Package 'sumFREGAT'

January 17, 2018

Title Fast Region-Based Association Tests on Summary Statistics

Version 1.0.0

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),
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

Suggests seqminer

Repository CRAN

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

License GPL-3

LazyLoad yes

R topics documented:

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

Description

The sumFREGAT package computes the most common and efficient tests for the region-based association analysis on 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), and MLR (multiple linear regression).

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.

BT *Family Burden Test*

Description

Burden test on summary statistics

Usage

```
BT(scoreFile, geneFile, regions, cor.path = "", annoType = "",
beta.par = c(1, 25), weights.function = ifelse(maf > 0,
dbeta(maf, beta.par[1], beta.par[2]), 0), write.file = FALSE)
```

Arguments

scoreFile	name of data file generated by `prep.score.files()`.
geneFile	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
regions	character vector of gene names to be analysed. If not set, function will attempt to analyse all genes listed in `geneFile`.

`cor.path` path to a folder with correlation files (one file per each gene to be analysed). Names of correlation files should be constructed as "geneName.cor" (e.g. "ABCG1.cor", "ADAMTS1.cor", etc.) 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 ...
...
One way to generate such file from original genotypes is:
`write.table(cor(g), file = paste0(geneName, ".cor"))`
where `g` is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (exactly the same coding that was used to generate betas).

`annoType` for files annotated with the `seqminer` package, a character (or character vector) indicating annotation types to be used (e.g. "Nonsynonymous", "Start_Loss", "Stop_loss", "Essential_Splice_Site")

`beta.par` two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of MAF in the default weights function (see Details). Default = c(1, 25).

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

`write.file` output file name. If specified, output (as it proceeds) will be written to the file.

Details

Burden test (collapsing technique) suggests that the effects of causal genetic variants within a region have the same direction. 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. 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 region, which is estimated from genotypes, and $B(a,b)$ is the beta function.

`beta.par = c(1, 1)` corresponds to the unweighted burden test.

Value

A data frame containing P values, estimates of betas and their s.e., numbers of variants and filtered variants for each of analyzed regions.

References

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.

Examples

```
## Run BT with example files:
VCFfileName <- system.file("testfiles/CFH.scores.anno.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")
out <- BT(VCFfileName, region = 'CFH', cor.path = cor.path)
```

FLM *Functional Linear Model*

Description

A region-based association test on summary statistics under functional linear models (functional data analysis approach)

Usage

```
FLM(scoreFile, geneFile, regions, cor.path = "", annoType = "",
n, beta.par = c(1, 1), weights.function = ifelse(maf > 0,
dbeta(maf, beta.par[1], beta.par[2]), 0), GVF = FALSE,
BSF = "fourier", kg = 30, kb = 25, order = 4, flip.genotypes = FALSE,
Fan = TRUE, write.file = FALSE)
```

Arguments

`scoreFile`	name of data file generated by `prep.score.files()`.
`geneFile`	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
`regions`	character vector of gene names to be analysed. If not set, function will attempt to analyse all genes listed in `geneFile`.
`cor.path`	path to a folder with correlation files (one file per each gene to be analysed). Names of correlation files should be constructed as "geneName.cor" (e.g. "ABCG1.cor", "ADAMTS1.cor", etc.) 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 ... ... One way to generate such file from original genotypes is: `write.table(cor(g), file = paste0(geneName, ".cor"))` where `g` is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (exactly the same coding that was used to generate betas).
`annoType`	for files annotated with the `seqminer` package, a character (or character vector) indicating annotation types to be used (e.g. "Nonsynonymous", "Start_Loss", "Stop_loss", "Essential_Splice_Site")
`n`	size of the sample on which summary statitics were obtained.

`beta.par` two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of MAF in the default weights function (see Details). Default = c(1, 1) corresponds to standard unweighted FLM.

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

`GVF` a basis function type for Genetic Variant Functions. Can be set to "bspline" (B-spline basis) or "fourier" (Fourier basis). The default `GVF = FALSE` assumes beta-smooth only. If `GVF = TRUE` the B-spline basis will be used.

`BSF` a basis function type for beta-smooth. Can be set to "bspline" (B-spline basis) or "fourier" (Fourier basis, default).

`kg` the number of basis functions to be used for `GVF` (default = 30, has no effect under `GVF = FALSE`).

