Package 'ARTP2'

November 30, 2018

Type Package

Title Pathway and Gene-Level Association Test

Version 0.9.45

Date 2018-11-30

Author Han Zhang, Bill Wheeler, Kai Yu, Yifan Yang, R Core Team

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Depends R (>= 3.4.0), stats, utils

Imports Formula, parallel, data.table (>= 1.9.4)

Enhances ARTP

Description Pathway and gene level association test using raw data or summary statistics.

License GPL-2 | GPL-3

URL https://github.com/zhangh12/ARTP2

BugReports https://github.com/zhangh12/ARTP2/issues

SystemRequirements C++11

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-11-30 21:30:03 UTC

R topics documented:

ARTP2-package *Pathway and Gene-Level Association Test*

Description

Pathway and gene level association test using raw data or summary statistics.

Details

Package:	ARTP2
Type:	Package
Version:	0.9.42
Date:	2018-02-05
License:	GPL-2 | GPL-3

It is increasingly recognized that pathway analyses, a joint test of association between the outcome and a group of single nucleotide polymorphisms (SNPs) within a biological pathway, could potentially complement single-SNP analysis and provide additional insights for the genetic architecture of complex diseases. Building upon existing P-value combining methods, we propose a class of highly flexible pathway analysis approaches based on an adaptive rank truncated product statistic that can effectively combine evidence of associations over different SNPs and genes within a pathway. The statistical significance of the pathway-level test statistics is evaluated using a highly efficient permutation algorithm that remains computationally feasible irrespective of the size of the pathway and complexity of the underlying test statistics for summarizing SNP- and gene-level associations.

The main functions in this package are `sARTP` when only summary level data are available, `rARTP` when genotype data are available, and `warm.start` for computing gene and pathway p-values when previously save information is available.

Author(s)

Han Zhang, Bill Wheeler, Kai Yu, Yifan Yang

Maintainer: Han Zhang <han.zhang2@nih.gov>

References

Zhang H, Wheeler W, Hyland LP, Yang Y, Shi J, Chatterjee N, Yu K. (2016) A powerful procedure for pathway-based meta-analysis using summary statistics identifies 43 pathways associated with type II diabetes in European populations. PLoS Genetics 12(6): e1006122

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N. (2009) Pathway analysis by adaptive combination of P-values. Genet Epidemiol 33(8): 700 - 709

Zhang H, Shi J, Liang F, Wheeler W, Stolzenberg-Solomon R, Yu K. (2014) A fast multilocus test with adaptive SNP selection for large-scale genetic association studies. European Journal of Human Genetics 22: 696 - 702

`data` *A dataset used in example of* `rARTP`.

Description

A data frame `data` with 6000 observations on subject IDs, outcome, covariates, and 152 SNPs.

Usage

```
data(data)
```

Examples

```
data(data)
head(data[, 1:7])
```

`exclude.snps` *Excluding SNPs from specified regions*

Description

`exclude.snps` is used to exclude SNPs from specified regions, for example, the GWAS hits and their neighbors.

Usage

```
exclude.snps(stat, excluded.regions)
```

Arguments

`stat`	a data frame containing at least two columns `Chr` and `Pos`, i.e., the chromosome numbers and base-pair position (bp units) of SNPs.
`excluded.regions`	a data frame specifying the regions to be excluded. It must contains columns `Chr`, `Start`, and `End`. The unit is base-pair (bp). SNPs within `[Start, End]` will be excluded.

Value

A data frame containing a subset of `stat` after excluding SNPs in specified regions.

`geno` *A vector of file names used in example of* `rARTP`.

Description

Names of 15 genotype files. Those files are plain text files, each stores genotypes data of a gene.

Usage

```
data(geno)
```

Examples

```
data(geno)
head(geno)
```

`inflation` *Computing the genomic control inflation factor from a given numeric vector*

Description

This function gives the inflation of a set of p-values (or 1-df chi-squared statistics).

Usage

```
inflation(p, is.p, na.rm = FALSE)
```

Arguments

`p`	a numeric vector. It can contain p-values or 1-df chi-squared statistics.
`is.p`	a logical value indicating whether `p` contain p-values or a vector of 1-df chi-squared statistics.
`na.rm`	a logical value indicating whether `NA` values should be stripped before the computation proceeds.

Value

The inflation factor is returned.

See Also

`meta`

Examples

```
set.seed(0)
p <- runif(10)
inflation(p, TRUE)
```

`meta` *Single-marker meta-analysis*

Description

Perform the single-marker meta-analysis with inflation adjustment. The inverse-variance weighting approach is used.

Usage

```
meta(summary.files, lambda = NULL, sel.snps = NULL, only.meta=TRUE, ambig.by.AF=FALSE)
```

Arguments

`summary.files`	a character vector of file names containing the summary results of SNPs included in one or multiple studies. Each file must be able to be read by `read.table`. Each file must have columns called "SNP", "RefAllele", "EffectAllele", "BETA", and at least one of "SE", "P".
`lambda`	a numeric vector of inflation factors. Each file in `summary.files` should have one inflation factor specified in `lambda`. `NULL` if inflation is not adjusted.
`sel.snps`	a character vector of SNPs to be used in meta-analysis. The default is `NULL`, i.e., all SNPs are used.
`only.meta`	`TRUE` if do not returned individual summary data. The default is TRUE.
`ambig.by.AF`	TRUE or FALSE to align SNPs with ambiguous alleles by allele frequency (see `options`). The default is FALSE.

Details

The inverse-variance method is used by assuming a fixed effect model. The standard error is rescaled by `sqrt{lambda}`.

Value

`meta` returns a list containing

`meta.stat`	a data frame of summary statistics from meta-analysis. The summary statistics of individual studies specified in `summary.files` are also returned in the data frame if `only.meta` is `FALSE`. The standard error of individual studies are rescaled by `sqrt{lambda}`.
`conf.snps`	a character vector of SNPs with conflictive allele information.

