Package 'sumFREGAT'

March 24, 2022

Title Fast Region-Based Association Tests on Summary Statistics

Version 1.2.4

Author Nadezhda M. Belonogova <belon@bionet.nsc.ru> and
Gulnara R. Svishcheva <gulsvi@bionet.nsc.ru>
with contributions from:
Seunggeun Lee (kernel functions),
Pierre Lafaye de Micheaux ('davies' method),
Thomas Lumley ('kuonen' method),
Minghui Wang, Yiyuan Liu, Shizhong Han (simpleM function),
Yaowu Liu (ACAT function),
James O. Ramsay (functional data analysis functions),
Xihao Li, Zilin Li, and Han Chen (conception of the STAAR procedure)

Maintainer Nadezhda M. Belonogova <belon@bionet.nsc.ru>

Depends R (>= 3.0.0)

Imports methods, Matrix, splines, seqminer, GBJ

Suggests data.table

Repository CRAN

Description An adaptation of classical region/gene-based association analysis techniques to the use of summary statistics (P values and effect sizes) and correlations between genetic variants as input. It is a tool to perform the most popular and efficient gene-based tests using the results of genome-wide association (meta-)analyses without having the original genotypes and phenotypes at hand.

License GPL-3

LazyLoad yes

NeedsCompilation yes

Date/Publication 2022-03-24 10:10:02 UTC

R topics documented:

`sumFREGAT-package` *sumFREGAT: Fast REGional Association Tests on summary statistics*

Description

The sumFREGAT package computes the most popular and efficient tests for the region-based association analysis using summary statistics data (beta and P values) and correlations between genetic variants. It does not require genotype or phenotype data. Methods implemented are SKAT and SKAT-O (sequence kernel association tests), BT (burden test), ACAT (aggregated Cauchy association test), FLM (functional linear model), PCA (principal components analysis), and others.

Details

Package: sumFREGAT
Type: Package
License: GPLv3

Author(s)

Nadezhda Belonogova <belon@bionet.nsc.ru>
Gulnara Svishcheva <gulsvi@bionet.nsc.ru>

The authors are grateful to Dr. Anatoly Kirichenko for technical support
and Prof. Tatiana Axenovich for scientific guidance.

`ACATO` *omnibus aggregated Cauchy association test*

Description

This test combines P values via aggregated Cauchy association test (Liu, Y. et al., 2019)

Usage

```
ACATO(p)
```

Arguments

`p`	a vector of P values to combine

Details

ACATO is an omnibus aggregated Cauchy association test recently proposed by Liu, Y. et al. (2019). This function can be used to combine P values obtained by different gene-based tests for the same gene (see Examples below).

Value

a single P value which is a combination of the P values given in input

References

Liu Y. et al. (2019) ACAT: a fast and powerful p value combination method for rare-variant analysis in sequencing studies. Am. J. Hum. Genet. 104, 410-421.

Examples

```
cor.path <- system.file("testfiles/", package = "sumFREGAT")
score.file <- system.file("testfiles/CFH.full.vcf.gz", package = "sumFREGAT")

p.bt <- BT(score.file, genes = "CFH", cor.path = cor.path)$pvalue
p.skat <- SKAT(score.file, genes = "CFH", cor.path = cor.path)$pvalue
p.pca <- PCA(score.file, genes = "CFH", cor.path = cor.path, n = 85)$pvalue

p.combined <- ACATO(c(p.bt, p.skat, p.pca))
```

---

gene-based test functions

Gene-based tests on summary statistics

Description

A set of tests for gene-based association analysis on GWAS summary statistics: sequence kernel association tests ('SKAT', 'SKATO'), sum of chi-squares ('sumchi'), aggregated Cauchy association test ('ACAT'), burden test ('BT'), principal component analysis-based test ('PCA'), functional linear model-based test ('FLM'), Bonferroni correction test ('simpleM'), minimum P value test ('minp').