`kb` the number of basis functions to be used for `BSF` (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 (relabeled) 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).

`write.file` output file name. If specified, output (as it proceeds) will be written to the file.

Details

The test assumes that the effects of multiple genetic variants (and also their genotypes if GVFs are used) 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 Kg and Kb) is less than the number of variants within the region, the famFLM test may have an advantage of using less degrees of freedom [Svishcheva, et al., 2015].

Several restrictions exist in combining B-spline or Fourier bases for construction of GVFs and BSF [Svishcheva, et al., 2015], and the famFLM function takes them into account. Namely:

1) $m \geq Kg \geq Kb$, where m is the number of polymorphic genetic variants within a region.

2) Under $Kg = Kb$, B-B and B-F models are equivalent to 0-B model, and F-F and F-B models are equivalent to 0-F model. 0-B and 0-F models will be used for these cases, respectively.

3) Under $m = Kb$, 0-B and 0-F models are equivalent to a standard multiple linear regression, and it will be used for these cases.

4) When Fourier basis is used, the number of basis functions should be an odd integer. Even values will be changed accordingly.

Because of these restrictions, the model in effect may not always be the same as it has been set. The ultimate model name is returned in results in the "model" column (see below).

`beta.par = c(a, b)` can be used to set weights for genetic variants. 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).

Value

A data frame containing P values, numbers of variants and filtered variants for each of analyzed regions. It also contains the names of the functional models used for each region (it may not always coincide with what was set, because of restrictions described in Details section). The first part of the name relates to the functional basis of GVFs and the second one to that of BSF, e.g. "F30-B25" means that 30 Fourier basis functions were used for construction of GVFs and 25 B-spline basis functions were used for construction of BSF. "0-F25" means that genotypes were not smoothed and 25 Fourier basis functions were used for beta-smooth. "MLR" means that standard multiple linear regression was applied.

References

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.
Vsevolozhskaya O.A., et al. (2014) Functional Analysis of Variance for Association Studies. PLoS ONE 9(9): e105074.
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.
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

```
## Run FLM with example files:
VCFfileName <- system.file("testfiles/CFH.scores.anno.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")
n <- 85 # your sample size
out <- FLM(VCFfileName, region = 'CFH', cor.path = cor.path, n = n)
```

`MLR` *Multiple Linear Regression*

Description

Multiple linear regression approach on summary statistics

Usage

```
MLR(scoreFile, geneFile, regions, cor.path = "", annoType = "",
n, write.file = FALSE)
```

Arguments

scoreFile	name of data file generated by `prep.score.files()`.
geneFile	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
regions	character vector of gene names to be analysed. If not set, function will attempt to analyse all genes listed in `geneFile`.
cor.path	path to a folder with correlation files (one file per each gene to be analysed). Names of correlation files should be constructed as "geneName.cor" (e.g. "ABCG1.cor", "ADAMTS1.cor", etc.) 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 ... ... One way to generate such file from original genotypes is: `write.table(cor(g), file = paste0(geneName, ".cor"))` where `g` is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (exactly the same coding that was used to generate betas).
annoType	for files annotated with the `seqminer` package, a character (or character vector) indicating annotation types to be used (e.g. "Nonsynonymous", "Start_Loss", "Stop_loss", "Essential_Splice_Site")
n	size of the sample on which summary statitics were obtained.
write.file	output file name to write results as they come (sequential mode only).
...	other arguments that could be passed to `null()`, `read.plink()` and `readVCFToMatrixByGene()`.

Value

A data frame containing P values, numbers of variants and filtered variants for each of analyzed regions.

Examples

```
## Run MLR with example files:
VCFfileName <- system.file("testfiles/CFH.scores.anno.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")
n <- 85 # your sample size
out <- MLR(VCFfileName, region = 'CFH', cor.path = cor.path, n = n)
```

PCA *Principal Components Analysis*

Description

Test for association between a trait and principal components of genotypes within a region, on summary statistics

Usage

```
PCA(scoreFile, geneFile, regions, cor.path = "", annoType = "",
n, beta.par = c(1, 1), weights.function = ifelse(maf > 0,
dbeta(maf, beta.par[1], beta.par[2]), 0), var.fraction = .85,
write.file = FALSE)
```