Examples

```
study1 <- system.file("extdata", package = "ARTP2", "study1.txt.gz")
study2 <- system.file("extdata", package = "ARTP2", "study2.txt.gz")
snps <- c('rs13266821', 'rs4824130', 'rs1792438', 'rs1000047',
          'rs1000017', 'rs6066771', 'rs12508128')

m1 <- meta(summary.files = c(study1, study2), lambda = c(1.10, 1.08),
           sel.snps = snps)
m2 <- meta(summary.files = c(study1, study2), lambda = c(1.10, 1.08),
           sel.snps = snps, only.meta=FALSE)

m1$conf.snps

m1$meta.stat
m2$meta.stat
```

`options` *options*

Description

The list to describe the options that are used in `sARTP`, `rARTP`. It will be set by function `options.default` by default.

Format

The format is a list.

`out.dir` output directory for temporary and output files. The default is the working directory `getwd`.

`id.str` character string that is appended to temporary file names. The default is "PID".

`seed` integer for random number generation. The default is 1.

Options for testing an association:

`method` 1 = AdaJoint, 2 = AdaJoint2, 3 = ARTP. The default is 3. It can also be 'AdaJoint', 'AdaJoint2', or 'ARTP'. The package will convert it into upper case, so for example, 'Adajoint' is also accepted. The ARTP method was the proposed in Yu et al. (2009) Genet Epi, while AdaJoint and AdaJoint2 methods were proposed in Zhang et al. (2014) EJHG. Note that AdaJoint2 could be more powerful if (1) two functional SNPs are negative correlated and have effects in the same direction; or (2) two functional SNPs are positively correlated and have opposite directions of their effects.

`nperm` the number of permutations. The default is 1E5.

`nthread` the number of threads for multi-threaded processors in Unix/Linux OS. The default is `detectCores()` to use all available processors.

Options for controlling data cleaning:

`snp.miss.rate` any SNP with missing rate greater than `snp.miss.rate` will be removed from the analysis. The default is 0.05.

`maf` any SNP with minor allele frequency less than `maf` will be removed from the analysis. The default is 0.05.

`HWE.p` any SNP with HWE exact p-value less than `HWE.p` will be removed from the analysis. The test is applied to the genotype data or reference data. The test is ignored if the imputed genotype are not encoded as 0/1/2. The default is 1E-5.

`gene.R2` a number between 0 and 1 to filter out SNPs that are highly correlated within each gene. The `cor` function will be called to compute the R^2 values between each pair of SNPs and remove one SNP with lower MAF in each pair with R^2 greater than `gene.R2`. The default is 0.95.

`chr.R2` a number between 0 and 1 to filter out SNPs that are highly correlated within each chromosome. The `cor` function will be called to compute the R^2 values between each pair of SNPs and remove one SNP with lower MAF in each pair with R^2 greater than `chr.R2`. The default is 0.95.

`gene.miss.rate` threshold to remove genes based on their missing rate. Genes with missing rate greater than `gene.miss.rate` will be removed from the analysis. The missing rate is calculated as the number of subjects with at least one missing genotype among all SNPs in the gene divided by the total number of subjects. The default is 1.0.

`rm.gene.subset` `TRUE` to remove genes which are subsets of other genes. The default is `TRUE`.

`turn.off.filters` a shortcut to turn off all SNP filters. If `TRUE`, it is equivalent to set `snp.miss.rate = 1`, `maf = 0`, `trim.huge.chr`, `gene.R2 = 1`, `chr.R2 = 1`, `huge.gene.R2 = 1`, `huge.chr.R2 = 1`, and `HWE.p = 0`. The default is `FALSE`.

`impute` `TRUE` to impute missing genotypes with the mean of a SNP. `FALSE` to use another way other than imputation to handle missing data when constructing the score statistics, which is considered to be more power but also more time-consuming. The default is `FALSE`. If the pathway is large and the missing rates are expected to be low, consider to set it to be `TRUE` manually for reducing computational burden. It could be beneficial in terms of power with `impute` set as `FALSE` if the missing rate is high, e.g., the data are combined from multiple studies, and a SNP has missing genotypes because it is not measured or successfully imputed in some of the participating studies.

`group.gap` an integer to regroup SNPs in a chromosome into independent groups. The unit is base-pair (bp). The position information will be collected from the fourth column of bim files. The default is `NULL`, i.e., regrouping is not performed.

`delete` `TRUE` to delete temporary files containing the test statistics for each gene. The default is `TRUE`.

`print` `TRUE` to print information to the console. The default is `TRUE`.

`tidy` the data frame `deleted.snps` in the returned object of `sARTP` containing information of SNPs excluded from the analysis and their reasons. Possible reason codes include `RM_BY_SNP_NAMES`, `RM_BY_REGIONS`, `NO_SUM_STAT`, `NO_RAW_GENO`, `NO_REF`, `SNP_MISS_RATE`, `SNP_LOW_MAF`, `SNP_CONST`, `SNP_HWE`, `GENE_R2`, `HUGE_GENE_R2`, `CHR_R2`, `HUGE_CHR`, `HUGE_CHR2`, `HUGE_CHR3`, `GENE_MISS_RATE`, `GENE_SUBSET`, `CONF_ALLELE_INFO`, `LACK_OF_ACCU_BETA`. Set `tidy` as `TRUE` to hide the SNPs with codes `NO_SUM_STAT` and `NO_REF`. The default is `TRUE`.

`save.setup` TRUE to save necessary data, e.g., working options, observed scores and covariance matrix, to local to repeat the analysis more quicly (skip loading and filtering data). It will be set to be TRUE if `only.setup` is TRUE. The default is FALSE.

`path.setup` character string of file name to save the setup for `warm.start` if `save.setup` is TRUE. The default is NULL so that it is set as `paste(out.dir, "/setup.", id.str, ".rda", sep = "")`.

