Package 'CollapsABEL'

December 11, 2016

Type Package

Title Generalized CDH (GCDH) Analysis

Version 0.10.11

Date 2016-12-11

Author Kaiyin Zhong, Fan Liu

Maintainer Kaiyin Zhong <kindlychung@gmail.com>

Depends R (>= 3.1.0), rJava (>= 0.9-6)

Imports R.utils, RSQLite, biganalytics, bigmemory, collUtils, dplyr, ggplot2, methods, stringr, stats, haplo.stats

Description Implements a generalized version of the CDH test (<DOI:10.1371/journal.pone.0028145> and <DOI:10.1186/s12859-016-1006-9>) for detecting compound heterozygosity on a genome-wide level, due to usage of generalized linear models it allows flexible analysis of binary and continuous traits with covariates.

License GPL-3

URL https://bitbucket.org/kindlychung/collapsabel2/overview

BugReports https://bitbucket.org/kindlychung/collapsabel2/issues

Suggests testthat

SystemRequirements PLINK2, Java (>= 8.0), mysql

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-11 20:35:07

R topics documented:

alphaNumeric *Alpha-numeric characters*

Description

Alpha-numeric characters

Usage

```
alphaNumeric
```

Format

An object of class `character` of length 62.

asBigMatrix *Coerce an R vector/matrix/data.frame into a big.matrix*

Description

This is a patched version of as.big.matrix from the bigmemory package. The patch allows you to omit colnames/rownames even when they exist in the R object.

Usage

```
asBigMatrix(x, type = NULL, separated = FALSE, backingfile = NULL,
  backingpath = NULL, descriptorfile = NULL, binarydescriptor = FALSE,
  shared = TRUE, dimnames = FALSE)

## S4 method for signature 'matrix,ANY,ANY,ANY,ANY,ANY,ANY,ANY,logical'
asBigMatrix(x,
  type = NULL, separated = FALSE, backingfile = NULL,
  backingpath = NULL, descriptorfile = NULL, binarydescriptor = FALSE,
  shared = TRUE, dimnames = FALSE)

## S4 method for signature 'data.frame,ANY,ANY,ANY,ANY,ANY,ANY,ANY,logical'
asBigMatrix(x,
  type = NULL, separated = FALSE, backingfile = NULL,
  backingpath = NULL, descriptorfile = NULL, binarydescriptor = FALSE,
  shared = TRUE, dimnames = FALSE)

## S4 method for signature 'vector,ANY,ANY,ANY,ANY,ANY,ANY,ANY,logical'
asBigMatrix(x,
  type = NULL, separated = FALSE, backingfile = NULL,
  backingpath = NULL, descriptorfile = NULL, binarydescriptor = FALSE,
  shared = TRUE, dimnames = FALSE)
```

Arguments

x	vector, matrix, or data.frmae
type	See bigmemory::as.big.matrix
separated	See bigmemory::as.big.matrix
backingfile	See bigmemory::as.big.matrix
backingpath	See bigmemory::as.big.matrix
descriptorfile	See bigmemory::as.big.matrix
binarydescriptor	See bigmemory::as.big.matrix
shared	See bigmemory::as.big.matrix
dimnames	logical. FALSE by default

Value

big.matrix object

Author(s)

Kaiyin Zhong, Fan Liu

---

assocFilter *Filter a PlGwasC object by the results of a* plink --assoc *run*

Description

This is meant for reduction in computational burden. The plink --assoc does not accept covariates makes some assumptions accordingly, and thus runs faster than --linear and --logistic. SNPs that does not produce a p-value more significant than a user-set threshold will be filtered out. A new PLINK file is made and a corresponding new PlGwasC object is returned.

Usage

```
assocFilter(pl_gwas, plink_out_stem = NULL, p_threshold = 0.1,
  db_setup = FALSE, force = TRUE)
```

Arguments

pl_gwas	PlGwasC object
plink_out_stem	character. Output plink file stem (without .bed extension). The default is to add a "_filtered_RANDOM_ID" suffix to the original.
p_threshold	numeric. P-value threshold.
db_setup	logical. Whether to setup the PlGwasC object.
force	logical. Overwrite existing PLINK files.

Value

a new PlGwasC object.

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
rbed_info = rbedInfo(bedstem = "mmp13", db_setup = FALSE)
pl_gwas = plGwas(rbed_info,
pheno = "mmp13.phe",
pheno_name = "Page",
gwas_tag = "mmp13_page_sex_age")
runGwas(pl_gwas)
x = readGwasOut(pl_gwas, c("SNP", "P"), rmGwasOut = FALSE)
pl_gwas1 = assocFilter(pl_gwas, p_threshold = 0.001)
runGwas(pl_gwas1)
x1 = readGwasOut(pl_gwas1, c("SNP", "P"), rmGwasOut = FALSE)
y = dplyr::inner_join(x, x1, by = "SNP")
all(y$P.x == y$P.y)
all(y$P.y < 0.001)

## End(Not run)
```

---

baseName *Basename of a FilePath object*

Description

Basename of a FilePath object

Usage

```
baseName(fp)

## S4 method for signature 'FilePath'
baseName(fp)
```

Arguments

fp	character, file paths.

Value

character vector of basenames

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
fp = filePath(R.home())
baseName(fp)

## End(Not run)
```

---

bedcollr *Shift bed files*

Description

This is a wrapper around the `bedcoll` commandline tool.

Usage

```
bedcollr(bfile = NULL, nshift_min = 1, nshift_max = NULL)
```

Arguments

`bfile`	bed filename, without the `.bed` extension.
`nshift_min`	Minimal shift number
`nshift_max`	Maximal shift number

---

bedSizeCorrect *Check whether bed file is of correct size*

Description

It is correct if its real size is the equal to its theoretical size.

Usage

```
bedSizeCorrect(rbed_info)
```

Arguments

`rbed_info`	RbedInfoC object

Value

logical.

Author(s)

Kaiyin Zhong, Fan Liu

`bimCorrectTypes` *Correct types of bim data.frame*

Description

CHR, BP and GDIST columns should be integers.

Usage

```
bimCorrectTypes(bim_dat)
```

Arguments

`bim_dat`	data.frame read from a .bim file

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

`bin2DescFilename` *Convert a .bin filename to a .desc filename*

Description

Convert a .bin filename to a .desc filename

Usage

```
bin2DescFilename(bin_file)
```

Arguments

`bin_file`	character. .bin filename

Value

character

Author(s)

Kaiyin Zhong, Fan Liu

`binPhe` *Check whether phenotype of a GWAS is binary*

Description

Check whether phenotype of a GWAS is binary

Usage

```
binPhe(pl_gwas, na_value = c(-9, 0))
```

Arguments

`pl_gwas`	PlGwasC object.
`na_value`	A vector of codes that represent missing values.

Value

logical

Author(s)

Kaiyin Zhong, Fan Liu

`bmAddCol` *Add column(s) to an existing big.matrix*

Description

This function provides an effecient way to append columns to a big.matrix (without copying columns that are already on disk).

Usage

```
bmAddCol(bin_file, dat)
```

Arguments

bin_file	character. Path to .bin file for file-backed big.matrix
dat	vector, matrix or data.frame. Coercion rules are the same as in big.matrix

Value

updated description object.

Author(s)

Kaiyin Zhong, Fan Liu

---

bmAttachBin *Attach a big.matrix by its bin filename*

Description

Attach a big.matrix by its bin filename

Usage

```
bmAttachBin(bin_file)
```

Arguments

bin_file	character. big.matrix bin filename

Author(s)

Kaiyin Zhong

---

bmConvertFun *Conversion function to use when appending values to a big.matrix*

Description

Conversion function to use when appending values to a big.matrix

Usage

```
bmConvertFun(desc)
```

Arguments

desc	description object

Value

conversion function.

Author(s)

Kaiyin Zhong, Fan Liu

`bmFilename` *Generate a big.matrix filename (.bin or .desc)*

Description

Generate a big.matrix filename (.bin or .desc)

Usage

```
bmFilename(mat_name, type)
```

Arguments

`mat_name`	character. Stem of filename.
`type`	character. Either "bin" or "desc"

Value

character. big.matrix filename

Author(s)

Kaiyin Zhong, Fan Liu

`bmFilepath` *Get the big.matrix file path according to GCDH task tag*

Description

Get the big.matrix file path according to GCDH task tag

Usage

```
bmFilepath(tag, mat_name, type)
```

Arguments

tag	character. GCDH task tag.
mat_name	character. nmiss, beta, stat, p, etc.
type	character. Either "bin" or "desc"

Author(s)

Kaiyin Zhong, Fan Liu

---

bytesSnp *Get number of bytes used by each SNP.*

Description

Get number of bytes used by each SNP.

Usage

```
bytesSnp(pl_info)
```

Arguments

pl_info	PlInfoC object

---

changeByMap *Transform a vector by a mapping*

Description

The mapping is represented by a data.frame: 1st column is the domain, 2st column is the range.

Usage

```
changeByMap(old_vector, mapping_dat, reverse = FALSE)
```

Arguments

old_vector	vector of any type.
mapping_dat	data.frame, first column must be the same type as the old_vector
reverse	logical. Reverse domain and range if set to TRUE

Value

The new vector (mapped from the old one).

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
names_dat = data.frame(c("a", "b", "c"), c("d", "e", "f"), stringsAsFactors=FALSE)
changeByMap(c("a", "a", "b"), names_dat) == c("d", "d", "e")
x = changeByMap(c(NA, "a", "b"), names_dat)
is.na(x[1])

## End(Not run)
```

---

charify *Convert certain columns of a data.frame to character type*

Description

Convert certain columns of a data.frame to character type

Usage

```
charify(dat, cols)
```

Arguments

dat	data.frame
cols	character. Names of columns to be converted.

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
x = data.frame(x = 1:3, y= 2:4)
all(colClasses(x) == c("integer", "integer"))
x = charify(x, "x")
all(colClasses(x) == c("character", "integer"))

## End(Not run)
```

checkFileExist *Stop when any file does not exist*

Description

Stop when any file does not exist

Usage

```
checkFileExist(files)
```

Arguments

files	character vector. File paths you want to check.

Author(s)

Kaiyin Zhong, Fan Liu

chExt *Change extension names*

Description

Change extension names

Usage

```
chExt(filename, ext_name)
```

Arguments

filename	character. File path
ext_name	character. New extension name

Author(s)

Kaiyin Zhong

cmh *Contrast Manhattan plot the simple way*

Description

Contrast Manhattan plot the simple way

Usage

```
cmh(gcdh_report, outfile = NULL)
```

Arguments

gcdh_report	data.frame, from a GCDH analysis
outfile	output image filepath. Any type (.png, .pdf, etc) supported by ggplot2::ggsave. Default to NULL. When it's not NULL, this function will try to save the plot to the specified path.

Value

A ggplot object

Author(s)

kaiyin

colClasses *Get classes of columns of a data.frame*

Description

Get classes of columns of a data.frame

Usage

```
colClasses(dat)
```

Arguments

dat	data.frame

Value

character. Classes of dat

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
dat = data.frame(x = 15L, y = 3.14, z = "abc",
  u = TRUE, stringsAsFactors = FALSE)
all(colClasses(dat) ==
c("integer", "numeric",
"character", "logical"))

## End(Not run)
```

---

colCors *Correlation coefficient of column-pairs of two data frames*

Description

Correlation coefficient of column-pairs of two data frames

Usage

```
colCors(dat1, dat2)
```

Arguments

dat1	first data.frame
dat2	second data.frame

Value

A vector of correlation coefficients.

Author(s)

Kaiyin Zhong

CollapsABEL *CollapsABEL: an R library for detecting compound heterozygote alleles in genome-wide association or sequencing studies*

Description

Compound Heterozygosity (CH) in classical genetics is the presen-ce of two different recessive mutations at a particular gene locus, one on each chromosome . The presence of CH has been found for nearly all autosomal recessive disorders as well as other phenotypes such as red hair color. A relaxed form of CH, i.e., in which the genetic variants are not necessarily coding, rare, and deleterious, is likely involved in a wide range of human polygenic traits and referred to as generalized CH (GCH). Howev-er, individually analyzing a large number of DNA sequence vari-ants, as being the routine in genome-wide association studies (GWAS), has limited power to detect genetic associations caused by GCH, which may be partially responsible for the currently still "missing heritability". Existing tools specifically designed for detecting GCH alleles are scarce, in particular for the analysis of densely imputed Single Nucleotide Polymorphism (SNP) array data or whole genome se-quencing data. Previously, we developed a collapsed double heter-ozygosity (CDH) test for detecting the association between CH genotypes and binary traits by applying a chi-squared statistic to pseudo-genotypes collapsed from a pair of SNPs, which was implemented as a function in the GenABEL R package . Here, we implement a generalized CDH (GCDH) method to overcome previous limitations and allow (1) fast analysis of densely imputed SNP data or whole genome se-quencing data; (2) flexible analysis of binary and quantitative traits with covariates; (3) empirical power estimation and type-I error control; and (4) easy interface with graphical utilities

Arguments

phe_file	character. Phenotype file.

Value

FALSE when the file is invalid, or a data.frame when it is.

Author(s)

Kaiyin Zhong, Fan Liu

collapse *Collpase genotypes*

Description

Collpase genotypes

Usage

