(demo,SNPsAnnotation)
all_data <- read_pvals(data_name=demo,snps_ann=SNPsAnnotation)

# Prepare Genome
genome_results <-genome_order(all_data=all_data)
# Results from genome_order
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs

# Create new environment to save data:
gper.env <- new.env()

# Get pathways
data(RHSA164843,RHSA446343,RHSA8876384,RHSA8964572,RHSA109582,RHSA1474244,envir=gper.env)

# Map Genes to pathways
paths_res <- read2_paths(ordered_alldata=ordered_alldata,gs_locs=gs_locs,
sets_from="workspace",sets_prefix="RHSA",level="gene",envir=gper.env)
pers_ids <- paths_res$per_ors
```

```

pathways<- paths_res$pathways

# Perform Permutations:
genes_permutation(ordered_alldata=ordered_alldata,
pers_ids=pers_ids,pathways=pathways,ntraits=c(7:9),
nper=10,threshold=0.05, saveto="workspace",
gs_locs=gs_locs,envir = gper.env)

# Results
results <- get_results(res_pattern="Permus",level="gene",
from="workspace",threshold=0.05,envir= gper.env)

```

---

genome\_order                      *Genome Order*

---

## Description

Orders the SNPs according to their genomic location

## Usage

```
genome_order(all_data = "")
```

## Arguments

all\_data                      SNPs to Genes Annotation and Trait Pvalues of Association  
all\_data = (read\_pvals output) OR matrix/dataframe.

## Details

Input Columns with "\*" must be included for analysis

NOTE: Trait p-values must start at Column #7

```

# *Column 1: "name" (SNP_IDs - any SNP ID as character)
# *Column 2: Chromosome Location
# *Column 3: SNP Location
# *Column 4: Gene ID
# Column 5: Symbol (OR Annotation Field 1)
# Column 6: Annotaiton Field 2
# *Column 7: First trait pvalues of association
# Column N: Next trait pvalues of association
# Example Input Data:
name            Chromosome   Location   GENE_ID   Symbol   Orientation   abpi
rs10000010        4   21618674    80333   KCNIP4            -   0.91
rs10000023        4   95733906     658   BMPR1B            +   0.86
rs10000092        4   21895517    80333   KCNIP4            -   0.20
rs1000022        13   100461219   171425   CLYBL            +   0.26
rs10000300        4   40466547    54502   RBM47            -   0.58

```

**Value**

ordered\_alldata           SNPs annotated to Genes and Trait p-values

gs\_locs                   Gene annotations, location indexes and number of observations

**Format**

SNPs annotated to Genes and Trait p-values

```
#ordered_alldata[1:5,1:8]
name  Chromosome Location GENE_ID  Symbol Orientation Trait1 Trait2
rs3934834 1 1005806 NA <NA> <NA> 0.97 0.92
rs3737728 1 1021415 54991 C1orf159 - 0.91 0.69
rs6687776 1 1030565 54991 C1orf159 - 0.71 0.45
rs9651273 1 1031540 54991 C1orf159 - 0.22 0.60
rs4970405 1 1048955 54991 C1orf159 - 0.77 0.56
```

Gene annotations, location indexes and number of observations

```
#gs_locs[1:5,]
# Symbol Chromosome Location Gene_ID Start_Indx Observations
# [1,] "A1BG" "19" "58864479" "1" "293976" "1"
# [2,] "A2M" "12" "9232268" "2" "215264" "5"
# [3,] "NAT1" "8" "18077310" "9" "151804" "1"
# [4,] "NAT2" "8" "18257280" "10" "151831" "2"
# [5,] "SERPINA3" "14" "95080803" "12" "249519" "2"
```

**See Also**

[read2\\_paths](#)

**Examples**

```
## DEMO WORKSPACE

data(demo, SNPsAnnotation)
all_data<-read_pvals(data_name=demo, snps_ann=SNPsAnnotation)
# GENOME ORDER
genome_results <- genome_order(all_data=all_data)
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs
```

---

get\_results

*Circular Permutation Results*

---

**Description**

Creates a summary dataframe of the genomic permutations datasets

**Usage**