Usage

```
SKAT(score.file, gene.file, genes = "all", cor.path = "cor/",
approximation = TRUE, anno.type = "", beta.par = c(1, 25),
weights.function = NULL, user.weights = FALSE, gen.var.weights = "se.beta",
method = "kuonen", acc = 1e-8, lim = 1e+6, rho = FALSE, p.threshold = 0.8,
write.file = FALSE, quiet = FALSE)

SKATO(score.file, gene.file, genes = "all", cor.path = "cor/",
```

```
approximation = TRUE, anno.type = "", beta.par = c(1, 25),
weights.function = NULL, user.weights = FALSE, method = "kuonen",
acc = 1e-8, lim = 1e+6, rho = TRUE, p.threshold = 0.8, write.file = FALSE,
quiet = FALSE)

sumchi(score.file, gene.file, genes = "all", cor.path = "cor/",
approximation = TRUE, anno.type = "", method = "kuonen", acc = 1e-8,
lim = 1e+6, write.file = FALSE, quiet = FALSE)

ACAT(score.file, gene.file, genes = "all", anno.type = "",
beta.par = c(1, 1), weights.function = NULL, user.weights = FALSE,
gen.var.weights = "none", mac.threshold = NA, n = NA,
write.file = FALSE, quiet = FALSE)

BT(score.file, gene.file, genes = "all", cor.path = "cor/",
anno.type = "", beta.par = c(1, 25), weights.function = NULL,
user.weights = FALSE, write.file = FALSE, quiet = FALSE)

PCA(score.file, gene.file, genes = "all", cor.path = "cor/",
approximation = TRUE, anno.type = "", n, beta.par = c(1, 1),
weights.function = NULL, user.weights = FALSE, reference.matrix = TRUE,
fun = "LH", var.fraction = 0.85, write.file = FALSE, quiet = FALSE)

FLM(score.file, gene.file, genes = "all", cor.path = "cor/",
approximation = TRUE, anno.type = "", n, beta.par = c(1, 1),
weights.function = NULL, user.weights = FALSE, basis.function = "fourier",
k = 25, order = 4, flip.genotypes = FALSE, Fan = TRUE,
reference.matrix = TRUE, fun = "LH", write.file = FALSE, quiet = FALSE)

simpleM(score.file, gene.file, genes = "all", cor.path = "cor/",
anno.type = "", var.fraction = .995, write.file = FALSE, quiet = FALSE)

minp(score.file, gene.file, genes = "all", cor.path = "cor/",
anno.type = "", write.file = FALSE, quiet = FALSE)
```

Arguments

score.file	name of data file generated by prep.score.files().
gene.file	can be a name of a custom text file listing genes in refFlat format. Values "hg19" and "hg38" can be set to use default gene files containing protein-coding genes with start and stop positions from corresponding build. Mind that the same build should be used in score.file and gene.file.
genes	character vector of gene names to be analyzed. Can be "chr1", "chr2" etc. to analyze all genes on a corresponding chromosome. If not set, function will attempt to analyze all genes listed in gene.file.
cor.path	path to a folder with correlation matrix files (one file per each gene to be analyzed). Correlation matrices in text format are allowed, though ".RData" is preferable as computationally efficient. Each file should contain a square matrix

with correlation coefficients (r) between genetic variants of a gene. An example of correlation file format:
"snpname1" "snpname2" "snpname3" ...
"snpname1" 1 0.018 -0.003 ...
"snpname2" 0.018 1 0.081 ...
"snpname3" -0.003 0.081 1 ...
...
If genotypes are available, matrices can be generated as follows:
`cor.matrix <-cor(g)`
`save(cor.matrix,file = paste0(geneName,".RData"))`
where g is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (coding should be exactly the same that was used to generate GWAS statistics).
If genotypes are not available, correlations can be approximated through reference samples. Reference matrices from 1KG European sample can be downloaded at http://mga.bionet.nsc.ru/sumFREGAT/.
Names of correlation files should be constructed as "geneName.RData" (e.g. "ABCG1.RData", "ADAMTS1.RData", etc.) for ".RData" format or "geneName.txt" for text format.
Example corfiles can be found as:
system.file("testfiles/CFH.cor", package = "sumFREGAT")
system.file("testfiles/CFH.RData", package = "sumFREGAT")

`approximation` a logical value indicating whether approximation for large genes (>= 500 SNPs) should be used. Applicable for SKAT, SKATO, sumchi, PCA, FLM (default = TRUE for these methods).