```
collapse(g1, g2, collapse_matrix = NULL)
```

Arguments

g1	numeric, genotype vector 1.
g2	numeric, genotype vector 2.
collapse_matrix	matrix of integers range from 0 to 3.

Value

numeric, collapsed genotype of g1 and g2.

Author(s)

Kaiyin Zhong

---

collapseMat *Collapse two genotype matrices, column by column*

Description

Each column is assummed to be the genotype for a SNP. The two genotype matrices should have the same size.

Usage

```
collapseMat(m1, m2, collapse_matrix = matrix(c(0L, 0L, 0L, 0L, 0L, 1L, 1L, 1L,
  0L, 1L, 0L, 3L, 0L, 1L, 3L, 3L), 4, 4))
```

Arguments

m1	first genotype matrix
m2	second genotype matrix
collapse_matrix	collapsed genotype matrix

Value

collapsed genotyp matrix

Author(s)

kaiyin

collClear *Clear up CollapsABEL workspace*

Description

The workspace folder is defined in `collenv$.collapsabel_dir`.

Usage

```
collClear()
```

Author(s)

Kaiyin Zhong, Fan Liu

collenv *An environment for storing CollapsABEL package local variables*

Description

.collapsabel_dir CollapsABEL home directory

Usage

```
collenv
```

Format

An object of class `environment` of length 12.

Details

.collapsabel_gwas CollapsABEL gwas directory

.collapsabel_gcdh CollapsABEL GCDH analysis directory

.assoc_header Plink .assoc file headers

.qassoc_header Plink .qassoc file headers

.logistic_header Plink .assoc.logistic file headers

.logistic_header_default Columns from plink .assoc.logistic file headers that are used by default

.linear_header Plink .assoc.linear file headers

.linear_header_default Columns from plink .assoc.linear file headers that are used by default

.plink_out_ext Plink output extensions

.plink_stdout Plink stdout

.plink_stderr Plink stderr

connectSnpPair *Annotate a pair of SNPs in the contrast Manhattan plot*

Description

Annotate a pair of SNPs in the contrast Manhattan plot

Usage

```
connectSnpPair(cplot, snp1, snp2, linetype = "dotted", hjust = 0,
  text_size = 3)
```

Arguments

cplot	ggplot object. The contrast Manhattan plot to be annotated.
snp1	character. First SNP.
snp2	character. Second SNP.
linetype	See ggplot2::geom_segment. Default to "dotted".
hjust	See ggplot2::annotate. Default to 0.
text_size	See ggplot2::annotate. Default to 3.

Value

ggplot object.

Author(s)

Kaiyin Zhong

contrastData *Prepare data for* contrastPlot

Description

Prepare data for contrastPlot

Usage

```
contrastData(chr, bp, p, gcdh_p, snp)
```

Arguments

chr	integer. Chromosome vector.
bp	integer. Position vector.
p	numeric. P-value vector.
gcdh_p	numeric. GCDH p-value vector.
snp	character. SNP name vector.

Author(s)

Kaiyin Zhong

---

contrastPlot *Produce contrast Manhattan plot*

Description

Overlays p-values from single-SNP method and GCDH.

Usage

```
contrastPlot(chr, bp, p, gcdh_p, snp, ...)
```

Arguments

chr	integer. Chromosome vector.
bp	integer. Position vector.
p	numeric. P-value vector.
gcdh_p	numeric. GCDH p-value vector.
snp	character. SNP name vector.
...	passed to `manhattanPlot`

Value

ggplot object.

Author(s)

Kaiyin Zhong

correctDesc *Correct description of big.matrix*

Description

Correct description of big.matrix

Usage

```
correctDesc(desc_file)
```

Arguments

desc_file	character. Path to description file

Value

list. Corrected description object.

Author(s)

Kaiyin Zhong, Fan Liu

correctTypes_methods *Convert columns of a data frame to certain types*

Description

Convert columns of a data frame to certain types

Usage

```
correctTypes(dat, col_names = NULL, types)
```

Arguments

dat	data.frame The data frame whose types you want to change.
col_names	character. Names of columns, the types of which you want to change.
types	character. Names of new types. Should be the same length as col_names

Value

data.frame. With specified classes.

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
dat = randNormDat(3, 3)
dat[, 2] = as.character(dat$V2)
dat1 = correctTypes(dat, types = rep("numeric", 3))
all(colClasses(dat1) == rep("numeric", 3))
dat2 = correctTypes(dat, 2, "numeric")
all(colClasses(dat2) == rep("numeric", 3))

## End(Not run)
```

`covarNames` *Get covariate names of a GWAS*

Description

Get covariate names of a GWAS

Usage

```
covarNames(pl_gwas)
```

Arguments

`pl_gwas`	PlGwasC object.

Value

character. Vector of covariate names.

Author(s)

Kaiyin Zhong, Fan Liu

cytoband *Find cytoband at a given position*

Description

Find cytoband at a given position

Usage

```
cytoband(chr, pos, ref = "hg19")
```

Arguments

chr	integer or character. Chromosome number. If it's an integer it should be in range [1, 22]. If it's a string it's should be in the format as "chr1, chr2, ..., chr22, chrX, chrY"
pos	integer. Position on chromosome.
ref	character. Reference data. Should be either "hg18" or "hg19"

Value

Vector of cytobands.

Author(s)

kaiyin

datToVec *Extract one row or column of a data frame as a vector*

Description

Extract one row or column of a data frame as a vector

Usage

```
datToVec(dat, i, row = TRUE)
```

Arguments

dat	data.frame
i	row or column number
row	Logical. If TRUE, then i is the row number, otherwise i is the column number

Value

A vector.

Author(s)

kaiyin

`desc2BinFilename` *Convert a .desc filename to a .bin filename*

Description

Convert a .desc filename to a .bin filename

Usage

```
desc2BinFilename(desc_file)
```

Arguments

`desc_file`	character. .desc filename

Value

character

Author(s)

Kaiyin Zhong, Fan Liu

`dir.create2` *Create directory if it does not already exist*

Description

Create directory if it does not already exist

Usage

```
dir.create2(dir)
```

Arguments

`dir`	character. Path of directory to be created.

Author(s)

Kaiyin Zhong, Fan Liu

dirName *Directory name of a file path*

Description

Directory name of a file path

Usage

```
dirName(fp)

## S4 method for signature 'FilePath'
dirName(fp)
```

Arguments

fp	FilePath object

Value

character vector of directories

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
fp = filePath(R.home())
dirName(fp)

## End(Not run)
```

eprint *Print quoted expression then its value*

Description

Print quoted expression then its value

Usage

```
eprint(expr)
```

Arguments

expr	expression to be evaluated.

---

evalFile *Eval R expressions from a file.*

Description

Eval R expressions from a file.

Usage

```
evalFile(filename)
```

Arguments

filename	character

Author(s)

Kaiyin Zhong, Fan Liu

---

famCorrectTypes *Correct types of fam data.frame*

Description

SEX and PHE columns should be integers.

Usage

```
famCorrectTypes(fam_dat)
```

Arguments

fam_dat	data.frame read from a .fam file

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

fidIid *FID and IID columns from fam file*

Description

FID and IID columns from fam file

Usage

```
fidIid(pl_info)
```

Arguments

pl_info	PlInfoC object

Value

data.frame of two columns "FID" and "IID"

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
pl_info = plInfo(bedstem = "mmp13", db_setup = TRUE)
fidiid = fidIid(pl_info)
fam = readFam("mmp13.fam", c("FID", "IID"))
all(fam == fidiid)

## End(Not run)
```

file.create2 *Create file if it does not already exist*

Description

Create file if it does not already exist

Usage

```
file.create2(filename)
```

Arguments

filename	character. Path of file to be created.

Author(s)

Kaiyin Zhong, Fan Liu

---

filePath *Constructor for FilePath class*

Description

Constructor for FilePath class

Usage

```
filePath(s)
```

Arguments

s	character, path to file or dir

Value

FilePath object

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
fp = filePath(R.home())
dirName(fp) == dirname(fp@path)
baseName(fp) == basename(fp@path)

## End(Not run)
```

FilePath-class *An S4 class to represent a file path*

Description

This class comes with a validation function, making sure that the file exists.

Slots

`path` character, file or dir path

Author(s)

Kaiyin Zhong, Fan Liu

fileSize *Get file size*

Description

Get file size

Usage

```
fileSize(filename)
```

Arguments

`filename`	character. Path to file.

Value

integer. Size of file.

Author(s)

Kaiyin Zhong, Fan Liu

gcdhBmCreate *Create a big.matrix under specified GCDH tag*

Description

Create a big.matrix under specified GCDH tag

Usage

```
gcdhBmCreate(tag, bm_name, nrow, ncol = 1)
```

Arguments

tag	character. GCDH tag.
bm_name	character. Name of the big.matrix to be created.
nrow	integer. Number of rows of the big.matrix
ncol	integer. Number of columns of the big.matrix. Default to 1.

Author(s)

Kaiyin Zhong, Fan Liu

gcdhDir *Create GCDH task directories by tag*

Description

The task folder is a subfolder of the value of `collenv$.collapsabel_gcdh`. It will be created if it does not yet exist.

Usage

```
gcdhDir(gcdh_tag)
```

Arguments

gcdh_tag	character. Tag for GCDH task.

Value

character. Directory of the task.

Author(s)

Kaiyin Zhong, Fan Liu

gcdhPower *GCDH power analysis*

Description

This function makes use of runTypeI. Random phenotypes are used to survey p-values under the null hypothesis (SNPs are not associated phenotype), and genome-wide significance thresholds for single-SNP approach and GCDH are calculated by a user given alpha-level. A custom phe_fun is supplied for simulating a phenotype associated with a certain pair of SNPs. Total number of such simulations is set by the n_simu parameter. In each simulation 4 p-values are generated:

Usage

```
gcdhPower(rbed_info, n_shift, n_simu, maf_min, maf_max, r_limit, beta,
  collapse_matrix = NULL, dist_threshold = 5e+05, alpha_level = 0.05)
```

Arguments

rbed_info	RbedInfoC object
n_shift	integer. n_shift for each GCDH run.
n_simu	integer. Number of simulations to run.
maf_min	numeric. Lower limit of MAF interval.
maf_max	numeric. Upper limit of MAF interval.
r_limit	numeric. Upper limit of correlation coefficient between the two causal SNPs.
beta	numeric. Effect size of the simulated phenotype.
collapse_matrix	See runGcdh.
dist_threshold	See runGcdh.
alpha_level	numeric. Control type-I error rate at this level.

Details

P_single: p-values from single-SNP approach.

P_GCDH: p-values from GCDH.

P_(single,no causal): p-values from single-SNP approach when causal SNPs are untyped.

P_(GCDH,no causal): p-values from GCDH when causal SNPs are untyped.

When all simulations are finished, 4 vectors of p-values are obtained: P_single_vec, P_GCDH_vec, P_(single,no causal)_vec, P_(GCDH,no causal)_vec. The power for each of the category (single-SNP, single-SNP without causal genotypes, GCDH, GCDH without causal genotypes) are proportions of these vectors that are more significant than the genome-wide significance thresholds we have obtained.

Author(s)

Kaiyin Zhong

gcdhRegion *Run GCDH over a region*

Description

A region around some SNP is extracted and GCDH analysis is conducted over that region.

Usage

```
gcdhRegion(pl_gwas, n_shift = NULL, snp, window = 500, out = NULL,
  gwas_col_select = collenv$.linear_header_default, collapse_matrix = NULL,
  rm_shifted_files = TRUE, dist_threshold = 5e+05)
```

Arguments

pl_gwas	PlGwasC object
n_shift	integer. Maximum shift number.
snp	character. SNP name
window	numeric. All variants with physical position no more than half the specified kb distance (decimal permitted) from the named variant are loaded.
out	character. Path to the regional bed file (without .bed extension).
gwas_col_select	character. See runGcdh
collapse_matrix	See runGcdh
rm_shifted_files	See runGcdh
dist_threshold	See runGcdh

Value

See runGcdh

Author(s)

Kaiyin Zhong, Fan Liu

gcdhReport *Generate a report from a GCDH run*

Description

For each p-value from a GCDH run, search for indices of the corresponding SNP pair. Combine statistics from single-SNP approach with GCDH statistics.

Usage

```
gcdhReport(run_res)
```

Arguments

run_res	Result from runGcdh

Value

path to SQLite database

Author(s)

Kaiyin Zhong, Fan Liu

---

getHaplo *Infer haplotypes for a pair of SNPs*

Description

Infer haplotypes for a pair of SNPs

Usage

```
getHaplo(geno, format_idx = NULL)
```

Arguments

geno	Genotype data frame. Must have 4 columns, the first two being "FID" and "IID", the last two being the genotypes.
format_idx	Column indices used for formatting haplotype string.

Value

A data frame of haplotypes

Author(s)

kaiyin

`getHaplos` *Inferring haplotypes from two genotype data frames, and join with phenotypes*

Description

Inferring haplotypes from two genotype data frames, and join with phenotypes

Usage

```
getHaplos(g1, g2, phe, pool = NULL)
```

Arguments

`g1`	First genotype data frame
`g2`	Second genotype data frame, must be of the same dimension as the first. The first two column must be FID and IID.
`phe`	Phenotype data frame, the first two columns must be FID and IID
`pool`	A genotype data frame, assumed to be different from g1 and g2, used for pooling.

Value

A data frame containing phenotype and haplotype for each individual.

Author(s)

kaiyin

`getOrElse-operator` *Default value for expression.*

Description

When an expression evals to NULL, take the default value instead. Copied from dplyr source.

Usage

