Package 'sumFREGAT'

November 15, 2018

Title Fast Region-Based Association Tests on Summary Statistics

Version 1.1.0

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),
Minghui Wang, Yiyuan Liu, Shizhong Han (simpleM function)
Thomas Lumley ('kuonen' method), and James O. Ramsay (functional data analysis functions)

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

Depends R (>= 3.0.0)

Imports methods, Matrix, splines, seqminer, GBJ

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 2018-11-15 09:20:02 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), FLM (functional linear model), PCA (principal components analysis), MLR (multiple linear regression), 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.

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'), burden test ('BT'), principal component analysis-based test ('PCA'), functional linear model-based test ('FLM'), multiple linear regression ('MLR'), Bonferroni correction test ('simpleM'), minimum P value test ('minp').

Usage

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

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

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

PCA(score.file, gene.file, genes = "all", cor.path = "cor/",
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)

FLM(score.file, gene.file, genes = "all", cor.path = "cor/",
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)

MLR(score.file, gene.file, genes = "all", cor.path = "cor/",
anno.type = "", n, reference.matrix = TRUE, fun = "LH",
write.file = FALSE)

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

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

Arguments

score.file	name of data file generated by prep.score.files().
gene.file	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
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 [ref]. 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")

`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. This results in a test analogous to a standard SKAT. 'af' - scores weighted by variance of genotypes calculated as AF * (1 - AF), where AF is allele frequency. It is a way to approximate the standard SKAT test in case when betas are not available. Unweighted sum chi-square test gives more weights to common variants compared with SKAT-like tests.

`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.

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 was recently 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.

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. If `var.fraction` = 1 then the results of PCA test and MLR-based test are identical.

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. "MLR" means that standard multiple linear regression was applied.
- 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.
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.scores.vcf.gz" and "CFH.scores.full.vcf.gz"
# generated by prep.score.files() function (see examples for prep.score.files())

# Tests available with minimal input (P values and rsID):

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

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

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

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

sumchi(score.file, genes = "CFH", cor.path = cor.path)
simpleM(score.file, genes = "CFH", cor.path = cor.path)
minp(score.file, genes = "CFH", cor.path = cor.path)
BT(score.file, genes = "CFH", cor.path = cor.path)
SKAT(score.file, genes = "CFH", cor.path = cor.path)
SKATO(score.file, genes = "CFH", cor.path = cor.path)

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

```
PCA(score.file, genes = "CFH", cor.path = cor.path, n = 85)
FLM(score.file, genes = "CFH", cor.path = cor.path, n = 85)
MLR(score.file, 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(input.file, reference.file = "", output.file.prefix)
```

Arguments

`input.file` a file with two mandatory columns (case-insensitive header):

"ID": names of genetic variants (we suggest to provide rsIDs when possible)
"P": P value
Additional columns that can be present in input file: "CHROM": chromosome
"POS": positions for the same build as in `gene.file` (see `gene-based test functions`) and `reference.file` (37.3 with default files)
"EA": effect allele
"BETA": effect size (betas and genetic correlations should be calculated for the same genotype coding)
"EAF": effect allele frequency
"ANNO": functional annotations

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. 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/) allow 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.file` path to a reference file with additional data. Reference file from 1000G is available at http://mga.bionet.nsc.ru/sumFREGAT/.

`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

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

## Not run:

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

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

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


## End(Not run)
```

Index