`anno.type` given (functional) annotations provided by user (see) `prep.score.files()`, a character (or character vector) indicating annotation types to be used.

`n` size of the sample on which summary statistics were obtained.

`beta.par` two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of minor allele frequency (MAF) in the default weights function (see Details).

`weights.function`
a function of MAF to assign weights for each genetic variant. By default, the weights will be calculated using the beta distribution (see Details).

`user.weights` a logical value indicating whether weights from the input file should be applied. Default = FALSE.

`gen.var.weights`
a character indicating whether scores should be weighted by the variance of genotypes: "none" - no weights applied, resulting in a sum chi-square test. "se.beta" - scores weighted by variance of genotypes estimated from P values and effect sizes (regression coefficients) if provided by user. "af" - scores weighted by variance of genotypes calculated as AF * (1 - AF), where AF is allele frequency. In `SKAT()`, "se.beta" weighting results in a test analogous to a standard SKAT, while "af" is a way to approximate the standard test in case when betas are not available. Unweighted sum chi-square test gives more weights to

	common variants compared with SKAT-like tests. In `ACAT()`, the simplest default unweighted test combines P values (Z scores) without any additional info. Scores can be weighted by via gen.var.weights = 'se.beta' or 'af', similarly to SKAT.
`mac.threshold`	an integer number. In ACAT, scores with MAC <= 10 will be combined using Burden test. MACs are calculated from MAFs, `n` must be set to apply `mac.threshold`.
`method`	the method for computing P value in kernel-based tests. Available methods are "kuonen", "davies" and "hybrid" (see Details). Default = "kuonen".
`acc`	accuracy parameter for "davies" method.
`lim`	limit parameter for "davies" method.
`rho`	if TRUE the optimal test (SKAT-O) is performed [Lee et al., 2012]. `rho` can be a vector of grid values from 0 to 1. The default grid is c(0, 0.1^2, 0.2^2, 0.3^2, 0.4^2, 0.5^2, 0.5, 1).
`p.threshold`	the largest P value that will be considered as important when performing computational optimization in SKAT-O. All P values larger than p.threshold will be processed via burden test.
`basis.function`	a basis function type for beta-smooth. Can be set to "bspline" (B-spline basis) or "fourier" (Fourier basis, default).
`k`	the number of basis functions to be used for beta-smooth (default = 25).
`order`	a polynomial order to be used in "bspline". Default = 4 corresponds to the cubic B-splines. as no effect if only Fourier bases are used.
`flip.genotypes`	a logical value indicating whether the genotypes of some genetic variants should be flipped (relabelled) for their better functional representation [Vsevolozhskaya et al., 2014]. Default = FALSE.
`Fan`	if TRUE (default) then linearly dependent genetic variants will be omitted, as it was done in the original realization of FLM test by Fan et al. (2013).
`reference.matrix`	logical indicating whether the correlation matrices were generated using reference matrix. If TRUE, regularization algorithms will be applied in order to ensure invertibility and numerical stability of the matrices. Use 'reference.matrix = FALSE' ONLY IF you are sure that correlation matrices were generated using the same genotype data as for GWAS summary statistics in input.
`fun`	one of two regularization algorithms, 'LH' (default) or 'derivLH'. Currently both give similar results.
`var.fraction`	minimal proportion of genetic variance within region that should be explained by principal components used (see Details for more info).
`write.file`	output file name. If specified, output (as it proceeds) will be written to the file.
`quiet`	quiet = TRUE suppresses excessive output from reading vcf file genewise.