```
x %||% y
```

Arguments

`x`	expression to be evaled.
`y`	default value.

Author(s)

Hadley Wickham

`getQuery` *Get query results from a SQLite database*

Description

Get query results from a SQLite database

Usage

```
getQuery(db_name, query_string)
```

Arguments

`db_name`	character. Path to database.
`query_string`	character. Query string.

Author(s)

Kaiyin Zhong, Fan Liu

`getr2` *Estimate percentage of variation explained*

Description

Estimate percentage of variation explained

Usage

```
getr2(df, yn)
```

Arguments

`df`	Dataframe
`yn`	Name of the independent variable, must be one of the columns of `df`

Author(s)

Fan Liu

glm2 *GLM with arbitrary column names*

Description

Substitute column names that are unsuitable for formulas and substitute back when returning results.

Usage

```
glm2(dat, y, xs, ...)
```

Arguments

dat	data.frame. Souce data to build GLM upon.
y	character. Column name of dependent variable.
xs	character. Column names of independent variable.
...	passed to glm.

Value

data.frame of coefficients.

Author(s)

Kaiyin Zhong, Fan Liu

glmIter *Perform glm iteratively through a number of independent variables with fixed dependent variables and covariates.*

Description

Perform glm iteratively through a number of independent variables with fixed dependent variables and covariates.

Usage

```
glmIter(dat, y, xs = NULL, covars = character(), ...)
```

Arguments

dat	data.frame
y	character. Name of dependent variable columns.
xs	character. Names of independent variable columns.
covars	character. Names of covriate columns.
...	passed to glm.

Value

matrix of coefficients

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasDat *Read genotype and phenotype data into R*

Description

Read genotype and phenotype data into R

Usage

```
gwasDat(pl_gwas, snp_vec)
```

Arguments

pl_gwas	PlGwasC object.
snp_vec	numeric or character. Vector of SNPs.

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasDir *GWAS results directory of a certain GWAS scan*

Description

GWAS results directory of a certain GWAS scan

Usage

```
gwasDir(pl_gwas)
```

Arguments

pl_gwas	PlGwasC object

Value

character.

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasLog *Plink log fie*

Description

Redirect stdout to this file when plink is running.

Usage

```
gwasLog(pl_gwas)
```

Arguments

pl_gwas	PlGwasC object.

Value

character. Path to log file.

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasOut *GWAS output file name*

Description

GWAS output file name

Usage

```
gwasOut(pl_gwas)
```

Arguments

pl_gwas	PlGwasC object.

Value

character

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasOutStem *Plink output filename*

Description

To be passed as the `--out` option to plink.

Usage

```
gwasOutStem(pl_gwas)
```

Arguments

`pl_gwas`	PlGwasC object.

Value

character. Plink output filename, without extension

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasR *Invoke a GWAS in R*

Description

Invoke a GWAS in R

Usage

```
gwasR(pl_gwas, snp_vec)
```

Arguments

`pl_gwas`	PlGwasC object.
`snp_vec`	numeric or character. Vector of SNPs.

Value

matrix. Coefficient matrix. One row for each SNP.

Author(s)

Kaiyin Zhong, Fan Liu

---

gwasRDS *Get RDS file path of a PlGwasC object*

Description

Get RDS file path of a PlGwasC object

Usage

```
gwasRDS(pl_gwas)
```

Arguments

pl_gwas	PlGwasC object.

Value

character. path of a PlGwasC object

Author(s)

Kaiyin Zhong, Fan Liu

---

head2 *Head and tail in two dimensions*

Description

Restrict not only the number of rows, but also the number of columns.

Usage

```
head2(x, m = 6, n = NULL)

tail2(x, m = 6, n = NULL)
```

Arguments

x	data.frame or matrix
m	integer. Number of rows to keep.
n	integer. Number of columns to keep.

Author(s)

kaiyin

---

headPhe *Read first n lines of a phenotype file*

Description

Read first n lines of a phenotype file

Usage

```
headPhe(pl_gwas, nrows = 5L)
```

Arguments

pl_gwas	PlGwasC object
nrows	number of lines to read

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

isBinary *Check whether a trait is binary*

Description

Check whether a trait is binary

Usage

```
isBinary(v, na_value = NULL)
```

Arguments

v	numeric vector.
na_value	a vector of numeric values which should be seen as NA.

Value

logical

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
!isBinary(c(1, 1.1, 1, 1.1, NA))
isBinary(c(1, 2, 1, 2, NA))
!isBinary(c(-9, 2.3, 4.1, -9, -9), -9)
isBinary(c(-9, 2, 4, -9, -9), -9)
isBinary(c(1, 2, 2, 1, -9, -9.9), c(-9, -9.9))

## End(Not run)
```

isS4Class *Check whether an S4 object is of a certain class*

Description

Check whether an S4 object is of a certain class

Usage

```
isS4Class(obj, c)
```

Arguments

obj	S4 object
c	Class name

Value

logical

Author(s)

Kaiyin Zhong, Fan Liu

---

isSetup *Check if a directory containing .bed .fam and .bim files is properly setup*

Description

Check if a directory containing .bed .fam and .bim files is properly setup

Usage

```
isSetup(pl_info)
```

Arguments

pl_info	PlInfoC object

Value

TRUE or FALSE

Author(s)

Kaiyin Zhong, Fan Liu

isSetupRbed *Check if an RbedInfoC object is properly set up*

Description

Check if an RbedInfoC object is properly set up

Usage

```
isSetupRbed(rbed_info)
```

Arguments

rbed_info	RbedInfoC object

Value

logical.

Author(s)

Kaiyin Zhong, Fan Liu

isSQLite3 *Check whether a file is a SQLite3 database.*

Description

Check whether a file is a SQLite3 database.

Usage

```
isSQLite3(filename)
```

Arguments

filename	character. Path to file to be checked.

Author(s)

Kaiyin Zhong, Fan Liu

`lagDistance` *Distance with lag*

Description

Calculate the distance between each element in a numeric vector and the element that is `lag` positions after it. For the last `lag` elemnts, this distance does not exist, so NA is used as a placeholder. The returned vector is of the same length as the input vector.

Usage

```
lagDistance(vec, lag = 1, reverse = FALSE)
```

Arguments

`vec`	numeric.
`lag`	integer.
`reverse`	logical. Default to FALSE, i.e. calculate `vec[i+lag] - vec[i]`. When set to TRUE, calculate `vec[i] - vec[i+lag]`

Value

numeric.

Author(s)

Kaiyin Zhong, Fan Liu

`lenCheck` *Check each element of a list has expected length Give a* `list(a, b, ...)` *and* `vector(l1, l2, ...)`*, check that length of a is equal to l1, length of b is equal to l2, etc.*

Description

Check each element of a list has expected length

Give a `list(a, b, ...)` and `vector(l1, l2, ...)`, check that length of a is equal to l1, length of b is equal to l2, etc.

Usage

```
lenCheck(ilist, ilengths)
```

Arguments

ilist	list of items you want to check.
ilengths	vector of lengths for these items.

Value

TRUE or a string

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
lenCheck(list(1, 2, 3), c(1, 1, 0))
grepl("\\nGiven: \n.*", lenCheck(list(1, 2, 3), c(1, 1, 0)))
grepl("\\nGiven: \n.*", lenCheck(list(1, c(1, 2, 3), list(4, 5, 6)), c(1, 1, 0)))
lenCheck(list(1, c(1, 2, 3), list(4, 5, 6)), c(1, 3, 3))

## End(Not run)
```

---

listEqual *Check equality of two lists*

Description

Check equality of two lists

Usage

```
listEqual(list1, list2)
```

Arguments

list1	list
list2	list

Author(s)

Kaiyin Zhong, Fan Liu

listGwasTags *List GWAS or GCDH tags*

Description

List GWAS or GCDH tags

Usage

```
listGwasTags(type = "gwas")

listTags(type = "gwas")
```

Arguments

type	character. Either "gwas" or "gcdh".

Author(s)

Kaiyin Zhong, Fan Liu

loadGwas *Load PlGwasC object by tag, from the RDS file*

Description

Load PlGwasC object by tag, from the RDS file

Usage

```
loadGwas(gwas_tag)
```

Arguments

gwas_tag	character. Tag of a GWAS run.

Value

PlGwasC object.

Author(s)

Kaiyin Zhong, Fan Liu

makePhe *Generate phenotype file from a fam file*

Description

Generate phenotype file from a fam file

Usage

```
makePhe(famfile, n_components)
```

Arguments

famfile	Character. Path of fam file.
n_components	Integer. Number of principle components to generate.

Value

Phenotype data.frame. The data frame contains the FID, IID, SEX, AFFECTEDNESS columns of the fam file, plus principle components of genetic information.

Author(s)

kaiyin

manhattanData *Prepare data for Manhattan plot.*

Description

Prepare data for Manhattan plot.

Usage

```
manhattanData(chr, bp, p, snp, color_vec = NULL, sort_chr_bp = TRUE)
```

Arguments

chr	integer. Chromosome vector.
bp	integer. Position vector.
p	numeric. P-value vector.
snp	character. SNP name vector.
color_vec	character/factor. Color vector. Doesn't have to be color names, any categorical variable will be fine.
sort_chr_bp	logical. Whether to sort the whole data frame by CHR and BP before return.

Value

A list with the following members (1) A data frame with columns including CHR, SNP, BP, P, etc. (2) Total number of SNPs. (3) A vector of unique chromosomes.

Author(s)

Kaiyin Zhong

---

manhattanPlot *Produce Manhattan plot*

Description

Produce Manhattan plot

Usage

```
manhattanPlot(mh_dat_res, hlines = NULL)
```

Arguments

mh_dat_res	list. Result from manhattanData
hlines	numeric. Horizontal lines to draw.

Value

ggplot object.

Author(s)

Kaiyin Zhong

---

nIndivApprPl *Get apparent number of individuals*

Description

Get apparent number of individuals

Usage

```
nIndivApprPl(pl_info)
```

Arguments

pl_info	PlInfoC object

nIndivPl *Get number of individuals*

Description

Get number of individuals

Usage

```
nIndivPl(pl_info)
```

Arguments

pl_info	PlInfoC object

nonExistentFiles *Non-existent files from a vector of filenames*

Description

This function receives a vector of filenames as parameter, and returns a vector of non-existent files among them.

Usage

```
nonExistentFiles(filenames)
```

Arguments

filenames	character A vector of filenames

Value

A character vector of file paths that do not exist.

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
nonExistentFiles(R.home())
nonExistentFiles(sapply(1:5, function(i) tempfile()))
nonExistentFiles(sapply(1:5, function(i) tempdir()))
nonExistentFiles(c("/tmp/f3412lds43289ajkfdlsa", R.home())) == "/tmp/f3412lds43289ajkfdlsa"

## End(Not run)
```

nSnpPl *Get number of SNPs.*

Description

Get number of SNPs.

Usage

```
nSnpPl(pl_info)
```

Arguments

pl_info	PlInfoC object

numVectorSQLRepr *String representation of a numeric vector for SQLite consumption*

Description

Transform a numeric vector (e.g. `c(1, 2)` into a string representation that can be used in a SQLite query (e.g. "(1, 2)").

Usage

```
numVectorSQLRepr(vec, print_out = FALSE)
```

Arguments

vec	numeric.
print_out	logical. Whether to print out the string representation.

Author(s)

Kaiyin Zhong

permutePhe *Permute a phenotype file*

Description

All columns except FID and IID are permuted.

Usage

```
permutePhe(phe_file, out_file, force = FALSE, valid = TRUE, ...)
```

Arguments

phe_file	character. Phenotype file.
out_file	character. Path to permuted phenotype file.
force	logical. When set to TRUE, existing file is overwritten.
valid	logical. Whether to validate the phenotype file first.
...	Passed to read.table

Author(s)

Kaiyin Zhong, Fan Liu

plGwas *Constructor for PlGwasC class*

Description

Constructor for PlGwasC class

Usage

```
plGwas(pl_gwas, pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'PlGwasC,character,character,character,character,logical,list'
plGwas(pl_gwas,
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'RbedInfoC,character,character,character,character,logical,list'
plGwas(pl_gwas,
```

```
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'RbedInfoC,character,character,character,character,missing,missing'
plGwas(pl_gwas,
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'RbedInfoC,character,character,character,character,missing,list'
plGwas(pl_gwas,
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'RbedInfoC,character,character,character,character,logical,missing'
plGwas(pl_gwas,
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)


  ## S4 method for signature
## 'RbedInfoC,character,character,missing,character,missing,missing'
plGwas(pl_gwas,
  pheno, pheno_name, covar_name, gwas_tag, assoc, opts)
```

Arguments

pl_gwas	PlGwasC or PlInfoC object
pheno	character. Phenotype file
pheno_name	character. Phenotype names.
covar_name	character. Covariate names.
gwas_tag	character. Tag for this GWAS.
assoc	logical. Whether use the "–assoc" option for PLINK.
opts	list. Options to be passed to PLINK.