```
get_results(res_pattern="Permus", level="snp", from="workspace",
threshold=0.05, envir = "")
```

**Arguments**

res_pattern	Pattern of the Permutation files/variable. eg. res=pattern="Permus"
level	Permutation level performed.level values "snp" or "gene"
from	Location of the permutation datasets.from values "workspace" or "directory"
threshold	Threshold of significance set
envir	R environment where save the data to

**Value**

results	Data frame with Pathway ID, Trait, Threshold set by permutations, Gene results include the theoretical hypergeometric p-value and the, observed (Empirical Hypergeometric p-values) SNP results include the count of significant SNPs and the overall score Score is the proportion of tests observed with more significant results
---------	---

**Format**

```
## SNP level results
  PathID  Trait Threshold RealCount Score
1 hsa00010  abpi         0         0 0.037
2 hsa00010 abpildfa      0         0 0.040
3 hsa04720  abpi         2         0 0.311
## Gene level results
  PathID Trait  Threshold  P-Value  Observed
1 hsa00010  abpi 0.040441176 0.058823529 1.0000000
2 hsa00020  abpi 0.000000000 0.000000000 0.1666667
3 hsa00030  abpi 0.040441176 0.058823529 1.0000000
```

**Examples**

```
data(demo,SNPsAnnotation)
all_data <- read_pvals(data_name=demo,snps_ann=SNPsAnnotation)
genome_results <-genome_order(all_data=all_data)
# Results from genome_order
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs

# Create new environment to save data
gper.env <- new.env()

# Get pathways
data(RHSA164843,RHSA446343,RHSA8876384,RHSA8964572,RHSA109582,
RHSA1474244,envir=gper.env)
```

```
paths_res <- read2_paths(ordered_alldata=ordered_alldata,gs_locs=gs_locs,
sets_from="workspace",sets_prefix="RHSA",level="snp",envir=gper.env)
pers_ids <- paths_res$per_ors
pathways<- paths_res$pathways

snps_permutation(ordered_alldata=ordered_alldata,pers_ids=pers_ids,
ntraits=c(7,9),nper=10,saveto="workspace",threshold=0.05,
gs_locs=gs_locs,envir= gper.env)

results <- get_results(res_pattern="Permus",level="snp",
from="workspace",threshold=0.05,envir = gper.env)
```

---

hyprbg

*Hypergeometric Test (phyper)*

---

### **Description**

Performs Hypergeometric test (phyper() from R)

### **Usage**

```
hyprbg(Sig_in_Paths, uniSig, gns_in_Paths, universe)
```

### **Arguments**

Sig_in_Paths	Number of significant genes in the pathway
uniSig	Number of significant genes in the dataset
gns_in_Paths	Number of genes in the pathway
universe	Number of genes in the dataset

### **Value**

Returns hypergeometric test

### **References**

hyprbg Imports phyper() (from stats)

---

plot_results	<i>Plot Results Circular Permutation</i>
--------------	--

---

**Description**

QQ plots

**Usage**

```
plot_results(results="",by="",plot_all=TRUE, var = "", save_plot=TRUE, plot_name="",
bf= FALSE , save_qq = TRUE)
```

**Arguments**

results	Results datarame from "get_results()"
by	Visualize results by "trait" OR by "set"
plot_all	plot_all = TRUE plots all the variables in the results dataframe and saves a pdf file in the working directory. Setting plot_all to FALSE plots a single variable(trait or set). The argument "var" must be declared.
var	Variable name to plot
save_plot	save_plot = TRUE saves the plots in the working directory. save_plot = FALSE the plot is visualized at the console. save_plot = FALSE can be used only when plot_all is set to FALSE. The plot displayed at the console is interactive, clicking on a point displays the points name.
plot_name	Argument used to save the file name for the plots. Default value = Results_genomicper_[set/trait]
bf	Displays the bonferroni correction
save_qq	TRUE returns the qq plot values

**Value**

qq	Data frame with qq plot values
----	--------------------------------

**See Also**

[get\\_results](#)

**Examples**

```
data(demo,SNPsAnnotation)
all_data <- read_pvals(data_name=demo,snps_ann=SNPsAnnotation)
genome_results <-genome_order(all_data=all_data)
# Results from genome_order
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs
```