Details

'SKAT' uses the linear weighted kernel function to set the inter-individual similarity matrix $K = GWWG^T$ for SKAT and $K = GW(I\rho + (1-\rho)ee^T)WG^T$ for SKAT-O, where G is the $n \times p$

genotype matrix for n individuals and p genetic variants in the region, W is the $p \times p$ diagonal weight matrix, I is the $p \times p$ identity matrix, ρ is pairwise correlation coefficient between genetic effects (which will be adaptively selected from given `rho`), and e is the $p \times 1$ vector of ones. Given the shape parameters of the beta function, `beta.par = c(a,b)`, the weights are defined using probability density function of the beta distribution:

$$W_i = (B(a,b))^{-1} MAF_i^{a-1}(1-MAF_i)^{b-1},$$

where MAF_i is a minor allelic frequency for the i^{th} genetic variant in the region, which is estimated from genotypes, and $B(a,b)$ is the beta function. This way of defining weights is the same as in original SKAT (see [Wu et al., 2011] for details). `beta.par = c(1,1)` corresponds to the unweighted SKAT. The same weighting principle can be applied in 'BT' (burden test, default weights `beta.par = c(1,25)`), as well as in 'PCA' and 'FLM' tests (unweighted by default).

Depending on the method option chosen, either Kuonen or Davies method is used to calculate P values from the score statistics in SKAT. Both an Applied Statistics algorithm that inverts the characteristic function of the mixture chisq [Davies, 1980] and a saddlepoint approximation [Kuonen, 1999] are nearly exact, with the latter usually being a bit faster.

A hybrid approach has been proposed by Wu et al. [2016]. It uses the Davies' method with high accuracy, and then switches to the saddlepoint approximation method when the Davies' method fails to converge. This approach yields more accurate results in terms of type I errors, especially for small significance levels. However, 'hybrid' method runs several times slower than the saddlepoint approximation method itself (i.e. 'kuonen' method). We therefore recommend using the hybrid approach only for those regions that show significant (or nearly significant) P values to ensure their accuracy.

'ACAT' is an adaptation of a set-based aggregated Cauchy association test (ACAT-V). It has been recently proposed by Liu, Y. et al. (2019). This in the only gene-based test statistic that can be calculated from P values (Z scores) only, and does not require SNP-SNP correlation info.

Burden test ('BT', collapsing technique) suggests that the effects of causal genetic variants within a region have the same direction and the majority of variants are causal. If this is not the case, other regional tests (SKAT and FLM) are shown to have higher power compared to burden test [Svishcheva et al., 2015]. By default, 'BT' assigns weights calculated using the beta distribution with shape parameters `beta.par = c(1,25)`.

'PCA' test is based on the spectral decomposition of correlation matrix among genetic variants. The number of top principal components will be chosen in such a way that >= `var.fraction` of region variance can be explained by these PCs. By default, `var.fraction` = 0.85, i.e. PCs explain >= 85% of region variance.

A similar principle is used in 'simpleM' to calculate the effective number of independent tests.

'FLM' test assumes that the effects of multiple genetic variants can be described as a continuous function, which can be modelled through B-spline or Fourier basis functions. When the number of basis functions (set by k) is less than the number of variants within the region, the FLM test has an

advantage of using less degrees of freedom [Svishcheva, et al., 2015].

For genes with $m <= k$, functional linear models are equivalent to a standard multiple linear regression, and the latter is used for these cases. The ultimate model name is returned in output (the "model" column).

Value

A data frame with map info, P values, numbers of variants and filtered variants for each of analyzed genes.
In addition:
- BT() returns gene-level estimates of effect sizes (betas) and their standard errors.
- FLM() returns the names of the functional models used for each region. Names shortly describe the functional basis and the number of basis functions used. E.g., "F25" means 25 Fourier basis functions, "B15" means 15 B-spline basis functions.
- PCA() returns the number of the principal components used for each region and the proportion of genetic variance they make up.