Value

PlGwasC object

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
gwas_tag = "mmp13_page_sex_age"
```

```
rbed_info = rbedInfo(bedstem = "mmp13")
pl_gwas = plGwas(rbed_info,
pheno = "mmp13.phe",
pheno_name = "Page",
covar_name = "Sex,Cage",
gwas_tag = gwas_tag)
runGwas(pl_gwas)
"mmp13_page_sex_age" %in% listGwasTags() == "TRUE"
gwas_out = readGwasOut(pl_gwas, rmGwasOut = FALSE)
colClasses(gwas_out) == c("integer", "character", "integer",
    "character", "character", "integer",
    "numeric", "numeric", "numeric")

## End(Not run)
```

`PlGwasC-class` *An S4 class representing info about GWAS on plink files*

Description

An S4 class representing info about GWAS on plink files

Slots

`gwas_tag` character. Tag for this GWAS.

`opts` list. Plink options.

`plInfo` *Constructor for PlInfoC class*

Description

Populates an PlInfoC object from a given plink bed filename stem (i.e. exclude extension name)

Usage

```
plInfo(pl_info, bedstem, db_setup)

## S4 method for signature 'PlInfoC,character,logical'
plInfo(pl_info, bedstem, db_setup)

## S4 method for signature 'PlInfoC,character,missing'
plInfo(pl_info, bedstem, db_setup)

## S4 method for signature 'missing,character,logical'
```

```
plInfo(pl_info, bedstem, db_setup)

## S4 method for signature 'missing,character,missing'
plInfo(pl_info, bedstem, db_setup)
```

Arguments

pl_info	a PlInfoC object, possibly empty.
bedstem	path of bed file excluding extension name
db_setup	logical. Whether to setup SQLite database for .bim, .fam and .frq files.

Value

a PlInfoC object

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
pl_info = plInfo(.PlInfoC(), "mmp13", db_setup = TRUE)
isSetup(pl_info)
bim_ff = getQuery(sqliteFilePl(pl_info), "select * from bim")
fam_ff = getQuery(sqliteFilePl(pl_info), "select * from fam")
frq_ff = getQuery(sqliteFilePl(pl_info), "select * from frq")

## End(Not run)
```

PlInfoC-class *An S4 class representing info about plink files*

Description

Info about plink files, including the root directory, paths of plink .bed, .bim, .fam and .frq files, ff backing directories for .bim, .fam and .frq files, etc.

Slots

`main_dir` Root directory where .bed, .bim and .fam files sit.

`plink_stem` character. Path to the .bed file sans the extension name

`plink_trio` character of length 3. Paths to .bed, .bim and .fam files (in that order).

`plink_trio_base` character. Basenames of `plink_trio`.

`plink_frq` character. Path to .frq file.

plinkr *A wrapper for plink*

Description

A wrapper for plink

Usage

```
plinkr(D = NULL, K = NULL, a1_allele = NULL, a2_allele = NULL,
  adjust = NULL, all = NULL, all_pheno = NULL, allele1234 = NULL,
  alleleACGT = NULL, allele_count = NULL, allow_extra_chr = NULL,
  allow_no_sex = NULL, alt_group = NULL, alt_snp = NULL,
  annotate = NULL, annotate_snp_field = NULL, aperm = NULL,
  assoc = NULL, attrib = NULL, attrib_indiv = NULL, autosome = NULL,
  autosome_num = NULL, autosome_xy = NULL, bcf = NULL, bd = NULL,
  bed = NULL, beta = NULL, bfile = NULL, bgen = NULL,
  biallelic_only = NULL, bim = NULL, blocks = NULL,
  blocks_inform_frac = NULL, blocks_max_kb = NULL, blocks_min_maf = NULL,
  blocks_recomb_highci = NULL, blocks_strong_highci = NULL,
  blocks_strong_lowci = NULL, bmerge = NULL, border = NULL,
  bp_space = NULL, case_only = NULL, cc = NULL, cell = NULL,
  cfile = NULL, chap = NULL, check_sex = NULL, chr = NULL,
  chr_set = NULL, ci = NULL, clump = NULL, clump_allow_overlap = NULL,
  clump_annotate = NULL, clump_best = NULL, clump_field = NULL,
  clump_index_first = NULL, clump_kb = NULL, clump_p1 = NULL,
  clump_p2 = NULL, clump_r2 = NULL, clump_range = NULL,
  clump_range_border = NULL, clump_replicate = NULL,
  clump_snp_field = NULL, clump_verbose = NULL, cluster = NULL,
  cluster_missing = NULL, cm_map = NULL, cnv_blue = NULL,
  cnv_border = NULL, cnv_brown = NULL, cnv_check_no_overlap = NULL,
  cnv_count = NULL, cnv_del = NULL, cnv_disrupt = NULL,
  cnv_drop_no_segment = NULL, cnv_dup = NULL, cnv_enrichment_test = NULL,
  cnv_exclude = NULL, cnv_exclude_off_by_1 = NULL,
  cnv_freq_excldue_above = NULL, cnv_freq_excldue_below = NULL,
  cnv_freq_excldue_exact = NULL, cnv_freq_exclude_above = NULL,
  cnv_freq_exclude_below = NULL, cnv_freq_exclude_exact = NULL,
  cnv_freq_incldue_exact = NULL, cnv_freq_include_exact = NULL,
  cnv_freq_method2 = NULL, cnv_freq_overlap = NULL, cnv_green = NULL,
  cnv_indiv_perm = NULL, cnv_intersect = NULL, cnv_kb = NULL,
  cnv_list = NULL, cnv_make_map = NULL, cnv_max_kb = NULL,
  cnv_max_score = NULL, cnv_max_sites = NULL, cnv_overlap = NULL,
  cnv_red = NULL, cnv_region_overlap = NULL, cnv_report_regions = NULL,
  cnv_score = NULL, cnv_seglist = NULL, cnv_sites = NULL,
  cnv_subset = NULL, cnv_test = NULL, cnv_test_1sided = NULL,
  cnv_test_2sided = NULL, cnv_test_region = NULL, cnv_test_window = NULL,
  cnv_track = NULL, cnv_union_overlap = NULL, cnv_unique = NULL,
```

```
cnv_verbose_report_regions = NULL, cnv_write = NULL,
cnv_write_freq = NULL, complement_sets = NULL,
compound_genotypes = NULL, compress = NULL, condition = NULL,
condition_list = NULL, consensus_match = NULL, const_fid = NULL,
control = NULL, counts = NULL, covar = NULL, covar_name = NULL,
covar_number = NULL, cow = NULL, d = NULL, data = NULL,
debug = NULL, decompress = NULL, dfam = NULL, distance = NULL,
distance_exp = NULL, distance_matrix = NULL, dog = NULL,
dominant = NULL, dosage = NULL, double_id = NULL, dprime = NULL,
dummy = NULL, dummy_coding = NULL, each_versus_others = NULL,
each_vs_others = NULL, epistasis = NULL, epistasis_summary_merge = NULL,
exclude = NULL, exclude_before_extract = NULL, exclude_snp = NULL,
exclude_snps = NULL, extract = NULL, fam = NULL, family = NULL,
fast_epistasis = NULL, fid = NULL, file = NULL,
fill_missing_a2 = NULL, filter = NULL, filter_cases = NULL,
filter_controls = NULL, filter_females = NULL, filter_founders = NULL,
filter_males = NULL, filter_nonfounders = NULL, fisher = NULL,
flip = NULL, flip_scan = NULL, flip_scan_threshold = NULL,
flip_scan_verbose = NULL, flip_scan_window = NULL,
flip_scan_window_kb = NULL, flip_subset = NULL, freq = NULL,
freqx = NULL, from = NULL, from_bp = NULL, from_kb = NULL,
from_mb = NULL, frqx = NULL, fst = NULL, gap = NULL, gates = NULL,
gc = NULL, gen = NULL, gene = NULL, gene_all = NULL,
gene_list = NULL, gene_list_border = NULL, gene_report = NULL,
gene_report_empty = NULL, gene_report_snp_field = NULL,
gene_subset = NULL, genedrop = NULL, genepi = NULL, geno = NULL,
genome = NULL, genome_full = NULL, genome_lists = NULL,
genome_minimal = NULL, genotypic = NULL, gfile = NULL, gplink = NULL,
grm = NULL, grm_bin = NULL, grm_gz = NULL, group_avg = NULL,
groupdist = NULL, gxe = NULL, hap... = NULL, hap = NULL,
hap_assoc = NULL, hap_freq = NULL, hap_impute = NULL,
hap_max_phase = NULL, hap_min_phase_prob = NULL, hap_miss = NULL,
hap_phase = NULL, hap_phase_wide = NULL, hap_pp = NULL,
hap_snps = NULL, hap_tdt = NULL, hap_window = NULL,
hard_call_threshold = NULL, hardy2 = NULL, hardy = NULL, help = NULL,
het = NULL, hethom = NULL, hide_covar = NULL, homog = NULL,
homozyg = NULL, homozyg_density = NULL, homozyg_gap = NULL,
homozyg_group = NULL, homozyg_het = NULL,
homozyg_include_missing = NULL, homozyg_kb = NULL, homozyg_match = NULL,
homozyg_snp = NULL, homozyg_verbose = NULL, homozyg_window_het = NULL,
homozyg_window_kb = NULL, homozyg_window_missing = NULL,
homozyg_window_snp = NULL, homozyg_window_threshold = NULL,
horse = NULL, hwe = NULL, hwe_all = NULL, ibc = NULL, ibm = NULL,
ibs_matrix = NULL, ibs_test = NULL, id_delim = NULL, id_dict = NULL,
id_match = NULL, iid = NULL, impossible = NULL, impute_sex = NULL,
ind_major = NULL, indep = NULL, indep_pairphase = NULL,
indep_pairwise = NULL, independent_effect = NULL, indiv_sort = NULL,
inter_chr = NULL, interaction = NULL, je_cellmin = NULL, keep = NULL,
```

```
keep_allele_order = NULL, keep_autoconv = NULL,
keep_before_remove = NULL, keep_cluster_names = NULL,
keep_clusters = NULL, keep_fam = NULL, lambda = NULL, lasso = NULL,
lasso_select_covars = NULL, ld = NULL, ld_snp = NULL,
ld_snp_list = NULL, ld_snps = NULL, ld_window = NULL,
ld_window_kb = NULL, ld_window_r2 = NULL, ld_xchr = NULL,
lfile = NULL, liability = NULL, linear = NULL, list = NULL,
list_23_indels = NULL, list_all = NULL, logistic = NULL,
lookup... = NULL, lookup = NULL, lookup_gene = NULL,
lookup_list = NULL, loop_assoc = NULL, maf = NULL, maf_succ = NULL,
make_bed = NULL, make_founders = NULL, make_grm = NULL,
make_grm_bin = NULL, make_grm_gz = NULL, make_just_bim = NULL,
make_just_fam = NULL, make_perm_pheno = NULL, make_pheno = NULL,
make_rel = NULL, make_set = NULL, make_set_border = NULL,
make_set_collapse_group = NULL, make_set_complement_all = NULL,
make_set_complement_group = NULL, map = NULL, mat = NULL,
match = NULL, match_type = NULL, matrix = NULL, max = NULL,
max_maf = NULL, mc = NULL, mcc = NULL, mcovar = NULL,
mds_cluster = NULL, mds_plot = NULL, me = NULL, me_exclude_one = NULL,
memory = NULL, mendel = NULL, mendel_duos = NULL,
mendel_multigen = NULL, merge = NULL, merge_equal_pos = NULL,
merge_list = NULL, merge_mode = NULL, merge_x = NULL,
meta_analysis = NULL, meta_analysis_..._field = NULL, mfilter = NULL,
mh = NULL, mhf = NULL, min = NULL, mind = NULL,
mishap_window = NULL, missing = NULL, missing_code = NULL,
missing_genotype = NULL, missing_phenotype = NULL,
missing_var_code = NULL, mlma = NULL, mlma_loco = NULL,
mlma_no_adj_covar = NULL, model = NULL, model_dom = NULL,
model_gen = NULL, model_rec = NULL, model_trend = NULL, mouse = NULL,
mperm = NULL, mperm_save = NULL, mperm_save_all = NULL, mpheno = NULL,
must_have_sex = NULL, mwithin = NULL, neighbour = NULL, no_fid = NULL,
no_parents = NULL, no_pheno = NULL, no_sex = NULL, no_snp = NULL,
no_x_sex = NULL, nonfounders = NULL, nop = NULL, not_chr = NULL,
nudge = NULL, null_group = NULL, null_snp = NULL,
oblig_cluster = NULL, oblig_clusters = NULL, oblig_missing = NULL,
out = NULL, output_chr = NULL, output_missing_genotype = NULL,
output_missing_phenotype = NULL, oxford_pheno_name = NULL,
parallel = NULL, parameters = NULL, parentdt1 = NULL,
parentdt2 = NULL, pat = NULL, pca = NULL, pca_cluster_names = NULL,
pca_clusters = NULL, ped = NULL, pedigree = NULL, perm = NULL,
perm_batch_size = NULL, perm_count = NULL, pfilter = NULL,
pheno = NULL, pheno_merge = NULL, pheno_name = NULL, pick1 = NULL,
plist = NULL, poo = NULL, pool_size = NULL, ppc = NULL,
ppc_gap = NULL, proxy_... = NULL, proxy_assoc = NULL,
proxy_b_kb = NULL, proxy_b_maxsnp = NULL, proxy_b_r2 = NULL,
proxy_b_threshold = NULL, proxy_b_window = NULL, proxy_dosage = NULL,
proxy_drop = NULL, proxy_flanking = NULL, proxy_geno = NULL,
proxy_genotypic_concordance = NULL, proxy_glm = NULL,
```

```
proxy_impute = NULL, proxy_impute_threshold = NULL, proxy_kb = NULL,
proxy_list = NULL, proxy_maf = NULL, proxy_maxsnp = NULL,
proxy_mhf = NULL, proxy_r2 = NULL, proxy_r2_no_filter = NULL,
proxy_replace = NULL, proxy_show_proxies = NULL,
proxy_sub_maxsnp = NULL, proxy_sub_r2 = NULL, proxy_tdt = NULL,
proxy_verbose = NULL, proxy_window = NULL, prune = NULL,
q_score_file = NULL, q_score_range = NULL, qfam... = NULL,
qmatch = NULL, qq_plot = NULL, qt = NULL, qt_means = NULL,
qual_geno_... = NULL, qual_geno_max_threshold = NULL,
qual_geno_scores = NULL, qual_geno_threshold = NULL,
qual_max_threshold = NULL, qual_scores = NULL, qual_threshold = NULL,
r2 = NULL, r = NULL, range = NULL, rank = NULL, read_dists = NULL,
read_freq = NULL, read_genome = NULL, read_genome_list = NULL,
read_genome_minimal = NULL, recessive = NULL, recode12 = NULL,
recode = NULL, recodeA = NULL, recodeAD = NULL, recodeHV = NULL,
recode_allele = NULL, recode_beagle = NULL, recode_bimbam = NULL,
recode_fastphase = NULL, recode_lgen = NULL, recode_rlist = NULL,
recode_structure = NULL, recode_vcf = NULL, recode_whap = NULL,
reference = NULL, reference_allele = NULL, regress_distance = NULL,
regress_pcs = NULL, regress_rel = NULL, rel_check = NULL,
rel_cutoff = NULL, remove = NULL, remove_cluster_names = NULL,
remove_clusters = NULL, remove_fam = NULL, rerun = NULL, rice = NULL,
sample = NULL, score = NULL, score_no_mean_imputation = NULL,
script = NULL, seed = NULL, set = NULL, set_by_all = NULL,
set_collapse_all = NULL, set_hh_missing = NULL, set_max = NULL,
set_me_missing = NULL, set_missing_nonsnp_ids = NULL,
set_missing_snp_ids = NULL, set_missing_var_ids = NULL,
set_names = NULL, set_p = NULL, set_r2 = NULL, set_r2_phase = NULL,
set_table = NULL, set_test = NULL, sex = NULL, sheep = NULL,
show_tags = NULL, silent = NULL, simulate = NULL,
simulate_haps = NULL, simulate_label = NULL, simulate_missing = NULL,
simulate_n = NULL, simulate_ncases = NULL, simulate_ncontrols = NULL,
simulate_prevalence = NULL, simulate_qt = NULL, simulate_tags = NULL,
snp = NULL, snps = NULL, snps_only = NULL, specific_haplotype = NULL,
split_x = NULL, standard_beta = NULL, subset = NULL,
swap_parents = NULL, swap_sibs = NULL, swap_unrel = NULL, tab = NULL,
tag_kb = NULL, tag_mode2 = NULL, tag_r2 = NULL, tail_pheno = NULL,
tdt = NULL, test_all = NULL, test_mishap = NULL, test_missing = NULL,
test_snp = NULL, tests = NULL, tfam = NULL, tfile = NULL,
thin = NULL, thin_count = NULL, threads = NULL, to = NULL,
to_bp = NULL, to_kb = NULL, to_mb = NULL, tped = NULL,
transpose = NULL, trend = NULL, tucc = NULL, twolocus = NULL,
unbounded = NULL, unrelated_heritability = NULL, update_alleles = NULL,
update_chr = NULL, update_cm = NULL, update_ids = NULL,
update_map = NULL, update_name = NULL, update_parents = NULL,
update_sex = NULL, vcf = NULL, vcf_filter = NULL,
vcf_half_call = NULL, vcf_idspace_to = NULL, vcf_min_qual = NULL,
vegas = NULL, version = NULL, vif = NULL, whap = NULL,
```

```
window = NULL, with_freqs = NULL, with_phenotype = NULL,
with_reference = NULL, within = NULL, write_cluster = NULL,
write_covar = NULL, write_dosage = NULL, write_set = NULL,
write_set_r2 = NULL, write_snplist = NULL, xchr_model = NULL,
zero_cluster = NULL, zero_cms = NULL, one = NULL, twothreefile = NULL,
stdout = collenv$.plink_stdout, stderr = collenv$.plink_stderr,
wait = TRUE)
```