```

# Create new environment to save the data:
gper.env <- new.env()

# Load Pathways
data(RHSA164843,RHSA446343,RHSA8876384,RHSA8964572,RHSA109582,RHSA1474244,envir=gper.env)

paths_res <- read2_paths(ordered_alldata=ordered_alldata,gs_locs=gs_locs,
sets_from="workspace",sets_prefix="RHSA",level="snp",envir=gper.env)
pers_ids <- paths_res$per_ors
pathways<- paths_res$pathways

snps_permutation(ordered_alldata=ordered_alldata,pers_ids=pers_ids,
ntraits=c(7,9),nper=10,saveto="workspace",threshold=0.05,
gs_locs=gs_locs,envir = gper.env)

results <- get_results(res_pattern="Permus",level="snp",
from="workspace",threshold=0.05,envir = gper.env)

#saves plots to working directory
## Not run:
qq <- plot_results(results=results,by="set",plot_all=TRUE)
qq <- plot_results(results=results,by="trait",plot_all=FALSE,var="trait1")
qq <- plot_results(results=results,by="set",
plot_all=FALSE,var="R-HSA-8964572",
save_plot=FALSE) ## IMPORTANT: to EXIT interactive plot
## right click on the plot to stop

## End(Not run)

```

---

read2\_paths

*Read to SNPs to sets; Map SNPs to gene-sets/pathways*


---

## Description

Reads the sets/pathways, map the SNPs and genes to the gene-sets/pathways read2\_paths uses the "genome\_order" output(ordered\_alldata, gs\_locs) to assign genomic location indexes to each element in the gene-set. The permutation method must be defined (i.e. level = "snp" OR level = "gene").

## Usage

```
read2_paths(ordered_alldata="",gs_locs="",sets_from="workspace",
sets_prefix="RHSA",level="snp",envir="")
```

## Arguments

ordered\_alldata

Ordered data according to the SNPs genomic location. Traits start at column 7

Return variable from:

```

genome_results <-genome_order(all_data=all_data)
ordered_alldata <- genome_results$ordered_alldata

gs_locs      Gene annotation,indexes and number of observations
              Return variable from genome_order():
              genome_results <-genome_order(all_data=all_data)
              gs_locs <- genome_results$gs_locs

sets_from    Location of the gene-sets. Default set to "workspace"
              sets_from="workspace" OR sets_from="directory"
              "directory", only will search for information in the working directory.

sets_prefix  Prefix of the gene-set variables or files.
              Default set to sets_prefix= "RHSA" e.g. Variables "RHSA164843","RHSA446343","RHSA8876384"
              each variable/file contains the list of gene identifiers part of that pathway

level       The level at which the permutations will be performed. Assigns the indexes
              according to snps or genes
              Default value "snp" level values = "snp" OR "gene"

envir       R environment where pathway data is stored. e.g(envir=.GlobalEnv, envir=gper.env)

```

**Value**

```

pathways    Pathway Id, Description,Number of Genes in the pathway, Number of genes
              found in the dataset, Number of SNPs found in the dataset

per_ors     A list of identifiers mapped to each pathway

```

**Format**

```

Input: Ordered_alldata
name      Chromosome  Location GENE_ID  Symbol Orientation Trait1 Trait2
rs1001567      1 9194614   <NA>    <NA>    <NA> 0.96 0.89
rs1000313      1 15405489  23254 KIAA1026    + 0.93 0.57
rs1002365      1 19797248  <NA>    <NA>    <NA> 0.68 0.58
rs1002706      1 25051153  <NA>    <NA>    <NA> 0.71 0.02
rs1002487      1 26865971  6195   RPS6KA1    + 0.98 0.78

```

```

Input:gs_locs
      Symbol  Chromosome Location  Gene_ID Start_Indx Observations
[1,] "ACYP2"  "2"      "54399633" "98"    "35"    "1"
[2,] "AMPD3"  "11"     "10514707" "272"   "898"   "1"
[3,] "ANK2"   "4"      "113830885" "287"   "479"   "4"

```

```

Input:pathway example
RHSA8964572
[1] 1149 128486 161247 29923 345275 63924

```

```

Output:pathways
      ID          GenesInPath GenesFound SNPsInPath

```

```

"RHSA109582" "681" "8" "11"
"RHSA1474244" "418" "7" "10"
"RHSA164843" "11" "0" "0"
"RHSA446343" "4" "1" "1"
"RHSA8876384" "32" "1" "1"
"RHSA8964572" "6" "1" "1"

```

## See Also

[genes\\_permutation](#) [snps\\_permutation](#) [genome\\_order](#)

## Examples