References

Davies R.B. (1980) Algorithm AS 155: The Distribution of a Linear Combination of chi-2 Random Variables, Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 29, N 3, P. 323-333.
Kuonen D. (1999) Saddlepoint Approximations for Distributions of Quadratic Forms in Normal Variables. Biometrika, Vol. 86, No. 4, P. 929-935.
Wu M.C., et al. (2011) Rare-variant association testing for sequencing data with the sequence kernel association test. Am. J. Hum. Genet., Vol. 89, P. 82-93.
Lee S., et al. (2012) Optimal unified approach for rare variant association testing with application to small sample case-control whole-exome sequencing studies. American Journal of Human Genetics, 91, 224-237.
Liu Y. et al. (2019) ACAT: a fast and powerful p value combination method for rare-variant analysis in sequencing studies. Am. J. Hum. Genet. 104, 410-421. Svishcheva G.R., Belonogova N.M. and Axenovich T.I. (2015) Region-based association test for familial data under functional linear models. PLoS ONE 10(6): e0128999.
Wu B., et al. (2016) On efficient and accurate calculation of significance p-values for sequence kernel association testing of variant set. Ann Hum Genet, 80(2): 123-135.
Vsevolozhskaya O.A., et al. (2014) Functional Analysis of Variance for Association Studies. PLoS ONE 9(9): e105074.
Fan R, Wang Y, Mills JL, Wilson AF, Bailey-Wilson JE, et al. (2013) Functional linear models for association analysis of quantitative traits. Genet Epidemiol 37: 726-42.

Examples

```
# Using example score files "CFH.vcf.gz" and "CFH.full.vcf.gz"
# generated by prep.score.files() function (see examples for prep.score.files())

# Test available with minimal input (P values only):
```

```
score.file <- system.file("testfiles/CFH.vcf.gz",
package = "sumFREGAT")
ACAT(score.file, gene.file = "hg19", genes = "CFH")

# Tests that require P values, rsID, and SNP-SNP correlation info:

score.file <- system.file("testfiles/CFH.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")

sumchi(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)
simpleM(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)
minp(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)

# Tests available with full input including P values, rsID,
# betas, effect allele, allele frequencies (for default weighting),
# and correlation matrices:

score.file <- system.file("testfiles/CFH.full.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")

BT(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)
SKAT(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)
SKATO(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path)

# Tests that require sample size to be provided as "n" argument:

PCA(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path, n = 85)
FLM(score.file, gene.file = "hg19", genes = "CFH", cor.path = cor.path, n = 85)
```

`prep.score.files` *Prepare score files*

Description

Calculates Z scores from P values and beta input

Usage

```
prep.score.files(data, reference = "ref1KG.MAC5.EUR_AF.RData",
output.file.prefix)
```

Arguments

`data`	a file name or data.frame with two mandatory columns (case-insensitive header):
	"ID": unique names of genetic variants (rsIDs if 1000G correlation matrices

will be used)
"P": P value

Chromosome and positions are desirable:

"CHROM": chromosome
"POS": positions for the same build as in `gene.file` (see `gene-based test functions`)

Map data are required to assign genetic variants to genes. If not present in input file, the function will attempt to link them from the reference file (see `reference` below).

Additional columns that can be present in input file to be used in gene-based tests:

"EA": effect allele
"BETA": effect size (betas and genetic correlations should be calculated for the same genotype coding)
"EAF": effect allele frequency

Effect allele and size data can be proccesed with reference file only (see `reference` below).