Arguments

D	Same as plink –D
K	Same as plink –K
a1_allele	Same as plink –a1-allele
a2_allele	Same as plink –a2-allele
adjust	Same as plink –adjust
all	Same as plink –all
all_pheno	Same as plink –all-pheno
allele1234	Same as plink –allele1234
alleleACGT	Same as plink –alleleACGT
allele_count	Same as plink –allele-count
allow_extra_chr	Same as plink –allow-extra-chr
allow_no_sex	Same as plink –allow-no-sex
alt_group	Same as plink –alt-group
alt_snp	Same as plink –alt-snp
annotate	Same as plink –annotate
annotate_snp_field	Same as plink –annotate-snp-field
aperm	Same as plink –aperm
assoc	Same as plink –assoc
attrib	Same as plink –attrib
attrib_indiv	Same as plink –attrib-indiv
autosome	Same as plink –autosome
autosome_num	Same as plink –autosome-num
autosome_xy	Same as plink –autosome-xy
bcf	Same as plink –bcf
bd	Same as plink –bd
bed	Same as plink –bed
beta	Same as plink –beta
bfile	Same as plink –bfile

bgen Same as plink –bgen

biallelic_only Same as plink –biallelic-only

bim Same as plink –bim

blocks Same as plink –blocks

blocks_inform_frac
Same as plink –blocks-inform-frac

blocks_max_kb Same as plink –blocks-max-kb

blocks_min_maf Same as plink –blocks-min-maf

blocks_recomb_highci
Same as plink –blocks-recomb-highci

blocks_strong_highci
Same as plink –blocks-strong-highci

blocks_strong_lowci
Same as plink –blocks-strong-lowci

bmerge Same as plink –bmerge

border Same as plink –border

bp_space Same as plink –bp-space

case_only Same as plink –case-only

cc Same as plink –cc

cell Same as plink –cell

cfile Same as plink –cfile

chap Same as plink –chap

check_sex Same as plink –check-sex

chr Same as plink –chr

chr_set Same as plink –chr-set

ci Same as plink –ci

clump Same as plink –clump

clump_allow_overlap
Same as plink –clump-allow-overlap

clump_annotate Same as plink –clump-annotate

clump_best Same as plink –clump-best

clump_field Same as plink –clump-field

clump_index_first
Same as plink –clump-index-first

clump_kb Same as plink –clump-kb

clump_p1 Same as plink –clump-p1

clump_p2 Same as plink –clump-p2

clump_r2 Same as plink –clump-r2

clump_range Same as plink –clump-range

clump_range_border
: Same as plink –clump-range-border

clump_replicate
: Same as plink –clump-replicate

clump_snp_field
: Same as plink –clump-snp-field

clump_verbose
: Same as plink –clump-verbose

cluster
: Same as plink –cluster

cluster_missing
: Same as plink –cluster-missing

cm_map
: Same as plink –cm-map

cnv_blue
: Same as plink –cnv-blue

cnv_border
: Same as plink –cnv-border

cnv_brown
: Same as plink –cnv-brown

cnv_check_no_overlap
: Same as plink –cnv-check-no-overlap

cnv_count
: Same as plink –cnv-count

cnv_del
: Same as plink –cnv-del

cnv_disrupt
: Same as plink –cnv-disrupt

cnv_drop_no_segment
: Same as plink –cnv-drop-no-segment

cnv_dup
: Same as plink –cnv-dup

cnv_enrichment_test
: Same as plink –cnv-enrichment-test

cnv_exclude
: Same as plink –cnv-exclude

cnv_exclude_off_by_1
: Same as plink –cnv-exclude-off-by-1

cnv_freq_excldue_above
: Same as plink –cnv-freq-excldue-above

cnv_freq_excldue_below
: Same as plink –cnv-freq-excldue-below

cnv_freq_excldue_exact
: Same as plink –cnv-freq-excldue-exact

cnv_freq_exclude_above
: Same as plink –cnv-freq-exclude-above

cnv_freq_exclude_below
: Same as plink –cnv-freq-exclude-below

cnv_freq_exclude_exact
: Same as plink –cnv-freq-exclude-exact

cnv_freq_incldue_exact
: Same as plink –cnv-freq-incldue-exact

`cnv_freq_include_exact`
: Same as plink –cnv-freq-include-exact

`cnv_freq_method2`
: Same as plink –cnv-freq-method2

`cnv_freq_overlap`
: Same as plink –cnv-freq-overlap

`cnv_green` Same as plink –cnv-green

`cnv_indiv_perm` Same as plink –cnv-indiv-perm

`cnv_intersect` Same as plink –cnv-intersect

`cnv_kb` Same as plink –cnv-kb

`cnv_list` Same as plink –cnv-list

`cnv_make_map` Same as plink –cnv-make-map

`cnv_max_kb` Same as plink –cnv-max-kb

`cnv_max_score` Same as plink –cnv-max-score

`cnv_max_sites` Same as plink –cnv-max-sites

`cnv_overlap` Same as plink –cnv-overlap

`cnv_red` Same as plink –cnv-red

`cnv_region_overlap`
: Same as plink –cnv-region-overlap

`cnv_report_regions`
: Same as plink –cnv-report-regions

`cnv_score` Same as plink –cnv-score

`cnv_seglist` Same as plink –cnv-seglist

`cnv_sites` Same as plink –cnv-sites

`cnv_subset` Same as plink –cnv-subset

`cnv_test` Same as plink –cnv-test

`cnv_test_1sided`
: Same as plink –cnv-test-1sided

`cnv_test_2sided`
: Same as plink –cnv-test-2sided

`cnv_test_region`
: Same as plink –cnv-test-region

`cnv_test_window`
: Same as plink –cnv-test-window

`cnv_track` Same as plink –cnv-track

`cnv_union_overlap`
: Same as plink –cnv-union-overlap

`cnv_unique` Same as plink –cnv-unique

`cnv_verbose_report_regions`
: Same as plink –cnv-verbose-report-regions

`cnv_write` Same as plink –cnv-write

`cnv_write_freq` Same as plink –cnv-write-freq

`complement_sets`
Same as plink –complement-sets

`compound_genotypes`
Same as plink –compound-genotypes

`compress` Same as plink –compress

`condition` Same as plink –condition

`condition_list` Same as plink –condition-list

`consensus_match`
Same as plink –consensus-match

`const_fid` Same as plink –const-fid

`control` Same as plink –control

`counts` Same as plink –counts

`covar` Same as plink –covar

`covar_name` Same as plink –covar-name

`covar_number` Same as plink –covar-number

`cow` Same as plink –cow

`d` Same as plink –d

`data` Same as plink –data

`debug` Same as plink –debug

`decompress` Same as plink –decompress

`dfam` Same as plink –dfam

`distance` Same as plink –distance

`distance_exp` Same as plink –distance-exp

`distance_matrix`
Same as plink –distance-matrix

`dog` Same as plink –dog

`dominant` Same as plink –dominant

`dosage` Same as plink –dosage

`double_id` Same as plink –double-id

`dprime` Same as plink –dprime

`dummy` Same as plink –dummy

`dummy_coding` Same as plink –dummy-coding

`each_versus_others`
Same as plink –each-versus-others

`each_vs_others` Same as plink –each-vs-others

`epistasis` Same as plink –epistasis

`epistasis_summary_merge`
Same as plink –epistasis-summary-merge

`exclude` Same as plink –exclude

`exclude_before_extract` Same as plink –exclude-before-extract

`exclude_snp` Same as plink –exclude-snp

`exclude_snps` Same as plink –exclude-snps

`extract` Same as plink –extract

`fam` Same as plink –fam

`family` Same as plink –family

`fast_epistasis` Same as plink –fast-epistasis

`fid` Same as plink –fid

`file` Same as plink –file

`fill_missing_a2` Same as plink –fill-missing-a2

`filter` Same as plink –filter

`filter_cases` Same as plink –filter-cases

`filter_controls` Same as plink –filter-controls

`filter_females` Same as plink –filter-females

`filter_founders` Same as plink –filter-founders

`filter_males` Same as plink –filter-males

`filter_nonfounders` Same as plink –filter-nonfounders

`fisher` Same as plink –fisher

`flip` Same as plink –flip

`flip_scan` Same as plink –flip-scan

`flip_scan_threshold` Same as plink –flip-scan-threshold

`flip_scan_verbose` Same as plink –flip-scan-verbose

`flip_scan_window` Same as plink –flip-scan-window

`flip_scan_window_kb` Same as plink –flip-scan-window-kb

`flip_subset` Same as plink –flip-subset

`freq` Same as plink –freq

`freqx` Same as plink –freqx

`from` Same as plink –from

`from_bp` Same as plink –from-bp

`from_kb` Same as plink –from-kb

`from_mb` Same as plink –from-mb

`frqx` Same as plink –frqx

`fst` Same as plink –fst

`gap` Same as plink –gap

`gates` Same as plink –gates

`gc` Same as plink –gc

`gen` Same as plink –gen

`gene` Same as plink –gene

`gene_all` Same as plink –gene-all

`gene_list` Same as plink –gene-list

`gene_list_border` Same as plink –gene-list-border

`gene_report` Same as plink –gene-report

`gene_report_empty` Same as plink –gene-report-empty

`gene_report_snp_field` Same as plink –gene-report-snp-field

`gene_subset` Same as plink –gene-subset

`genedrop` Same as plink –genedrop

`genepi` Same as plink –genepi

`geno` Same as plink –geno

`genome` Same as plink –genome

`genome_full` Same as plink –genome-full

`genome_lists` Same as plink –genome-lists

`genome_minimal` Same as plink –genome-minimal

`genotypic` Same as plink –genotypic

`gfile` Same as plink –gfile

`gplink` Same as plink –gplink

`grm` Same as plink –grm

`grm_bin` Same as plink –grm-bin

`grm_gz` Same as plink –grm-gz

`group_avg` Same as plink –group-avg

`groupdist` Same as plink –groupdist

`gxe` Same as plink –gxe

`hap...` Same as plink –hap...