`only.setup` TRUE if only the setup is needed while the testing procedure is not. The R code to create the setup uses single thread but the testing procedure can be multi-threaded. The best practice to use ARTP2 on a multi-threaded cluster is to firstly create the setup in single-thread mode, and then call the `warm.start` to compute the p-values in multiple-thread mode, which uses the saved setup at `path.setup` as input. `save.setup` will be set to be TRUE if `only.setup` is TRUE. The default is FALSE.

`keep.geno` TRUE if the reference genotypes of SNPs in pathway is returned. The default is FALSE.

`excluded.snps` character vector of SNPs to be excluded in the analysis. NULL if no SNP is excluded. The default is NULL.

`selected.snps` character vector of SNPs to be selected in the analysis. NULL if all SNPs are selected but other filters may be applied. The default is NULL.

`excluded.regions` data frame with three columns `Chr`, `Start`, `End`, or three columns `Chr`, `Pos`, `Radius`. The unit is base-pair (bp). SNPs within `[Start, End]` or `[Pos - Radius, Pos + Radius]` will be excluded. See `Examples` in sARTP. This option is only available for sARTP. The default is NULL.

`excluded.subs` character vector of subject IDs to be excluded in the analysis. These IDs must match with those in the second column (Individual ID) of the `fam` files in `reference`. The default is NULL.

`selected.subs` character vector of subject IDs to be selected in the analysis. These IDs must match with those in the second column (Individual ID) of the `fam` files in `reference`. The default is NULL.

`excluded.genes` character vector of genes to be excluded in the analysis. NULL if no gene is excluded. The default is NULL.

`meta` TRUE if return meta-analysis summary data from sARTP. The default is FALSE.

`ambig.by.AF` TRUE or FALSE to align SNPs with ambiguous alleles by allele frequency (see details). The default is FALSE.

Options for handling huge pathways:

`trim.huge.chr` oversized chromosomes could be further trimmed to accelerate the testing procedure. If TRUE the additional options below are in effect. The default is TRUE.

`huge.gene.size` a gene with number of SNPs larger than `huge.gene.size` will be further trimmed with `huge.gene.R2` if `trim.huge.chr` is TRUE. The default is 1000.

`huge.chr.size` a chromosome with number of SNPs larger than `huge.chr.size` will be further trimmed with `huge.chr.R2` if `trim.huge.chr` is TRUE. The default is 2000.

`huge.gene.R2` more stringent R^2 threshold to filter out SNPs in a gene. Similar to `gene.R2`. The default is `gene.R2` - 0.05.

`huge.chr.R2` more stringent R^2 threshold to filter out SNPs in a chromosome. Similar to `chr.R2`. The default is `chr.R2` - 0.05.

Options for gene-based test:

`inspect.snp.n` the number of candidate truncation points to inspect the top SNPs in a gene. The default is 5. (See `Details`)

`inspect.snp.percent` a value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top SNPs. The default is 0 so that the truncation points will be 1:`inspect.snp.n`. (See `Details`)

Options for pathway-based test:

`inspect.gene.n` the number of candidate truncation points to inspect the top genes in the pathway. The default is 10.

`inspect.gene.percent` a value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top genes. If 0 then the truncation points will be 1:`inspect.gene.n`. The default is 0.05.

Details

Order of removing SNPs, genes and subjects:
1. Apply the options `excluded.snps` and `selected.snps` if non-NULL. Code: `RM_BY_SNP_NAMES`.
2. Apply the option `excluded.regions` if non-NULL and if `sARTP` is used. Code: `RM_BY_REGIONS`.
3. Remove SNPs without summary statistics in `summary.files`. Code: `NO_SUM_STAT`; or remove SNPs without raw genotype data in `data` or `geno.files`. Code: `NO_RAW_GENO`.
4. Remove SNPs not in `bim` files in `reference` if `sARTP` is used. Code: `NO_REF`.
5. Remove SNPs with conflictive allele information in summary and reference data if `sARTP` is used. Code: `CONF_ALLELE_INFO`.
6. Remove SNPs with missing RAF or EAF if `sARTP` and `options$ambig.by.AF` are used. Code: `NO_VALID_EAF_RAF`.
7. Remove SNPs with high missing rate. Code: `SNP_MISS_RATE`.
8. Remove SNPs with low MAF. Code: `SNP_LOW_MAF`.
9. Remove constant SNPs. Code: `SNP_CONST`.
10. Remove SNPs fail to pass HWE test. Code: `SNP_HWE`.
11. Remove highly correlated SNPs within each gene. Code: `GENE_R2` or `HUGE_GENE_R2`.
12. Remove highly correlated SNPs within each chromosome. Code: `CHR_R2`, `HUGE_CHR`, `HUGE_CHR2` or `HUGE_CHR3`.
13. Remove genes with high missing rate. Code: `GENE_MISS_RATE`.
14. Remove genes which are subsets of other genes. Code: `GENE_SUBSET`.

Example truncation points defined by `inspect.snp.n` and `inspect.snp.percent`: Assume the number of SNPs in a gene is 100. Below are examples of the truncation points for different values of `inspect.snp.n` and `inspect.snp.percent`. Similar values are applied to `inspect.gene.n` and `inspect.gene.percent`.

inspect.snp.n	inspect.snp.percent	truncation points
1	0	1
1	0.05	5
1	0.25	25
1	1	100
2	0	1, 2

2	0.05	5, 10
2	0.25	25, 50
2	1	100
3	0.2	20, 40, 60

SNPs with ambiguous alleles:
A SNP with alleles A and T (or C and G) is ambiguous because the strand cannot be determined. Without strand information, it is sometimes better to match SNPs with ambiguous alleles by allele frequency instead of by matching the alleles. By default, this package matches all SNPs by alleles. If matching by allele frequency for the SNPs with ambiguous alleles is desired, then summary files must contain a variable called "RAF" (reference allele frequency) or a variable "EAF" (effect allele frequency).

See Also

`options.default`

Examples

```
options <- options.default()
str(options)
names(options)
```

`options.default` *options.default*

Description

A function to return default `options` in `sARTP` and `rARTP`. This page is for illustration. Users do not need to call this function explicitly.