"REF": reference allele
"ALT": alternative allele

"REF" and "ALT" columns can be used to compare alleles with those in reference file and exclude genetic variants if alleles do not match

Annotation columns:

"ANNO": functional annotations (like "intron_variant", "synonymous", "missense" etc.)
"PROB", "PROB1", "PROB2", "PROB3" etc.: probabilities of a variant to be causal, can be passed to `sumSTAAR()` (PHRED scale) or `FFGAS()`

For example:

CHROM POS ID EA P BETA EAF
1 196632134 1:196632134 T 0.80675 0.22946 0.00588
1 196632386 1:196632386 A 0.48694 0.65208 0.00588
1 196632470 1:196632470 G 0.25594 -0.19280 0.19412

Avoid rounding of betas and P values as this can affect the precision of regional tests.

The more data (columns) is present in input file, the more gene-based tests are

available to run. ACAT test is available with minimal input (rsIDs and P values). Together with correlation matrices (reference matrices calculated from 1000G data are available at http://mga.bionet.nsc.ru/sumFREGAT/) it allows to run `minp()`, `simpleM`, and `sumchi()` tests. Adding info on effect allele ("EA") and effect size ("BETA") enables essentially all sumFREGAT tests. Adding allele frequencies enables standard weighting via beta distribution (see `gene-based test functions` for details).

`reference` path to a reference file or data.frame with additional data needed to recode user `data` according to correlation matrices that will be used. Reference file for 1000G correlation matrices is available at http://mga.bionet.nsc.ru/sumFREGAT/. Reference file contains "ID" column with names of genetic variants that are used in correlation matrices as well as "REF" and "ALT" columns with alleles that were coded during calculation of correlation coefficients as 0 and 1, respectively. Effect sizes from `data` will be inverted for variants with effect alleles different from "ALT" alleles in reference data. If presented, "REF" and "ALT" columns from the input data will be used to sort out variants with alleles different from those in reference data. The reference file can also be a source of map data and allele frequencies if they are not present in `data`. "AF" column in the reference file represents the allele frequency of "ALT" allele.

`output.file.prefix`
if not set, the input file name will be used as output prefix.

Value

does not return any value, writes output files with Z scores to be used in any type of gene-based analysis in sumFREGAT (see `gene-based test functions`).

Examples

```
## Not run:

data <- system.file("testfiles/CFH.dat", package = "sumFREGAT")
prep.score.files(data, output.file.prefix = "CFH")

# requires reference file "ref1KG.MAC5.EUR_AF.RData" (can be downloaded
# at http://mga.bionet.nsc.ru/sumFREGAT/)

data <- system.file("testfiles/CFH.full.input.dat", package = "sumFREGAT")
prep.score.files(data, reference = "ref1KG.MAC5.EUR_AF.RData",
output.file.prefix = "CFH.full")

data <- system.file("testfiles/CFH.prob.dat", package = "sumFREGAT")
prep.score.files(data, reference = "ref1KG.MAC5.EUR_AF.RData",
output.file.prefix = "CFH.prob")

## End(Not run)
```

sumSTAAR *variant-Set Test for Association using Annotation infoRmation on summary statistics*

Description

STAAR procedure developed by Li et al. (2020) adapted to use on summary statistics, with extentions

Usage

```
sumSTAAR(score.file, gene.file, genes = 'all', cor.path = 'cor/',
tests = c('BT', 'SKAT', 'ACAT'), beta.par.matrix = rbind(c(1, 1), c(1, 25)),
prob.causal = 'all', phred = TRUE, n = NA, mac.threshold = NA, approximation = TRUE,
write.file = FALSE, staar.output = TRUE, quiet = FALSE)
```