`hap` Same as plink –hap

`hap_assoc` Same as plink –hap-assoc

`hap_freq` Same as plink –hap-freq

hap_impute Same as plink –hap-impute

hap_max_phase Same as plink –hap-max-phase

hap_min_phase_prob
Same as plink –hap-min-phase-prob

hap_miss Same as plink –hap-miss

hap_phase Same as plink –hap-phase

hap_phase_wide Same as plink –hap-phase-wide

hap_pp Same as plink –hap-pp

hap_snps Same as plink –hap-snps

hap_tdt Same as plink –hap-tdt

hap_window Same as plink –hap-window

hard_call_threshold
Same as plink –hard-call-threshold

hardy2 Same as plink –hardy2

hardy Same as plink –hardy

help Same as plink –help

het Same as plink –het

hethom Same as plink –hethom

hide_covar Same as plink –hide-covar

homog Same as plink –homog

homozyg Same as plink –homozyg

homozyg_density
Same as plink –homozyg-density

homozyg_gap Same as plink –homozyg-gap

homozyg_group Same as plink –homozyg-group

homozyg_het Same as plink –homozyg-het

homozyg_include_missing
Same as plink –homozyg-include-missing

homozyg_kb Same as plink –homozyg-kb

homozyg_match Same as plink –homozyg-match

homozyg_snp Same as plink –homozyg-snp

homozyg_verbose
Same as plink –homozyg-verbose

homozyg_window_het
Same as plink –homozyg-window-het

homozyg_window_kb
Same as plink –homozyg-window-kb

homozyg_window_missing
Same as plink –homozyg-window-missing

homozyg_window_snp
: Same as plink –homozyg-window-snp

homozyg_window_threshold
: Same as plink –homozyg-window-threshold

horse
: Same as plink –horse

hwe
: Same as plink –hwe

hwe_all
: Same as plink –hwe-all

ibc
: Same as plink –ibc

ibm
: Same as plink –ibm

ibs_matrix
: Same as plink –ibs-matrix

ibs_test
: Same as plink –ibs-test

id_delim
: Same as plink –id-delim

id_dict
: Same as plink –id-dict

id_match
: Same as plink –id-match

iid
: Same as plink –iid

impossible
: Same as plink –impossible

impute_sex
: Same as plink –impute-sex

ind_major
: Same as plink –ind-major

indep
: Same as plink –indep

indep_pairphase
: Same as plink –indep-pairphase

indep_pairwise
: Same as plink –indep-pairwise

independent_effect
: Same as plink –independent-effect

indiv_sort
: Same as plink –indiv-sort

inter_chr
: Same as plink –inter-chr

interaction
: Same as plink –interaction

je_cellmin
: Same as plink –je-cellmin

keep
: Same as plink –keep

keep_allele_order
: Same as plink –keep-allele-order

keep_autoconv
: Same as plink –keep-autoconv

keep_before_remove
: Same as plink –keep-before-remove

keep_cluster_names
: Same as plink –keep-cluster-names

keep_clusters
: Same as plink –keep-clusters

keep_fam
: Same as plink –keep-fam

lambda
: Same as plink –lambda

lasso	Same as plink –lasso
lasso_select_covars	Same as plink –lasso-select-covars
ld	Same as plink –ld
ld_snp	Same as plink –ld-snp
ld_snp_list	Same as plink –ld-snp-list
ld_snps	Same as plink –ld-snps
ld_window	Same as plink –ld-window
ld_window_kb	Same as plink –ld-window-kb
ld_window_r2	Same as plink –ld-window-r2
ld_xchr	Same as plink –ld-xchr
lfile	Same as plink –lfile
liability	Same as plink –liability
linear	Same as plink –linear
list	Same as plink –list
list_23_indels	Same as plink –list-23-indels
list_all	Same as plink –list-all
logistic	Same as plink –logistic
lookup...	Same as plink –lookup...
lookup	Same as plink –lookup
lookup_gene	Same as plink –lookup-gene
lookup_list	Same as plink –lookup-list
loop_assoc	Same as plink –loop-assoc
maf	Same as plink –maf
maf_succ	Same as plink –maf-succ
make_bed	Same as plink –make-bed
make_founders	Same as plink –make-founders
make_grm	Same as plink –make-grm
make_grm_bin	Same as plink –make-grm-bin
make_grm_gz	Same as plink –make-grm-gz
make_just_bim	Same as plink –make-just-bim
make_just_fam	Same as plink –make-just-fam
make_perm_pheno	Same as plink –make-perm-pheno
make_pheno	Same as plink –make-pheno
make_rel	Same as plink –make-rel
make_set	Same as plink –make-set

make_set_border
: Same as plink –make-set-border

make_set_collapse_group
: Same as plink –make-set-collapse-group

make_set_complement_all
: Same as plink –make-set-complement-all

make_set_complement_group
: Same as plink –make-set-complement-group

map
: Same as plink –map

mat
: Same as plink –mat

match
: Same as plink –match

match_type
: Same as plink –match-type

matrix
: Same as plink –matrix

max
: Same as plink –max

max_maf
: Same as plink –max-maf

mc
: Same as plink –mc

mcc
: Same as plink –mcc

mcovar
: Same as plink –mcovar

mds_cluster
: Same as plink –mds-cluster

mds_plot
: Same as plink –mds-plot

me
: Same as plink –me

me_exclude_one
: Same as plink –me-exclude-one

memory
: Same as plink –memory

mendel
: Same as plink –mendel

mendel_duos
: Same as plink –mendel-duos

mendel_multigen
: Same as plink –mendel-multigen

merge
: Same as plink –merge

merge_equal_pos
: Same as plink –merge-equal-pos

merge_list
: Same as plink –merge-list

merge_mode
: Same as plink –merge-mode

merge_x
: Same as plink –merge-x

meta_analysis
: Same as plink –meta-analysis

meta_analysis_..._field
: Same as plink –meta-analysis-...-field

mfilter
: Same as plink –mfilter

mh
: Same as plink –mh

mhf
: Same as plink –mhf

`min` Same as plink –min

`mind` Same as plink –mind

`mishap_window` Same as plink –mishap-window

`missing` Same as plink –missing

`missing_code` Same as plink –missing-code

`missing_genotype` Same as plink –missing-genotype

`missing_phenotype` Same as plink –missing-phenotype

`missing_var_code` Same as plink –missing-var-code

`mlma` Same as plink –mlma

`mlma_loco` Same as plink –mlma-loco

`mlma_no_adj_covar` Same as plink –mlma-no-adj-covar

`model` Same as plink –model

`model_dom` Same as plink –model-dom

`model_gen` Same as plink –model-gen

`model_rec` Same as plink –model-rec

`model_trend` Same as plink –model-trend

`mouse` Same as plink –mouse

`mperm` Same as plink –mperm

`mperm_save` Same as plink –mperm-save

`mperm_save_all` Same as plink –mperm-save-all

`mpheno` Same as plink –mpheno

`must_have_sex` Same as plink –must-have-sex

`mwithin` Same as plink –mwithin

`neighbour` Same as plink –neighbour

`no_fid` Same as plink –no-fid

`no_parents` Same as plink –no-parents

`no_pheno` Same as plink –no-pheno

`no_sex` Same as plink –no-sex

`no_snp` Same as plink –no-snp

`no_x_sex` Same as plink –no-x-sex

`nonfounders` Same as plink –nonfounders

`nop` Same as plink –nop

`not_chr` Same as plink –not-chr

`nudge` Same as plink –nudge

null_group Same as plink –null-group

null_snp Same as plink –null-snp

oblig_cluster Same as plink –oblig-cluster

oblig_clusters Same as plink –oblig-clusters

oblig_missing Same as plink –oblig-missing

out Same as plink –out

output_chr Same as plink –output-chr

output_missing_genotype
Same as plink –output-missing-genotype

output_missing_phenotype
Same as plink –output-missing-phenotype

oxford_pheno_name
Same as plink –oxford-pheno-name

parallel Same as plink –parallel

parameters Same as plink –parameters

parentdt1 Same as plink –parentdt1

parentdt2 Same as plink –parentdt2

pat Same as plink –pat

pca Same as plink –pca

pca_cluster_names
Same as plink –pca-cluster-names

pca_clusters Same as plink –pca-clusters

ped Same as plink –ped

pedigree Same as plink –pedigree

perm Same as plink –perm

perm_batch_size
Same as plink –perm-batch-size

perm_count Same as plink –perm-count

pfilter Same as plink –pfilter

pheno Same as plink –pheno

pheno_merge Same as plink –pheno-merge

pheno_name Same as plink –pheno-name

pick1 Same as plink –pick1

plist Same as plink –plist

poo Same as plink –poo

pool_size Same as plink –pool-size

ppc Same as plink –ppc

ppc_gap Same as plink –ppc-gap

proxy_... Same as plink –proxy-...

proxy_assoc Same as plink –proxy-assoc

proxy_b_kb Same as plink –proxy-b-kb

proxy_b_maxsnp Same as plink –proxy-b-maxsnp

proxy_b_r2 Same as plink –proxy-b-r2

proxy_b_threshold
Same as plink –proxy-b-threshold

proxy_b_window Same as plink –proxy-b-window

proxy_dosage Same as plink –proxy-dosage

proxy_drop Same as plink –proxy-drop

proxy_flanking Same as plink –proxy-flanking

proxy_geno Same as plink –proxy-geno

proxy_genotypic_concordance
Same as plink –proxy-genotypic-concordance

proxy_glm Same as plink –proxy-glm

proxy_impute Same as plink –proxy-impute

proxy_impute_threshold
Same as plink –proxy-impute-threshold

proxy_kb Same as plink –proxy-kb

proxy_list Same as plink –proxy-list

proxy_maf Same as plink –proxy-maf

proxy_maxsnp Same as plink –proxy-maxsnp

proxy_mhf Same as plink –proxy-mhf

proxy_r2 Same as plink –proxy-r2

proxy_r2_no_filter
Same as plink –proxy-r2-no-filter

proxy_replace Same as plink –proxy-replace

proxy_show_proxies
Same as plink –proxy-show-proxies

proxy_sub_maxsnp
Same as plink –proxy-sub-maxsnp

proxy_sub_r2 Same as plink –proxy-sub-r2

proxy_tdt Same as plink –proxy-tdt

proxy_verbose Same as plink –proxy-verbose

proxy_window Same as plink –proxy-window

prune Same as plink –prune

q_score_file Same as plink –q-score-file

q_score_range Same as plink –q-score-range

qfam... Same as plink –qfam...

qmatch Same as plink –qmatch

qq_plot Same as plink –qq-plot

qt Same as plink –qt

qt_means Same as plink –qt-means

qual_geno_... Same as plink –qual-geno-...