```

## DEMO - SNP Level data stored in workspace #####
# library(genomicper)
data(demo, SNPsAnnotation)
all_data <- read_pvals(data_name=demo, snps_ann=SNPsAnnotation)
genome_results <- genome_order(all_data=all_data)
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs

# Create new environment to save variables (e.g. pathways, permutations):
gper.env <- new.env()

data(RHSA164843, RHSA446343, RHSA8876384, RHSA8964572, RHSA109582, RHSA1474244, envir=gper.env)

paths_res <- read2_paths(ordered_alldata=ordered_alldata,
gs_locs=gs_locs, sets_from="workspace", sets_prefix="RHSA",
level="snp", envir=gper.env)
pers_ids <- paths_res$per_ors
pathways <- paths_res$pathways
#####

```

---

read\_pvals

*Read GWAS p-values of association and Merge with SNP annotations*

---

## Description

Read GWAS p-values of association and Merge with SNP annotations for analysis

## Usage

```
read_pvals(data_name="", snps_ann="", from="workspace")
```

**Arguments**

data_name	GWAS p_values (tab delimited file)(SNP_IDs Trait1 Trait2 ...TraitN)
snps_ann	SNPs Annotation (SNPsAnnotation). Genomicper uses entrez gene ids to annotate associate SNPs-to genes-pathways. The annotation MUST match your data input (coordinates and chromosome format) Any SNP ID is valid, as long the ID is set as character The examples below show an option on how to annotate the SNPs prior the use of genomicper
from	Datasets location. Values "workspace" OR "directory"

**Value**

Dataframe: name; chromosome; Location; GeneID; Symbol; Orientation; Trait1; TraitN

**Formats**

GWAS p\_values (tab delimited file)(SNP\_IDs Trait1 Trait2 ...TraitN)

name	Trait1	Trait2	TraitN
rs10000010	0.9122360	0.30088096	0.2332038
rs10000023	0.8642906	0.52064064	0.9243443
rs10000030	0.2832705	0.99021664	0.8359339

SNPs Annotation (SNPsAnnotation)

name	Chromosome	Location	GENE_ID	Symbol	Orientation
rs1000313	1	15405489	23254	KIAA1026	+
rs1000533	1	168282491	9095	TBX19	+
rs1000731	1	231963491	27185	DISC1	+

Output:

name	Chromosome	Location	GENE_ID	Symbol	Orientation	Trait1
rs10000010	4	21618674	80333	KCNIP4	-	0.9122360
rs10000023	4	95733906	658	BMPR1B	+	0.8642906
rs10000030	4	103374154	NA	<NA>	<NA>	0.2832705

**See Also**

[genome\\_order](#)

**Examples**

```
## DEMO // WORKSPACE
data(demo, SNPsAnnotation)
all_data <- read_pvals(data_name=demo, snps_ann=SNPsAnnotation)
```

RHSAXXXX

*Reactome Pathway examples*

---

**Description**

Each file "RHSAXXXX" contains the gene identifiers.

**Usage**

```
data(RHSA164843)
```

**Format**

The format is: num [1:6] 11168 155030 155348 155459 155908 2547...

**Source**

reactome.db

**Examples**

```
data(RHSA164843)
```

---

SNPsAnnotation

*SNPs-Genes annotation to Distance 0 (SNPs within a gene)*

---

**Description**

SNPs annotated to genes. Annotation only when the SNPs fall within start and end of transcription of the genes.

**Usage**

```
data(SNPsAnnotation)
```

**Format**

Sample data frame with 339096 SNP observations on the following 6 variables.

name a character vector

Chromosome a character vector

Location a numeric vector of the SNP location

GENE\_ID a numeric vector with entrez geneID

Symbol a character vector ; other annotation slot 1

Orientation a character vector; other annotation slot 2

name	Chromosome	Location	GENE_ID	Symbol	Orientation
rs1000313	1	15405489	23254	KIAA1026	+
rs1000533	1	168282491	9095	TBX19	+
rs1000731	1	231963491	27185	DISC1	+

### Source

NCBI Gene database,(<http://www.ncbi.nlm.nih.gov/gene> ; Build.37.1).

### Examples