Arguments

score.file	name of data file generated by prep.score.files().
gene.file	can be a name of a custom text file listing genes in refFlat format. Values "hg19" and "hg38" can be set to use default gene files containing protein-coding genes with start and stop positions from corresponding build. Mind that the same build should be used in score.file and gene.file.
genes	character vector of gene names to be analyzed. Can be "chr1", "chr2" etc. to analyze all genes on a corresponding chromosome. If not set, function will attempt to analyze all genes listed in gene.file.
cor.path	path to a folder with correlation matrix files (one file per each gene to be analyzed). Correlation matrices in text format are allowed, though ".RData" is preferable as computationally efficient. Each file should contain a square matrix with correlation coefficients (r) between genetic variants of a gene. An example of correlation file format: "snpname1" "snpname2" "snpname3" ... "snpname1" 1 0.018 -0.003 ... "snpname2" 0.018 1 0.081 ... "snpname3" -0.003 0.081 1 ... ... If genotypes are available, matrices can be generated as follows: cor.matrix <-cor(g) save(cor.matrix,file = paste0(geneName,".RData")) where g is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (coding should be exactly the same that was used to generate GWAS statistics). If genotypes are not available, correlations can be approximated through reference samples. Reference matrices from 1KG European sample can be downloaded at http://mga.bionet.nsc.ru/sumFREGAT/.

Names of correlation files should be constructed as "geneName.RData" (e.g. "ABCG1.RData", "ADAMTS1.RData", etc.) for ".RData" format or "geneName.txt" for text format.
Example corfiles can be found as:
system.file("testfiles/CFH.cor", package = "sumFREGAT")
system.file("testfiles/CFH.RData", package = "sumFREGAT")

`tests` a character vector with gene-based methods to be applied. By default, three methods are used: 'BT', 'SKAT', and 'ACAT'. Other weighted tests ('SKATO', 'PCA', 'FLM') can be included. Tests that do not imply weighting ('simpleM', 'minP', 'sumchi'), if listed, will be calculated once and combined along with other tests into the total sumSTAAR p-value. ACAT performs with `gen.var.weights = "af"` for consistency with STAAR.

`beta.par.matrix`
an (n x 2) matrix with rows corresponding to n `beta.par` values used sequentially. Default value `rbind(c(1,1),c(1,25))` means that `beta.par = c(1,1)` and `beta.par = c(1,25)` will be applied. `beta.par` are two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of minor allele frequency (MAF).

`prob.causal` a character vector to define a set of annotations to be used. Annotations should be initially passed to `prep.score.files` (see `prep.score.files()` function) as input file columns "PROB", "PROB1", "PROB2", "PROB3" etc. The same naming should be used for `prob.causal` argument. By default, all "PROB" columns will be used. Annotations are expected to be in PHRED scale, set `phred = FALSE` to use simple probabilities.

`phred` a logical value indicating whether probabilities are in PHRED scale. If `TRUE` (default for `sumSTAAR()`), PHRED-to-probability transformation will be performed for values indicated in `prob.causal`.

`n` size of the sample on which summary statistics were obtained. Should be assigned if 'PCA' or 'FLM' are included in `tests`.

`mac.threshold` an integer number. In ACAT, scores with MAC <= 10 will be combined using Burden test. MACs are calculated from MAFs, `n` must be set to apply `mac.threshold`. The original STAAR procedure performs with `mac.threshold = 10`.

`approximation` logical value indicating whether approximation should be used for SKAT, SKATO, PCA and FLM.

`write.file` output file name. If specified, output for all tests (as it proceeds) will be written to corresponding files.

`staar.output` logical value indicating whether extensive output format should be used (see Details).

`quiet` `quiet = TRUE` suppresses excessive output from reading vcf file genewise.

Details

The STAAR procedure has been recently proposed by Li et al. (2020) and described in detail therein. It calculates a set of P values using a range of gene-based tests, beta distribution weights

parameters, and weighting by each of 10 functional annotations. The P values are then combined using Cauchy method (see `ACATO()` function and Liu, Y. et al., (2019)).

Value

With `staar.output = FALSE` returns a data table with a single P value for each test (combinations of differently weighted and unweighted iterations of the same test) and a total sumSTAAR P value. ACAT-O method is used to combine P values (Liu, Y. et al., 2019).