Usage

```
options.default()
```

Value

A list of `options`.

See Also

`options`

Examples

```
options <- options.default()
str(options)
names(options)
```

pathway *A data frame used in example of* rARTP.

Description

A data frame pathway defining the pathway.

Usage

```
data(pathway)
```

Examples

```
data(pathway)
head(pathway)
```

rARTP *ARTP test for raw data*

Description

Calculate gene and pathway p-values using the ARTP test and raw genotype data

Usage

```
rARTP(formula, data, pathway, family, geno.files = NULL, lambda = 1.0,
      subset = NULL, options = NULL)
```

Arguments

formula	an object of class formula: a symbolic description of basic risk model to be fitted. Only the outcome and covariates are included. See more details of formula in glm.
data	a data frame containing the variables specified in formula. If geno.files is NULL, then it also contains genotypes.
pathway	a character of the name of file containing definition of a pathway. It must be able to be read by read.table and have columns called SNP, Gene, Chr. It also can be a data frame with the three columns. The SNP column can also have values of the form loc1-loc2, where loc1 and loc2 are base-pair locations denoting a region of SNPs to use.
family	a character taking values of 'gaussian' or 'binomial'.
geno.files	a character vector containing paths of plain text files containing the genotype data. Those files can be compressed as gz files and are able to be read by read.table. It can be a data frame with columns bed, bim, and fam. The data frame contains paths of (multiple sets of) PLINK files containing the genotype data. It can be NULL if all genotype data are already been put in data.

lambda	a numeric specifying inflation factor. The default is 1.0.
subset	an optional integer vector specifying a subset of observations in data. The default is NULL, i.e., all observations are used.
options	a list of options to control the test procedure. If NULL, default options will be used. See options.

Details

This function computes gene and pathway p-values when raw genotype data is available. The ARTP test modified from Yu et al. (2009) and AdaJoint test from Zhang et al. (2014) are released with this package. ARTP is the Adaptive Rank Truncated Product test.

The raw (i.e. individual-level) genotype data, can be encoded as 0, 1, or 2 (counts of effect allele), or any quantitative values (e.g., output from genotype imputation program).

Value

rARTP returns an object of class ARTP2. It is a list containing the following components:

pathway.pvalue	final pathway p-value accounting for multiple comparisons.
gene.pvalue	a data frame containing gene name, number of SNPs in the gene that were included in the analysis, chromosome name, and the p-value for the gene accounting for multiple comparisons.
pathway	a data frame defining the pathway that was actually tested after various filters applied.
model	a list containing detailed information of selected SNPs in each gene.
most.sig.genes	a character vector of genes selected by ARTP2. They are the most promising candidates, although their statistical significance is not guaranteed.
deleted.snps	a data frame containing SNPs excluded from the analysis and their reasons.
deleted.genes	a data frame containing genes excluded from the analysis because they are subsets of other remaining genes. Set options$rm.gene.subset to be FALSE to include all genes even if they are subsets of other genes.
options	a list of options used in the analysis. See options.
accurate	TRUE if options$nperm is large enougth to accurately estimate p-values, i.e., if the criteria sqrt(pvalue*(1-pvalue)/nperm)/pvalue < 0.1 is satisfied.
setup	a list containing necessary input for warm.start. It can be written to a file by using the function save, then its path can be the input of warm.start. It also contains a data frame of outcome and covariates that are specified in formula (setup$yx), a data frame of genotypes of SNPs in pathway (setup$raw.geno), and a formula object setup$formula corresponding to setup$yx, if options$keep.geno is TRUE.

References

Zhang H, Wheeler W, Hyland LP, Yang Y, Shi J, Chatterjee N, Yu K. (2016) A powerful procedure for pathway-based meta-analysis using summary statistics identifies 43 pathways associated with type II diabetes in European populations. PLoS Genetics 12(6): e1006122

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N. (2009) Pathway analysis by adaptive combination of P-values. Genet Epidemiol 33(8): 700 - 709

Zhang H, Shi J, Liang F, Wheeler W, Stolzenberg-Solomon R, Yu K. (2014) A fast multilocus test with adaptive SNP selection for large-scale genetic association studies. European Journal of Human Genetics 22: 696 - 702

See Also

options, warm.start, sARTP, example.