qual_geno_max_threshold
Same as plink –qual-geno-max-threshold

qual_geno_scores
Same as plink –qual-geno-scores

qual_geno_threshold
Same as plink –qual-geno-threshold

qual_max_threshold
Same as plink –qual-max-threshold

qual_scores Same as plink –qual-scores

qual_threshold Same as plink –qual-threshold

r2 Same as plink –r2

r Same as plink –r

range Same as plink –range

rank Same as plink –rank

read_dists Same as plink –read-dists

read_freq Same as plink –read-freq

read_genome Same as plink –read-genome

read_genome_list
Same as plink –read-genome-list

read_genome_minimal
Same as plink –read-genome-minimal

recessive Same as plink –recessive

recode12 Same as plink –recode12

recode Same as plink –recode

recodeA Same as plink –recodeA

recodeAD Same as plink –recodeAD

recodeHV Same as plink –recodeHV

recode_allele Same as plink –recode-allele

recode_beagle Same as plink –recode-beagle

recode_bimbam Same as plink –recode-bimbam

recode_fastphase
Same as plink –recode-fastphase

recode_lgen Same as plink –recode-lgen

recode_rlist Same as plink –recode-rlist

recode_structure
Same as plink –recode-structure

`recode_vcf` Same as plink –recode-vcf

`recode_whap` Same as plink –recode-whap

`reference` Same as plink –reference

`reference_allele`
Same as plink –reference-allele

`regress_distance`
Same as plink –regress-distance

`regress_pcs` Same as plink –regress-pcs

`regress_rel` Same as plink –regress-rel

`rel_check` Same as plink –rel-check

`rel_cutoff` Same as plink –rel-cutoff

`remove` Same as plink –remove

`remove_cluster_names`
Same as plink –remove-cluster-names

`remove_clusters`
Same as plink –remove-clusters

`remove_fam` Same as plink –remove-fam

`rerun` Same as plink –rerun

`rice` Same as plink –rice

`sample` Same as plink –sample

`score` Same as plink –score

`score_no_mean_imputation`
Same as plink –score-no-mean-imputation

`script` Same as plink –script

`seed` Same as plink –seed

`set` Same as plink –set

`set_by_all` Same as plink –set-by-all

`set_collapse_all`
Same as plink –set-collapse-all

`set_hh_missing` Same as plink –set-hh-missing

`set_max` Same as plink –set-max

`set_me_missing` Same as plink –set-me-missing

`set_missing_nonsnp_ids`
Same as plink –set-missing-nonsnp-ids

`set_missing_snp_ids`
Same as plink –set-missing-snp-ids

`set_missing_var_ids`
Same as plink –set-missing-var-ids

`set_names` Same as plink –set-names

`set_p` Same as plink –set-p

`set_r2` Same as plink –set-r2

`set_r2_phase` Same as plink –set-r2-phase

`set_table` Same as plink –set-table

`set_test` Same as plink –set-test

`sex` Same as plink –sex

`sheep` Same as plink –sheep

`show_tags` Same as plink –show-tags

`silent` Same as plink –silent

`simulate` Same as plink –simulate

`simulate_haps` Same as plink –simulate-haps

`simulate_label` Same as plink –simulate-label

`simulate_missing` Same as plink –simulate-missing

`simulate_n` Same as plink –simulate-n

`simulate_ncases` Same as plink –simulate-ncases

`simulate_ncontrols` Same as plink –simulate-ncontrols

`simulate_prevalence` Same as plink –simulate-prevalence

`simulate_qt` Same as plink –simulate-qt

`simulate_tags` Same as plink –simulate-tags

`snp` Same as plink –snp

`snps` Same as plink –snps

`snps_only` Same as plink –snps-only

`specific_haplotype` Same as plink –specific-haplotype

`split_x` Same as plink –split-x

`standard_beta` Same as plink –standard-beta

`subset` Same as plink –subset

`swap_parents` Same as plink –swap-parents

`swap_sibs` Same as plink –swap-sibs

`swap_unrel` Same as plink –swap-unrel

`tab` Same as plink –tab

`tag_kb` Same as plink –tag-kb

`tag_mode2` Same as plink –tag-mode2

`tag_r2` Same as plink –tag-r2

`tail_pheno` Same as plink –tail-pheno

`tdt` Same as plink –tdt

test_all	Same as plink –test-all
test_mishap	Same as plink –test-mishap
test_missing	Same as plink –test-missing
test_snp	Same as plink –test-snp
tests	Same as plink –tests
tfam	Same as plink –tfam
tfile	Same as plink –tfile
thin	Same as plink –thin
thin_count	Same as plink –thin-count
threads	Same as plink –threads
to	Same as plink –to
to_bp	Same as plink –to-bp
to_kb	Same as plink –to-kb
to_mb	Same as plink –to-mb
tped	Same as plink –tped
transpose	Same as plink –transpose
trend	Same as plink –trend
tucc	Same as plink –tucc
twolocus	Same as plink –twolocus
unbounded	Same as plink –unbounded
unrelated_heritability	Same as plink –unrelated-heritability
update_alleles	Same as plink –update-alleles
update_chr	Same as plink –update-chr
update_cm	Same as plink –update-cm
update_ids	Same as plink –update-ids
update_map	Same as plink –update-map
update_name	Same as plink –update-name
update_parents	Same as plink –update-parents
update_sex	Same as plink –update-sex
vcf	Same as plink –vcf
vcf_filter	Same as plink –vcf-filter
vcf_half_call	Same as plink –vcf-half-call
vcf_idspace_to	Same as plink –vcf-idspace-to
vcf_min_qual	Same as plink –vcf-min-qual
vegas	Same as plink –vegas
version	Same as plink –version

vif	Same as plink –vif
whap	Same as plink –whap
window	Same as plink –window
with_freqs	Same as plink –with-freqs
with_phenotype	Same as plink –with-phenotype
with_reference	Same as plink –with-reference
within	Same as plink –within
write_cluster	Same as plink –write-cluster
write_covar	Same as plink –write-covar
write_dosage	Same as plink –write-dosage
write_set	Same as plink –write-set
write_set_r2	Same as plink –write-set-r2
write_snplist	Same as plink –write-snplist
xchr_model	Same as plink –xchr-model
zero_cluster	Same as plink –zero-cluster
zero_cms	Same as plink –zero-cms
one	Same as plink –1
twothreefile	Same as plink –23file
stdout	Passed to system2, see its documentation.
stderr	Passed to system2, see its documentation.
wait	Logical. If FALSE, the plink process will fork into the background.

`plTrim` *Trim plink files*

Description

This function calculates number of individuals in .fam file (n1) and number of individuals in phenotype file (n2). If n1 > n2, then all the individuals not included in the phenotype file will be removed from plink files.

Usage

```
plTrim(pl_gwas, suffix = "trimmed")
```

Arguments

pl_gwas	PlGwasC object.
suffix	character. Suffix to the new plink file names.

Value

PlGwasC object

Author(s)

Kaiyin Zhong, Fan Liu

---

qq *QQ plot of one p-value vector*

Description

QQ plot of one p-value vector

Usage

```
qq(pvector)
```

Arguments

pvector	p-value vector

Value

A ggplot object

Author(s)

kaiyin

---

qq2 *QQ plot of two p-value vector*

Description

QQ plot of two p-value vector

Usage

```
qq2(p1, p2)
```

Arguments

p1	First p-value vector
p2	Second p-value vector

Value

A ggplot object

Author(s)

kaiyin

`qqmulti` *QQ plot of multiple p-value vectors*

Description

QQ plot of multiple p-value vectors

Usage

```
qqmulti(...)
```

Arguments

`...`	p-value vectors. These vectors don't have to have the same length.

Value

A ggplot object. One QQ plot for each p-value vector and they superposed one after another.

Author(s)

kaiyin

`randNormDat` *Generate a m by n data.frame from normal distribution*

Description

Generate a m by n data.frame from normal distribution

Usage

```
randNormDat(m, n)
```

Arguments

`m`	integer. Number of rows.
`n`	integer. Number of columns.

Author(s)

Kaiyin Zhong

`randomString` *Generate a single alpha-numeric random string*

Description

Generate a single alpha-numeric random string

Usage

```
randomString(string_length = 6)
```

Arguments

`string_length`	integer.

Value

character.

Author(s)

Kaiyin Zhong, Fan Liu

`randomStrings` *Generate random strings*

Description

Generate random strings

Usage

```
randomStrings(n, string_length = 6)
```

Arguments

`n`	integer. Number of string to generate.
`string_length`	integer. Length of each string.

Value

character.

Author(s)

Kaiyin Zhong, Fan Liu

`rbedInfo` *Constructor of RbedInfoC class*

Description

Constructor of RbedInfoC class

Usage

```
rbedInfo(bedstem, db_setup = FALSE)
```

Arguments

`bedstem`	character. Path to bed file without extension.
`db_setup`	logical. Whether to setup SQLite database for .bim, .fam and .frq files.

Value

An RbedInfoC object.

Author(s)

Kaiyin Zhong, Fan Liu

`RbedInfoC-class` *S4 class for necessary info to read a bed file into R*

Description

S4 class for necessary info to read a bed file into R

Slots

`pl_info` PlInfoC object

`jbed` jobjRef object, of Bed class in java

`nsnp` numeric. Number of SNPs.

`nindiv` numeric. Number of individuals.

`nindiv_appr` numeric. Apparent number of individuals.

`bytes_snp` numeric. Number of bytes used for each SNP.

read.phe.table *Read phenotype file*

Description

Read phenotype file

Usage

```
read.phe.table(file)
```

Arguments

file	character, path to phenotype file.

Value

data.frame

Author(s)

kaiyin

readAssoc *Read PLINK .assoc files*

Description

Read PLINK .assoc files

Usage

```
readAssoc(filename, cn_select = collenv$.assoc_header)
```

Arguments

filename	character. Filename
cn_select	character. Columns to read.

Value

data.frame

Author(s)

Kaiyin Zhong

readBed *Read genotypes from PLINK bed file into R*

Description

Read genotypes from PLINK bed file into R

Usage

```
readBed(rbed_info, snp_vec, fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,ANY,logical,logical'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,missing,missing,missing'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,ANY,missing,missing'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,missing,logical,missing'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,ANY,logical,missing'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,missing,missing,logical'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,ANY,missing,logical'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)

## S4 method for signature 'RbedInfoC,missing,logical,logical'
readBed(rbed_info, snp_vec,
  fid_iid = TRUE, snp_names_as_colnames = TRUE)
```

Arguments

rbed_info	RbedInfoC object

snp_vec	numeric. Vector of SNP index. Either row numbers in the bim file or a vector of SNP names.
fid_iid	logical. Whether the FID and IID columns should be included.
snp_names_as_colnames	logical. Whether SNP names should be used as colnames in the returned data frame

Value

data.frame Genotype data from bed file.

Author(s)

Kaiyin Zhong, Fan Liu

---

readBim *Read plink .bim files*

Description

Read plink .bim files

Usage

```
readBim(filename, cn_select = "..all")
```

Arguments

filename	.bim file path
cn_select	a character vector for selected colnames

Value

a data.frame

Author(s)

Kaiyin Zhong, Fan Liu

readBmBin *Read columns into an R matrix from a big.matrix .bin file*

Description

Read columns into an R matrix from a big.matrix .bin file

Usage

```
readBmBin(bin_file, ncols_to_read)
```

Arguments

bin_file	character. Path to .bin file
ncols_to_read	integer.

Value

matrix

Author(s)

Kaiyin Zhong, Fan Liu

readDesc *Read big.matrix .desc file*

Description

Read big.matrix .desc file

Usage

```
readDesc(desc_filename)
```

Arguments

desc_filename	character. Path to .desc file

Value

description object

Author(s)

Kaiyin Zhong, Fan Liu

readFam *Read plink .fam files*

Description

Read plink .fam files

Usage

```
readFam(filename, cn_select = "..all")
```

Arguments

filename	.fam file path
cn_select	a character vector for selected colnames

Value

a data.frame

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
bim = readBim("mmp13.bim")
bim1 = readBim("mmp13.bim", "..all")
fam = readFam("mmp13.fam", "..all")

## End(Not run)
```

readFunFactory *Generate read_fun for ReadInfo class*

Description

Generate read_fun for ReadInfo class

Usage

```
readFunFactory(header)
```

Arguments

header	logical. Whether the input file has a header line.

Value

function.

Author(s)

Kaiyin Zhong, Fan Liu

---

readGwasOut *Read GWAS output from plink If the GWAS is finished, returns a data.frame, otherwise returns NULL.*

Description

Read GWAS output from plink If the GWAS is finished, returns a data.frame, otherwise returns NULL.

Usage

```
readGwasOut(pl_gwas, cn_select = "..all", rmGwasOut = TRUE)
```

Arguments

pl_gwas	PlGwasC object.
cn_select	Colnames to select. Default to "..all"
rmGwasOut	Logical. Whether to remove GWAS output files after finished reading them. Default to TRUE.

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

readInfo *ReadInfo constructor*

Description

This function takes a file path as parameter, assuming the file is whitspace delimited, not quoted, and has a header line. It returns a ReadInfo object.

Usage

```
readInfo(filename, cnames)

## S4 method for signature 'character,missing'
readInfo(filename)

## S4 method for signature 'character,character'
readInfo(filename, cnames)
```

Arguments

filename	Path of the file to read
cnames	character. Expected column names (header).

Value

ReadInfo object

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
ri = readInfo("mmp13.frq")
ri@cnames
ri@filename
ri@header

## End(Not run)
```

ReadInfo-class *An S4 class to represent information about a whitespace-delimited text file to be read into R*

Description

An S4 class to represent information about a whitespace-delimited text file to be read into R

Slots

`filename` Path of the file

`cnames` character vector of column names

`header` logical. Whether the first line is header

`read_fun` function. The function to be used when reading this file

readLiteral *Read a file literally (all columns as character)*

Description

Read a file literally (all columns as character)

Usage

```
readLiteral(filename, ...)
```

Arguments

`filename`	Path of file to be read
`...`	Passed to `read.table`

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
df = data.frame(x = c("T", "%T", "10341"),
y = c("F", "f%t", "431"),
z = c("T", "TRUE", "FALSE"))
tmpf = tempfile()
write.table(df, file = tmpf, quote = FALSE,
row.names = FALSE, col.names = FALSE)
df1 = readLiteral(file = tmpf)
all(df1 == df)

## End(Not run)
```

`readLogistic` *Read PLINK logistic regression output files.*

Description

Read PLINK logistic regression output files.

Usage

```
readLogistic(filename, cn_select = collenv$.linear_header)
```

Arguments

`filename`	character. Filename.
`cn_select`	character. Columns to read.

Value

data.frame

Author(s)

Kaiyin Zhong

readPhe *Read phenotype file*

Description

Read phenotype file

Usage

```
readPhe(pl_gwas, cn_select = "..all")
```

Arguments

pl_gwas	PlGwasC object
cn_select	Colnames to select. Default to "..all", which means all columns are read in.

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

readPlinkOut *Read plink output files*

Description

Read plink output files

Usage

```
readPlinkOut(filename, ...)
```

Arguments

filename	Filenames of plink output files, see collenv$.plink_out_ext
...	passed to one of readAssoc, readQassoc, readLinear, readLogistic

Value

data.frame

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
dat1 = readPlinkOut("assoc/mmp13.assoc")
dat2 =    readAssoc("assoc/mmp13.assoc")
all(na.omit(dat1 == dat2))
dat1 = readPlinkOut("assoc/mmp13.assoc", c("CHR", "SNP", "P", "OR"))
dat2 =    readAssoc("assoc/mmp13.assoc", c("CHR", "SNP", "P", "OR"))
all(na.omit(dat1 == dat2))
dat1 = readPlinkOut("assoc/mmp13.qassoc")
dat2 =   readQassoc("assoc/mmp13.qassoc")
all(na.omit(dat1 == dat2))
dat1 = readPlinkOut("assoc/mmp13.qassoc", c("CHR", "SNP", "P", "R2"))
dat2 =   readQassoc("assoc/mmp13.qassoc", c("CHR", "SNP", "P", "R2"))
all(na.omit(dat1 == dat2))

## End(Not run)
```

---

readQassoc *Read .qassoc files*

Description

Read .qassoc files

Usage

```
readQassoc(filename, cn_select = collenv$.qassoc_header)
```

Arguments

filename	Path of the file to read
cn_select	a character vector for selected colnames

Value

data.frame.