If `staar.output = TRUE` the function returns a data frame of size (n.genes x (n.tests x n.beta.pars x (n.annotations + 1) + n.tests + 2)) containing P values for combined tests and all individual tests. The output is analogous to that of original STAAR procedure. For example, by default, the output will contain columns:

gene # gene symbol
BT.1.1.PROB0 # P value for burden test with beta.par = c(1, 1) and no annotations
BT.1.1.PROB1 # P value for burden test with beta.par = c(1, 1) and weighted by first annotation
BT.1.1.PROB2 # P value for burden test with beta.par = c(1, 1) and weighted by second annotation
...
BT.1.1.PROB10 # P value for burden test with beta.par = c(1, 1) and weighted by tenth annotation
BT.1.1.STAAR # combined P value of all burden tests with beta.par = c(1, 1)
BT.1.25.PROB0 # P value for burden test with beta.par = c(1, 25) and no annotations
BT.1.25.PROB1 # P value for burden test with beta.par = c(1, 25) and weighted by first annotation
...
BT.1.25.PROB10 # P value for burden test with beta.par = c(1, 25) and weighted by tenth annotation
BT.1.25.STAAR # combined P value of all burden tests with beta.par = c(1, 25)
SKAT.1.1.PROB0 # P value for SKAT with beta.par = c(1, 1) and no annotations
SKAT.1.1.PROB1 # P value for SKAT with beta.par = c(1, 1) and weighted by first annotation
...
SKAT.1.1.PROB10 # P value for SKAT with beta.par = c(1, 1) and weighted by tenth annotation
SKAT.1.1.STAAR # combined P value of all SKATs with beta.par = c(1, 1)
SKAT.1.25.PROB0 # P value for SKAT with beta.par = c(1, 25) and no annotations
...
SKAT.1.25.PROB10 # P value for SKAT with beta.par = c(1, 25) and weighted by tenth annotation
SKAT.1.25.STAAR # combined P value of all SKATs with beta.par = c(1, 25)
ACAT.1.1.PROB0 # P value for ACAT with beta.par = c(1, 1) and no annotations
ACAT.1.1.PROB1 # P value for ACAT with beta.par = c(1, 1) and weighted by first annotation
...
ACAT.1.1.PROB10 # P value for ACAT with beta.par = c(1, 1) and weighted by tenth annotation
ACAT.1.1.STAAR # combined P value of all ACATs with beta.par = c(1, 1)
ACAT.1.25.PROB0 # P value for ACAT with beta.par = c(1, 25) and no annotations
...
ACAT.1.25.PROB10 # P value for ACAT with beta.par = c(1, 25) and weighted by tenth annotation
ACAT.1.25.STAAR # combined P value of all ACATs with beta.par = c(1, 25)
sumSTAAR.ACAT_O # combined P value of all 'PROB0' gene-based tests (without weighting by annotations)
sumSTAAR.STAAR_O # combined P value of all gene-based tests

References

Li X., et al. (2020) Dynamic incorporation of multiple in silico functional annotations empowers rare variant association analysis of large whole-genome sequencing studies at scale. Nature Genetics. DOI: 10.1038/s41588-020-0676-4.
Liu Y. et al. (2019) ACAT: a fast and powerful p value combination method for rare-variant analysis in sequencing studies. Am. J. Hum. Genet. 104, 410-421.

Examples

```
cor.path <- system.file("testfiles/", package = "sumFREGAT")
score.file <- system.file("testfiles/CFH.prob.vcf.gz",
package = "sumFREGAT")
sumSTAAR(score.file, prob.causal = "PROB", gene.file = "hg19",
genes = "CFH", cor.path, quiet = TRUE)

## Not run:

score.file <- system.file("testfiles/CFH.prob.phred.vcf.gz",
package = "sumFREGAT")
res <- sumSTAAR(score.file, gene.file = "hg19", genes = "CFH",
cor.path, quiet = TRUE)

## End(Not run)
```

Index