Examples

```
library(ARTP2)

## Load the sample data
data(data, package = "ARTP2")
head(data[, 1:7])

## Load a build-in data frame containing pathway definition
## it can also be the path of the file
data(pathway, package = "ARTP2")
head(pathway)

## Define the formula of base risk model
formula <- formula(case_control ~ sex + age + bmi + factor(study))

## binary outcome
family <- "binomial"

## Set the options.
## Accumulate signal from the top 5 SNPs in each gene
## 1e5 replicates of resampling to estimate the p-value
options <- list(inspect.snp.n = 5, nperm = 1e5,
                maf = .01, HWE.p = 1e-6,
                gene.R2 = .9,
                id.str = "unique-pathway-id",
                out.dir = getwd(), save.setup = FALSE)

## pathway test, can take a while
## data contains outcome, covariates and genotypes
# ret1 <- rARTP(formula, data = data, pathway, family, options = options)

# ret1$pathway.pvalue
## [1] 0.03218968 # Mac OS
## [1] 0.02188978 # Linux with 1 thread
## [1] 0.03455965 # Linux with 32 threads

## Mac OS
# head(ret1$gene.pvalue)
##     Gene Chr N.SNP      Pvalue
## 1  USP30  12    18 0.001319987
```

```
## 2  DCAF7  17     9 0.071644284
## 3   CANX   5    13 0.266337337
## 4  SOX12  20    15 0.349406506
## 5 CDKN2C   1     6 0.358031420
## 6   FEN1  11     4 0.415345847

## Linux with 1 thread
# head(ret1$gene.pvalue)
##     Gene Chr N.SNP      Pvalue
## 1  USP30  12    18 0.000899991
## 2  DCAF7  17     9 0.070219298
## 3   CANX   5    13 0.269772302
## 4  SOX12  20    15 0.350061499
## 5 CDKN2C   1     6 0.357766422
## 6   FEN1  11     4 0.414760852

## Linux with 32 threads
# head(ret1$gene.pvalue)
##     Gene Chr N.SNP      Pvalue
## 1  USP30  12    18 0.001454985
## 2  DCAF7  17     9 0.070379296
## 3   CANX   5    13 0.266927331
## 4  SOX12  20    15 0.350481495
## 5 CDKN2C   1     6 0.357701423
## 6   FEN1  11     4 0.414425856

# table(ret1$deleted.snps$reason)
# head(ret1$deleted.genes)


################################################
## Another way to use this function
## Load a vector 'geno' containing file names of genotype
data(geno, package = 'ARTP2')

## Set the paths of genotype files
## in this example, each file contains SNPs in a gene
geno.files <- system.file("extdata", package = "ARTP2", geno)

## data contains outcome, covariates
## Genotypes are instead included in files specified in geno.files
## geno.files are plain text files (or .gz file), which can be read by read.table
# ret2 <- rARTP(formula, data = data[, 2:6], pathway, family, geno.files,
#               options = options)
# ret2$pathway.pvalue == ret1$pathway.pvalue


################################################
## The third way
## Genotypes are instead stored as binary PLINK files (bed, bim, and fam)
bed <- system.file("extdata", package = "ARTP2", "raw.bed")
bim <- system.file("extdata", package = "ARTP2", "raw.bim")
fam <- system.file("extdata", package = "ARTP2", "raw.fam")
```

```
geno.files <- data.frame(fam, bim, bed, stringsAsFactors = FALSE)

## a column SUBID must be included in data, in this example, first column is SUBID
# ret3 <- rARTP(formula, data = data[, 1:6], pathway, family, geno.files,
#               options = options)
# ret3$pathway.pvalue == ret1$pathway.pvalue
```

`read.bed` *Reading data from binary PLINK files*

Description

Loads genotype data from PLINK format files .bed, .bim, and .fam.

Usage

```
read.bed(bed, bim, fam, sel.snps = NULL, sel.subs = NULL, encode012 = TRUE)
```

Arguments

`bed`	the name of the bed file.
`bim`	the name of the bim file. For a SNP without a rs number, use any character (including any white space or '.') in the second column of the bim file.
`fam`	the name of the fam file.
`sel.snps`	a character vector of SNPs to be extracted from the plink files. The default is `NULL`, i.e., all SNPs are extracted. SNPs could be named by its rs number (e.g. rs1234), or by `Chr:Pos` (e.g. 13:234567, or C13P234567) if a rs number is not available. All other naming methods for a SNP are not accepted in current version.
`sel.subs`	an optional character vector specifying a subset of subject IDs to be extracted from the plink files. These IDs should be matched with the second column of `fam` files. The default is `NULL`, i.e., all subjects are extracted.
`encode012`	logical. Encoding the genotypes using 0/1/2 if `TRUE`, or using symbols of the reference and effect alleles if `FALSE`. The default is `TRUE`.

Value

A data frame of genotypes of specified subjects in the plink files. For a SNP in `sel.snps` specified in the format `Chr:Pos`, e.g. 13:234567, it will be named to be C13P234567 in the returned data frame.

Examples

```
# Load the sample data

bed <- system.file("extdata", package = 'ARTP2', 'chr1.bed')
bim <- system.file("extdata", package = 'ARTP2', 'chr1.bim')
fam <- system.file("extdata", package = 'ARTP2', 'chr1.fam')

## first five SNPs
b <- read.table(bim, header = FALSE, as.is = TRUE, nrows = 5)
## first 50 subjects
f <- read.table(fam, header = FALSE, as.is = TRUE, nrows = 50)
geno <- read.bed(bed, bim, fam, sel.snps = b[, 2], sel.subs = f[, 2])

dim(geno) # 50 x 5
```

---

ref.does *Example reference genotype data coded as expected dosages.*

Description

A list with names `ref.geno` and `allele.info` containing genotypes coded as expected dosages and the corresponding allele information for each SNP.

Usage

```
data(ref.does)
```

Examples

```
data(ref.does)
head(ref.does$ref.geno[, 1:5])
head(ref.does$allele.info[, 1:5])
```

---

ref.geno *A dataset used in example of* sARTP.

Description

A data frame `ref.geno` with genotypes of 503 observations on 2654 SNPs. The genotypes are encoded by SNPs' minor and major alleles so that this data frame can be used as `reference` of function sARTP.

Usage

```
data(ref.geno)
```

Examples

```
data(ref.geno)
head(ref.geno[, 1:5])
```

---

sARTP *ARTP test for summary data*

Description

Calculate gene and pathway p-values using the ARTP test and summary data.

Usage

```
sARTP(summary.files, pathway, family, reference, lambda,
      ncases, ncontrols, nsamples, options = NULL)
```

Arguments

summary.files	a character vector of file names containing the summary results of SNPs included in one or multiple studies. Each file must be able to be read by read.table. Each file must have columns called SNP, RefAllele, EffectAllele, Beta, and at least one of SE, P. An optional column called Direction describing studies information can also be included if the summary results were calculated from multiple studies by inverse weighting method. Two optional columns called Chr and Pos are required if excluded.regions is specified in options. SNPs within excluded.regions are going to be excluded from the analysis. If options$ambig.by.AF is TRUE, then a column called "RAF" or "EAF" is required. See Details.
pathway	a character of the name of file containing definition of a pathway. It must be able to be read by read.table and have columns called SNP, Gene, and Chr. It can also be a data frame with the three columns. The SNP column can also have values of the form loc1-loc2, where loc1 and loc2 are base-pair locations denoting a region of SNPs to use.
family	a character taking values of 'gaussian' or 'binomial'.
reference	a data.frame containing the paths of binary PLINK files of reference dataset. It must have columns called bed, bim and fam. The current version allows users to specify multiple sets of bed/bim/fam PLINK files as separate rows of the data frame.
lambda	a numeric vector of inflation factors. Each file in summary.files should have one inflation factor specified in lambda.

ncases	a list of numeric vectors specifying sample sizes of cases of participating studies. This argument should be specified only if `family == 'binomial'`, otherwise `list()`. See `Examples`.
ncontrols	a list of numeric vectors specifying sample sizes of controls of participating studies. This argument should be specified only if `family == 'binomial'`, otherwise `list()`. See `Examples`.
nsamples	a list of numeric vectors specifying total sample sizes of participating studies. This argument should be specified only if `family == 'gaussian'`, otherwise `list()`. See `Examples`.
options	a list of options to control the test procedure. If `NULL`, default options will be used. See `options`.