```
data(SNPsAnnotation)
```

---

snps_permutation	<i>SNP-level permutations</i>
------------------	-------------------------------

---

### Description

Performs SNP-level circular genomic permutations. In each permutation, the complete set of SNP association p-values are permuted by rotation with respect to the SNPs' genomic locations.

Once these 'simulated' p-values are assigned, the proportion of SNPs per set above a pre-defined threshold is calculated

### Usage

```
snps_permutation(ordered_alldata = "", pers_ids = "", ntraits = "",
nper = 100, threshold = 0.05, seed=10,saveto = "workspace",
gs_locs = "",envir = "")
```

### Arguments

ordered_alldata	Return variable from "genome_order". Ordered genome and trait p-values
gs_locs	Return variable from "genome_order". SNP indexes
pers_ids	Return variable "per_ors" from "read2_paths". SNP indexes
ntraits	Traits INDEX to be analysed. Index according to "ordered_alldata". Trait Columns index must start at 7. Example: ntraits=c(7:9), ntraits=7
nper	Number of permutations.Example: nper=1000
threshold	Threshold to be set by the hypergeometric test. threshold=0.05
seed	Set number for random sampling
saveto	Save permutation results to "workspace" OR "directory"
envir	R environment to save the Permutations to when saveto is set to "workspace"

**Value**

Returns "Permus\_genesetsname" variables or files (permutation datasets).

**See Also**

[genes\\_permutation](#)

**Examples**

```
data(demo, SNPsAnnotation)
all_data <- read_pvals(data_name=demo, snps_ann=SNPsAnnotation)
genome_results <- genome_order(all_data=all_data)
# Results from genome_order
ordered_alldata <- genome_results$ordered_alldata
gs_locs <- genome_results$gs_locs

# Create new environment to save the permutations to:
gper.env <- new.env()

data(RHSA164843, RHSA446343, RHSA8876384, RHSA8964572,
RHSA109582, RHSA1474244, envir=gper.env)

paths_res <- read2_paths(ordered_alldata=ordered_alldata, gs_locs=gs_locs,
sets_from="workspace", sets_prefix="RHSA", level="snp", envir=gper.env)
pers_ids <- paths_res$per_ors
pathways <- paths_res$pathways

# SNP permutations
snps_permutation(ordered_alldata=ordered_alldata, pers_ids=pers_ids,
ntraits=c(7,9), nper=10, saveto="workspace", threshold=0.05,
gs_locs=gs_locs, envir = gper.env)

# Get results
results <- get_results(res_pattern="Permus", level="snp",
from="workspace", threshold=0.05, envir = gper.env)
```

# Index

- \* **annotation**
    - read2\_paths, 12
  - \* **datasets**
    - demo, 4
    - RHSAXXXXX, 16
    - SNPsAnnotation, 16
  - \* **gene-level**
    - genes\_permutation, 5
  - \* **gene-set**
    - read2\_paths, 12
  - \* **genome\_order**
    - genome\_order, 7
  - \* **gwas**
    - read\_pvals, 14
  - \* **package**
    - genomicper-package, 2
  - \* **pathways**
    - read2\_paths, 12
  - \* **permutations**
    - genes\_permutation, 5
    - snps\_permutation, 17
  - \* **pvalues**
    - read\_pvals, 14
  - \* **results**
    - get\_results, 8
    - plot\_results, 11
  - \* **snp-level**
    - snps\_permutation, 17
- read2\_paths, 3, 8, 12
- read\_pvals, 3, 14
- RHSA109582 (RHSAXXXXX), 16
- RHSA1474244 (RHSAXXXXX), 16
- RHSA164843 (RHSAXXXXX), 16
- RHSA446343 (RHSAXXXXX), 16
- RHSA8876384 (RHSAXXXXX), 16
- RHSA8964572 (RHSAXXXXX), 16
- RHSAXXXXX, 16
- snps\_permutation, 3, 6, 14, 17
- SNPsAnnotation, 16
- demo, 4
- genes\_permutation, 3, 5, 14, 18
- genome\_order, 3, 7, 14, 15
- genomicper (genomicper-package), 2
- genomicper-package, 2
- get\_results, 3, 8, 11
- hyprbg, 10
- plot\_results, 3, 11