Author(s)

Kaiyin Zhong

realBedSize *File size of bed file*

Description

File size of bed file

Usage

```
realBedSize(rbed_info)
```

Arguments

rbed_info	RbedInfoC object

Value

numeric. Size of bed file.

Author(s)

Kaiyin Zhong, Fan Liu

removeTag *Remove GWAS results by tag*

Description

Remove GWAS results by tag

Usage

```
removeTag(x, type = "gwas")
```

Arguments

x	character. Tag name.
type	character. Type of tag.

Author(s)

Kaiyin Zhong, Fan Liu

reprClasses *Represent classes of a data.frame in a character vector*

Description

Represent classes of a data.frame in a character vector

Usage

```
reprClasses(dat)
```

Arguments

dat	data.frame

Value

character vector

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
dat = randNormDat(4, 2)
x = capture.output(reprClasses(dat), file = NULL)
x = eval(parse(text = x))
all(x == colClasses(dat))

## End(Not run)
```

rmFilesByStem *Remove files by matching the starting part*

Description

If x is a string, then this function matches x* by globbing. If x is a "PlInfoC" object, it matches x@plink_stem*, If x is a "RbedInfoC" object, it matches x@pl_info@plink_stem*. Otherwise nothing is removed.

Usage

```
rmFilesByStem(x)
```

Arguments

x	character, PlInfoC, or RbedInfoC object.

Author(s)

Kaiyin Zhong, Fan Liu

---

runGcdh *Run GCDH analysis*

Description

Runs GCDH over the given PlGwasC object. The PlGwasC object is first filtered by p-values from a `plink --assoc` run if a p-value threshold is given. New PlGwasC objects are generated by shifting the PLINK bed file (e.g. shift1.bed, shift2.bed, ...) one by one. A GWAS is run for each of these PlGwasC objects and results are collected into big.matrix files.

Usage

```
runGcdh(pl_gwas, n_shift, gwas_col_select = NULL, collapse_matrix = NULL,
  rm_shifted_files = TRUE, dist_threshold = 5e+05)
```

Arguments

pl_gwas	PlGwasC object
n_shift	integer. Maximum shift number.
gwas_col_select	character. Columns to read from a GWAS output file. Default to `collenv$.linear_header_default`
collapse_matrix	matrix. 4 by 4 matrix used for generating collapsed genotypes.
rm_shifted_files	logical. Whether to remove shifted bed files after analysis is done.
dist_threshold	integer. SNPs beyond this distance will be ignored. Default to 500kb.

Value

A list with the following members: (1) the input PlGwasC object. (2) an info data frame with CHR, BP and SNP columns. (3) One big.matrix object for each of the names in `gwas_col_select`

Author(s)

Kaiyin Zhong, Fan Liu

runGwas *Run a GWAS*

Description

Run a GWAS

Usage

```
runGwas(pl_gwas, wait = TRUE, save_pl_gwas = FALSE)
```

Arguments

pl_gwas	PlGwasC object
wait	logical. Wait until GWAS is finished if this is set to TRUE. Default to FALSE.
save_pl_gwas	logical. Whether to save the plGwas object. Default to FALSE.

Author(s)

Kaiyin Zhong, Fan Liu

runTypeI *Run simulations to control type-I error*

Description

Simulate a new phenotype N times and run GCDH with each. The phe_fun function is used to generate new phenotype file. When this function is not given, the phenotype file from the PlGwasC object will be permuted and used as the new phenotype file (permutation analysis). Thus when no phe_fun is supplied, this function can be used to survey p-values under the null distribution. A threshold for Genome-wide significance can be calculated from these p-values by 5 any other alpha-level) quantile.

Usage

```
runTypeI(pl_gwas, n_shift, n_simu, phe_fun = NULL, dist_threshold = 5e+05,
  p_threshold = NULL, collapse_matrix = NULL, rm_shifted_files = TRUE)
```

Arguments

pl_gwas	PlGwasC object
n_shift	integer. n_shift for each GCDH run.
n_simu	integer. Number of simulations to run.
phe_fun	function. Used to generate new phenotype file.
dist_threshold	See runGcdh.
p_threshold	numeric or NULL. When it's not NULL, the PlGwasC object is filtered by assocFilter first.
collapse_matrix	See runGcdh.
rm_shifted_files	See runGcdh.

Value

A list with the following members: (1) tag of this simulation, can be used to remove related files. (2) a list of SNP pairs. If "snp_pair" is a member of the result from phe_fun, then this list will be non-empty, otherwise it will be empty. (3) a list of reports from all the GCDH analysis. (4) global minimal p-values of the single-SNP approach. (4) global minimal p-values of GCDH.

Author(s)

Kaiyin Zhong, Fan Liu

---

saveDesc *Save big.matrix description object to disk*

Description

Binary format is used exclusively.

Usage

```
saveDesc(desc_obj, desc_filename)
```

Arguments

desc_obj	big.matrix description object
desc_filename	character. Output file description file path.

Author(s)

Kaiyin Zhong, Fan Liu

sendQuery *Send query to SQLite database*

Description

Send query to SQLite database

Usage

```
sendQuery(db_name, query_string)
```

Arguments

db_name	character. Path to database.
query_string	character. Query string.

Author(s)

Kaiyin Zhong, Fan Liu

setOptModel *Set analysis model*

Description

Set analysis model

Usage

```
setOptModel(pl_gwas, mod = "linear")
```

Arguments

pl_gwas	PlGwasC object.
mod	character. One of "linear", "logistic" or "assoc", default to "linear".

Value

PlGwasC object

Author(s)

Kaiyin Zhong, Fan Liu

`setup` *Setup up a directory containing plink files*

Description

Setup up a directory containing plink files

Usage

```
setup(pl_info)
```

Arguments

`pl_info`	PlInfoC object

Author(s)

Kaiyin Zhong, Fan Liu

`setupRbed` *Setup an RbedInfoC object*

Description

The setup job includes the following tasks: 1. Set up the PlInfoC object. 2. Calculate number of bytes used by each SNP. 3. Calculate the Number of individuals. 4. Calculate total number of SNPs. 5. Validate the RbedInfoC object.

Usage

```
setupRbed(rbed_info)
```

Arguments

`rbed_info`	RbedInfoC object

Value

RbedInfoC object

Author(s)

Kaiyin Zhong, Fan Liu

`shiftBed` *Shift bed files*

Description

Generates collapsed genotypes by shifting the bed file (i.e. SNP1 collapsed with SNP2, SNP2 collapsed with SNP3, etc, when `n_shift == 1`).

Usage

```
shiftBed(rbed_info, n_shift, db_setup = FALSE, collapse_matrix = NULL)
```

Arguments

`rbed_info`	RbedInfoC object
`n_shift`	integer.
`db_setup`	logical. Whether to setup SQLite database for .bim, .fam and .frq files.
`collapse_matrix`	matrix of integers. See details.

Details

Collapsing matrix. The collapse_matrix parameter allows collapsing of two genotypes in a arbitrary way. Each genotype is represented by either 0, 1, 2, or 3:

0 Homozygote of the minor allele.

1 NA

2 Heterozygote.

3 Homozygote of the major allele.

The collapsing function is implemented as a matrix lookup function, i.e. $Collapse(S1, S2) = CollapseMatrix[S1][S2]$.

The default collapsing matrix is:

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 3 \\ 0 & 1 & 3 & 3 \end{matrix}$$

Value

RbedInfoC object, with the shifted bed file path in it.

Author(s)

Kaiyin Zhong, Fan Liu

shiftedStem *Add a "shift" suffix to a stem*

Description

Add a "shift" suffix to a stem

Usage

```
shiftedStem(stem, n_shift)
```

Arguments

stem	character.
n_shift	numeric.

Value

character.

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
# add suffix to stem
shiftedStem("a", 100) == "a_shift_0100"
shiftedStem("home/a", 100) == "home/a_shift_0100"
shiftedStem("/home/a", 100) == "/home/a_shift_0100"
shiftedStem(c("/home/a", "/home/b"), 100) == c("/home/a_shift_0100",
"/home/b_shift_0100")

## End(Not run)
```

slurp *Read a text file into a single string*

Description

Read a text file into a single string

Usage

```
slurp(filename)
```

Arguments

filename	character. Input filename.

Value

character

Author(s)

Kaiyin Zhong, Fan Liu

---

snpPos *Retrive SNP positions from UCSU database*

Description

Retrive SNP positions from UCSU database

Usage

```
snpPos(snps, rm_underscore = TRUE, ref = c("hg18", "hg19"),
  snpdb = c("snp138", "snp137"))
```

Arguments

snps	A vector of SNP names
rm_underscore	Remove irregular chromosome names
ref	Either "hg18" or "hg19"
snpdb	Either "snp138" or "snp137"

Value

A data frame containing positions of given SNPs

Author(s)

kaiyin

snpRowId *Get row number of SNPs from their names*

Description

Get row number of SNPs from their names

Usage

```
snpRowId(pl_info, snp_names)
```

Arguments

pl_info	PlInfoC object.
snp_names	character. Vector of SNP names.

Value

integer. Vector of row numbers.

Author(s)

Kaiyin Zhong, Fan Liu

spit *Write strings to a file*

Description

Write strings to a file

Usage

```
spit(s, filename)
```

Arguments

s	character. Strings to write.
filename	character. Path to output file.

Author(s)

Kaiyin Zhong, Fan Liu

sqliteFilePl *SQLite file of a PlInfoC object*

Description

SQLite file of a PlInfoC object

Usage

```
sqliteFilePl(x)
```

Arguments

x	PlInfoC or PlGwasC object

Value

character. Path to SQLite database file.

Author(s)

Kaiyin Zhong, Fan Liu

stopFormat *Stop with format string*

Description

Stop with format string

Usage

```
stopFormat(...)
```

Arguments

...	passed to sprintf

Author(s)

Kaiyin Zhong, Fan Liu

strConcat *Concatenate a vector of strings*

Description

Concatenate a vector of strings

Usage

```
strConcat(ss, sep = "")
```

Arguments

ss	vector of strings
sep	a length-1 string used as separator, default to ""

Value

a string

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
strConcat(letters)
strConcat(letters, " ")

## End(Not run)
```

strVectorRepr *String Representation of a character vector*

Description

String Representation of a character vector

Usage

```
strVectorRepr(ss, print_out = FALSE, single_quote = TRUE,
  start_with_c = TRUE)
```

Arguments

ss	character.
print_out	logical. Whether to print out the string representation.
single_quote	Logical, whether to use single quote for wrap strings. Default to TRUE, when set to FALSE, double quote is used.
start_with_c	Logical, whether the representation should start with "c(", when set to FALSE, "(" is used. Default to TRUE.

Value

character.

Author(s)

Kaiyin Zhong, Fan Liu

Examples

```
## Not run:
strVectorRepr(letters[1:3]) == 'c("a", "b", "c")'
strVectorRepr(
  as.character(1:3)) == 'c("1", "2", "3")'
all(eval(parse(text = strVectorRepr(as.character(1:3)))) ==
  c("1", "2", "3"))

## End(Not run)
```

---

strVectorSQLRepr *String representation of a character vector for SQLite consumption*

Description

Transform a character vector (e.g. `c("a", "b")` into a string representation that can be used in a SQLite query (e.g. "('a', 'b')").

Usage

```
strVectorSQLRepr(vec, print_out = FALSE, single_quote = TRUE)
```

Arguments

vec	character.
print_out	logical. Print out the string representation when set to TRUE.
single_quote	logical. Whether to use single quote for each element. Use double quote if set to FALSE. Default to TRUE.

Author(s)

Kaiyin Zhong

`systemFormat` *Call system command with format string*

Description

Call system command with format string

Usage

```
systemFormat(...)
```

Arguments

`...`	passed to `sprintf`

Author(s)

Kaiyin Zhong, Fan Liu

`theoBedSize` *Theoretical size of bed file*

Description

Computed from dimensions of bim an fam files.

Usage

```
theoBedSize(rbed_info)
```

Arguments

`rbed_info`	RbedInfoC object

Value

numeric. Theoretical size of bed file.

Author(s)

Kaiyin Zhong, Fan Liu

validPhe *Validate a phenotype file*

Description

Validate a phenotype file

Usage

```
validPhe(phe_file, ...)
```

Arguments

phe_file	character. Phenotype file.
...	Passed to read.table

Value

FALSE when the file is invalid, or a data.frame when it is.

Author(s)

Kaiyin Zhong, Fan Liu

write.phe.table *Write a phenotype data.frame to file*

Description

Write a phenotype data.frame to file

Usage

```
write.phe.table(phe, file)
```

Arguments

phe	data.frame
file	character, path to phenotype file.

Author(s)

Kaiyin Zhong

Index