Details

This function computes gene and pathway p-values when only summary data is available. Only the ARTP test modified from Yu et al. (2009) is well tested and is released with this package. ARTP is the Adaptive Rank Truncated Product test.

Each file in `summary.files` must contain

- `SNP` SNP name
- `RefAllele` reference allele. Can be different in studies
- `EffectAllele` effect allele. Can be different in studies
- `Beta` estimated effect in linear regression model or log odds ratio in logistic regression model

and must contain one of the optional columns

- `SE` estimated standard error of Beta
- `P` p-value of Wald's, LRT or score test for testing `H_0: Beta = 0`. Can be generated by `lm`, `glm`, `anova` in `R` or other standard statistical softwares.

An optional column `Direction` is encouraged to be provided by the user

- `Direction` a character vector indicating which studies include a SNP. Any symbol except for '?' means a SNP is included in that study. Please note that the real direction of a SNP in studies ('+' or '-') does not matter, e.g., '++-?+' and '**+?-' provide exact the same information. See `Examples`.

Another two optional columns `Chr` and `Pos` are needed if `excluded.regions` is specified in `options`. `ARTP2` will convert the column names to be upper case, so for example, either `Beta` or `BETA` or `beta` are accepted. See `Examples`.

- `Chr` chromosome.
- `Pos` base-pair position (bp units).

If the option `ambig.by.AF` is set to 1, then the summary files must contain at least one of:

- `RAF` reference allele frequency.
- `EAF` effect allele frequency.

The order of columns in files `summary.files`, `pathway` or in data frame `reference` are arbitrary, and all unnecessary columns (if any) are discarded in the analysis.

Value

sARTP returns an object of class ARTP2. It is a list containing the following components:

pathway.pvalue	final pathway p-value accounting for multiple comparisons.
gene.pvalue	a data frame containing gene name, number of SNPs in the gene that were included in the analysis, chromosome name, and the p-value for the gene accounting for multiple comparisons.
pathway	a data frame defining the pathway that was actually tested after various filters applied.
model	a list containing detailed information of selected SNPs in each gene.
most.sig.genes	a character vector of genes selected by ARTP2. They are the most promising candidates, although their statistical significance is not guaranteed.
deleted.snps	a data frame containing SNPs excluded from the analysis and their reasons.
deleted.genes	a data frame containing genes excluded from the analysis because they are subsets of other remaining genes. Set options$rm.gene.subset to be FALSE to include all genes even if they are subsets of other genes.
options	a list of options used in the analysis. See options
accurate	TRUE if options$nperm is large enougth to accurately estimate p-values, i.e., if the criteria sqrt(pvalue*(1-pvalue)/nperm)/pvalue < 0.1 is satisfied.
setup	a list containing necessary input for warm.start. It can be written to a file by using the function save, then its path can be the input of warm.start. Loading from reference, it also contains a data frame of genotypes of used SNPs (setup$ref.geno), if options$keep.geno is TRUE.

References

Zhang H, Wheeler W, Hyland LP, Yang Y, Shi J, Chatterjee N, Yu K. (2016) A powerful procedure for pathway-based meta-analysis using summary statistics identifies 43 pathways associated with type II diabetes in European populations. PLoS Genetics 12(6): e1006122

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N. (2009) Pathway analysis by adaptive combination of P-values. Genet Epidemiol 33(8): 700 - 709

Zhang H, Shi J, Liang F, Wheeler W, Stolzenberg-Solomon R, Yu K. (2014) A fast multilocus test with adaptive SNP selection for large-scale genetic association studies. European Journal of Human Genetics 22: 696 - 702

See Also

options, warm.start, rARTP

Examples

```
library(ARTP2)

## Path of files containing summary statistics
## Only required columns will be loaded
```

```
study1 <- system.file("extdata", package = "ARTP2", "study1.txt.gz")
study2 <- system.file("extdata", package = "ARTP2", "study2.txt.gz")

## Path of a build-in file containing pathway definition
pathway <- system.file("extdata", package = "ARTP2", "pathway.txt.gz")

## Create data frame containing paths of build-in PLINK files that are going to used as reference
## As an example, use all chromosomes
chr <- 1:22
nchr <- length(chr)

fam <- vector("character", nchr)
bim <- vector("character", nchr)
bed <- vector("character", nchr)

for(i in 1:nchr){
  fam[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".fam", sep = ""))
  bim[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".bim", sep = ""))
  bed[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".bed", sep = ""))
}

reference <- data.frame(fam, bim, bed, stringsAsFactors = FALSE)

## Set the options.
## Accumulate signal from the top 2 SNPs in each gene
## 1e5 replicates of resampling to estimate the p-value
options <- list(inspect.snp.n = 2, nperm = 1e4,
                maf = .01, HWE.p = 1e-6,
                gene.R2 = .9,
                id.str = "unique-pathway-id",
                out.dir = getwd(), save.setup = FALSE)

## different inflation factors are adjusted in two studies
lambda <- c(1.10, 1.08)

## two summary files, so there are two elements in each of two lists ncases and ncontrols
## the first summary file includes data calculated from meta-analysis of two sub-studies,
## each with sample size 63390 (9580 cases and 53810 controls) and 5643 (2591 cases and
## 3052 controls)
## see a few rows in study1
# s <- read.table(study1, header = TRUE, as.is = TRUE, nrows = 10)
# s$Direction
## [1] "+?" "+?" "++" "+?" "+?" "+?" "+?" "+?" "+?" "+?" "++"
## sub-study1 has 9580 cases, and sub-study2 has 2591 cases
## sub-study1 has 53810 cases, and sub-study2 has 3052 cases
## '?' means a SNP is not included in that sub-study
## any other symbols means a SNP is included in that sub-study
ncases <- list()
ncontrols <- list()
ncases[[1]] <- c(9580, 2591)
ncontrols[[1]] <- c(53810, 3052)

## the second summary file includes data calculated from one sub-studies with sample size
```

```
## 61957 (7638 cases and 54319 controls)
ncases[[2]] <- 7638
ncontrols[[2]] <- 54319

# logistic regression is used in base model, thus ncases and ncontrols should be specified.
family <- 'binomial'

## pathway test with two study files
# ret <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#              ncases, ncontrols, options = options)

# ret$pathway.pvalue
## [1] 0.04594541  # Mac OS
## [1] 0.05149485  # Linux with 1 thread
## [1] 0.03969603  # Linux with 32 threads

## Mac OS
# head(ret$gene.pvalue)
##       Gene Chr N.SNP      Pvalue
## 1     BDH2   4    10 0.000749925
## 2   UBE2D3   4     6 0.001849815
## 3     PBX2   6    22 0.003849615
## 4 PPP1R14D  15     9 0.003849615
## 5   MRPL10  17    18 0.011448855
## 6    SCYL1  11     3 0.019848015

## Linux with 1 thread
# head(ret$gene.pvalue)
##       Gene Chr N.SNP      Pvalue
## 1     BDH2   4    10 0.000949905
## 2   UBE2D3   4     6 0.001699830
## 3 PPP1R14D  15     9 0.003949605
## 4     PBX2   6    22 0.004299570
## 5   MRPL10  17    18 0.012448755
## 6    SCYL1  11     3 0.017148285

## Linux with 32 threads
# head(ret$gene.pvalue)
##       Gene Chr N.SNP      Pvalue
## 1   UBE2D3   4     6 0.000849915
## 2     BDH2   4    10 0.001049895
## 3 PPP1R14D  15     9 0.003949605
## 4     PBX2   6    22 0.004899510
## 5   MRPL10  17    18 0.012798720
## 6    SCYL1  11     3 0.015048495

## pathway test with each of two studies
# ret1 <- sARTP(summary.files = study1, pathway, family, reference, lambda[1],
#               ncases[1], ncontrols[1], options = options)

# ret2 <- sARTP(summary.files = study2, pathway, family, reference, lambda[2],
#               ncases[2], ncontrols[2], options = options)
```

```
# ret1$pathway.pvalue
## [1] 0.04279572  # Mac OS
## [1] 0.03519648  # Linux with 1 thread
## [1] 0.04644536  # Linux with 32 threads

# ret2$pathway.pvalue
## [1] 0.3092691  # Mac OS
## [1] 0.2870213  # Linux with 1 thread
## [1] 0.3010699  # Linux with 32 threads

################################################
## The reference is passed as an individual-level genotype data frame

data(ref.geno)
# ret.ref <- sARTP(summary.files = c(study1, study2), pathway, family, ref.geno, lambda,
#                  ncases, ncontrols, options = options)

# ret.ref$pathway.pvalue == ret$pathway.pvalue

################################################
## The reference genotype data can also be merged into a single set of PLINK files

bed <- system.file("extdata", package = "ARTP2", "ref.bed")
bim <- system.file("extdata", package = "ARTP2", "ref.bim")
fam <- system.file("extdata", package = "ARTP2", "ref.fam")

reference <- data.frame(fam, bim, bed)
# ret.comb <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#                   ncases, ncontrols, options = options)
# ret.comb$pathway.pvalue == ret$pathway.pvalue

################

## exclude some regions
exc.reg1 <- data.frame(Chr = c(1, 1, 22),
                       Pos = c(1706160, 11979231, 51052379),
                       Radius = c(5000, 0, 2000))
options$excluded.regions <- exc.reg1

# ret.exc1 <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#                   ncases, ncontrols, options = options)

# ret.exc1$pathway.pvalue
## [1] 0.04619538 # Mac OS
## [1] 0.0510449  # Linux with 1 thread
## [1] 0.04054595 # Linux with 32 threads

# sum(ret.exc1$deleted.snps$reason == 'RM_BY_REGIONS')

## or equivalently
exc.reg2 <- data.frame(Chr = c(1, 1, 22),
                       Start = c(1701160, 11979231, 51050379),
                       End = c(1711160, 11979231, 51054379))
```

```
options$excluded.regions <- exc.reg2

# ret.exc2 <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#                   ncases, ncontrols, options = options)
# ret.exc1$pathway.pvalue == ret.exc2$pathway.pvalue

#################

## select a subset of subjects in plink files as the reference
## options$selected.subs should be in the same format as the first column of fam file
## load character vector subj.id of 400 subjects from build-in dataset
data(subj.id, package = "ARTP2")
head(subj.id)
options$selected.subs <- subj.id
options$excluded.regions <- NULL

# ret.sel <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#                  ncases, ncontrols, options = options)
# ret.sel$pathway.pvalue
## [1] 0.03469653 # Mac OS
## [1] 0.05284472 # Linux with 1 thread
## [1] 0.04164584 # Linux with 32 threads
```

`subj.id` *A vector of subject IDs used in example of* `sARTP`.

Description

Subject IDs of 400 samples.

Usage

```
data(subj.id)
```

Examples

```
data(subj.id)
head(subj.id)
```

`warm.start` *Performing ARTP2 test with warm start*

Description

This function is designed to accelerate the `ARTP2` test in practice. It uses pre-calculated and reusable statistics as input and allow the users to try different testing configuration more efficiently. See `Details` for more information.

Usage

```
warm.start(setup, nperm = NULL, lambda = 1.0, nthread = NULL)
```

Arguments

setup	an R object created by sARTP or rARTP. It is a list containing necessary statistics for computing p-values.
nperm	the number of permutations. If it is NULL as default, then the value in the saved setup will be used. See Details.
lambda	inflation factor to be adjusted in pathway analysis. lambda in this function can only be a single numeric number, which is different from the one in sARTP. The default is 1.0.
nthread	number of threads to be used in permutation. NULL if setup$options$nthread is used.

Details

An ARTP2 test has two major steps in testing an association. The first step applys data clean criteria and creates necessary and reusable statistics, which can be time-consuming for large pathways. The second step performs the testing procedure to estimate the pathway or gene-level p-value. warm.start focuses on the second step.

The first step can be done by using sARTP or rARTP if their options$only.setup is set as TRUE. Their output object, setup, can be used as the first argument of warm.start. With warm.start, users can try different configurations to perform various tests allowed by the ARTP2 framework, but avoid long waiting time for data cleaning. Commonly used options in setup$options include method, inspect.snp.n, inspect.gene.n, nperm, etc.

Note that both sARTP and rARTP can produce the final p-value directly if options$only.setup is FALSE.

The setup is supposed to have all components defined in sARTP and rARTP. If nperm is NULL, then it will be set as setup$options$nperm. The users can also pass lambda if a second round genomic control is needed. However, unlike in sARTP, lambda here can only be a single numberic number rather than a vector. Options nperm and lambda are the most useful ones in using warm.start so we highlight them in the interface. Users can modify any option values in setup$options directly to get more controls of the testing procedure. See options for more details about how to set setup$options.

Except for setup$options, all other components in setup should not be modified by users.

Value

warm.start returns an object of class ARTP2. It is a list containing the following components:

pathway.pvalue	final pathway p-value accounting for multiple comparisons.
gene.pvalue	a data frame containing gene name, number of SNPs in the gene that were included in the analysis, chromosome name, and the p-value for the gene accounting for multiple comparisons.

pathway	a data frame defining the pathway that was actually tested after various filters applied.
model	a list containing detailed information of selected SNPs in each gene.
most.sig.genes	a character vector of genes selected by ARTP2. They are the most promising candidates, although their statistical significance is not guaranteed.
deleted.snps	a data frame containing SNPs excluded from the analysis and their reasons.
deleted.genes	a data frame containing genes excluded from the analysis because they are subsets of other remaining genes.
options	a list of options used in the analysis. See options
test.timing	timing information (in sec)
accurate	TRUE if options$nperm is large enougth to accurately estimate p-values, i.e., if the criteria sqrt(pvalue*(1-pvalue)/nperm)/pvalue < 0.1 is satisfied.

References

Zhang H, Wheeler W, Hyland LP, Yang Y, Shi J, Chatterjee N, Yu K. (2016) A powerful procedure for pathway-based meta-analysis using summary statistics identifies 43 pathways associated with type II diabetes in European populations. PLoS Genetics 12(6): e1006122

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N. (2009) Pathway analysis by adaptive combination of P-values. Genet Epidemiol 33(8): 700 - 709

Zhang H, Shi J, Liang F, Wheeler W, Stolzenberg-Solomon R, Yu K. (2014) A fast multilocus test with adaptive SNP selection for large-scale genetic association studies. European Journal of Human Genetics: 22, 696 - 702

See Also

sARTP, rARTP, options

Examples

```
## firstly, run the example in sARTP
## users can adjust the second round inflation in warm.start
## the first round inflation can be study-specific and adjusted in rARTP
## or sARTP

library(ARTP2)
study1 <- system.file("extdata", package = "ARTP2", "study1.txt.gz")
study2 <- system.file("extdata", package = "ARTP2", "study2.txt.gz")
pathway <- system.file("extdata", package = "ARTP2", "pathway.txt.gz")
chr <- 1:22
nchr <- length(chr)
fam <- vector("character", nchr)
bim <- vector("character", nchr)
bed <- vector("character", nchr)
for(i in 1:nchr){
  fam[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".fam", sep = ""))
  bim[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".bim", sep = ""))
```

```
  bed[i] <- system.file("extdata", package = "ARTP2", paste("chr", chr[i], ".bed", sep = ""))
}
reference <- data.frame(fam, bim, bed)
options <- list(inspect.snp.n = 2, nperm = 1e4,
                maf = .01, HWE.p = 1e-6,
                gene.R2 = .9,
                id.str = "unique-pathway-id",
                out.dir = getwd(), save.setup = FALSE)

## different inflation factors are adjusted in two studies
## first round adjustment
lambda <- c(1.10, 1.08)
ncases <- list()
ncontrols <- list()
ncases[[1]] <- c(9580, 2591)
ncontrols[[1]] <- c(53810, 3052)
ncases[[2]] <- 7638
ncontrols[[2]] <- 54319

family <- 'binomial'

## do not run permutation
options$only.setup <- TRUE
## the first round study-specific inflation is adjusted as lambda = c(1.10, 1.08)
# setup <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda,
#                ncases, ncontrols, options = options)

## the two rounds of inflation is adjusted as lambda2 = c(1.17370, 1.15236)
lambda2 <- lambda * 1.067
## run permutation to calculate p-value
options$only.setup <- FALSE
# ret1 <- sARTP(summary.files = c(study1, study2), pathway, family, reference, lambda2,
#               ncases, ncontrols, options = options)

## or adjust the second round of inflation in warm.start
# ret2 <- warm.start(setup, lambda = 1.067)

# two ways of inflation adjustment should give same results
# ret1$pathway.pvalue == ret2$pathway.pvalue

#################################################################
#################################################################
## modify or specify the method
# setup$options$method <- 2
# setup$options$inspect.snp.n <- 3

## nthread = 2 for Linux only
## nthread will be reset to 1 under Windows and Mac OS
# ret3 <- warm.start(setup, nperm = 1e5, nthread = 2)
```

Index