Arguments

`scoreFile`	name of data file generated by `prep.score.files()`.
`geneFile`	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
`regions`	character vector of gene names to be analysed. If not set, function will attempt to analyse all genes listed in `geneFile`.
`cor.path`	path to a folder with correlation files (one file per each gene to be analysed). Names of correlation files should be constructed as "geneName.cor" (e.g. "ABCG1.cor", "ADAMTS1.cor", etc.) 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 ... ... One way to generate such file from original genotypes is: `write.table(cor(g), file = paste0(geneName, ".cor"))` where `g` is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (exactly the same coding that was used to generate betas).
`annoType`	for files annotated with the `seqminer` package, a character (or character vector) indicating annotation types to be used (e.g. "Nonsynonymous", "Start_Loss", "Stop_loss", "Essential_Splice_Site")
`n`	size of the sample on which summary statitics were obtained.
`beta.par`	two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of MAF in the default weights function (see Details). Default = c(1, 1).
`weights.function`	a function of minor allele frequency (MAF) to assign weights for each genetic variant. By default, the weights will be calculated using the beta distribution (see Details).
`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

PCA test is a useful tool to detect association between genetic variants of a region and a trait when genetic variants are strongly correlated. 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. >= 85% of region variance will be explained by PCs. If `var.fraction` = 1 then the results of PCA test and MLR-based test are identical.

`beta.par = c(a, b)` can be used to set weights for genetic variants. 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).
Precision of input values (betas and P) can be important for perfect correspondence between PCA on summary statistics and PCA performed on genotypes. We suggest to keep as much decimal points as possible for input values.

Value

a data frame containing P values, numbers of variants and filtered variants for each of analyzed regions. It also contains the number of the principal components used for each region and the proportion of genetic variance they make up.

References

Jolliffe, I.T. A note on the use of principal components in regression. J R Stat Soc Ser C 31, 300-303 (1982).

Examples

```
## Run PCA with example files:
VCFfileName <- system.file("testfiles/CFH.scores.anno.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")
n <- 85 # your sample size
out <- PCA(VCFfileName, region = 'CFH', cor.path = cor.path, n = n)
```

---

prep.score.files *Prepare score files*

Description

Calculates Z scores from P values and beta input

Usage

```
prep.score.files(input.file, output.file.prefix)
```

Arguments

`input.file` a file with the following columns:

"CHROM": chromosome
"POS": positions
"ID": names of genetic variants, same as in files with genetic correlations
"REF": reference allele
"ALT": alternative allele
"P": p value
"BETA": effect size (betas and genetic correlations should be calculated for the same genotype coding)
"EAF": effect allele freequency

For example:

CHROM POS ID REF ALT pvalue beta EAF
1 196632134 1:196632134 C T 0.80675 0.22946 0.00588
1 196632386 1:196632386 G A 0.48694 0.65208 0.00588
1 196632470 1:196632470 A G 0.25594 -0.19280 0.19412
Avoid rounding of betas and pvalues as this can affect the precision of regional tests.

`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: `SKAT()`, `BT()`, `MLR()`, `FLM()`, `PCA()`.

Examples

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

SKAT *Sequence kernel association test*

Description

Sequence kernel association tests (SKAT and SKAT-O) on summary statistics

Usage

```
SKAT(scoreFile, geneFile, regions, cor.path = "", annoType = "",
beta.par = c(1, 25), weights.function = ifelse(maf > 0, dbeta(maf,
beta.par[1], beta.par[2]), 0), method = "kuonen", acc = 1e-8,
lim = 1e+6, rho = FALSE, write.file = FALSE)
```

Arguments

`scoreFile`	name of data file generated by `prep.score.files()`.
`geneFile`	name of a text file listing genes in refFlat format. If not set, hg19 file will be used (see Examples below).
`regions`	character vector of gene names to be analysed. If not set, function will attempt to analyse all genes listed in `geneFile`.
`cor.path`	path to a folder with correlation files (one file per each gene to be analysed). Names of correlation files should be constructed as "geneName.cor" (e.g. "ABCG1.cor", "ADAMTS1.cor", etc.) 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 ... ... One way to generate such file from original genotypes is: `write.table(cor(g), file = paste0(geneName, ".cor"))` where `g` is a genotype matrix (nsample x nvariants) for a given gene with genotypes coded as 0, 1, 2 (exactly the same coding that was used to generate betas).
`annoType`	for files annotated with the `seqminer` package, a character (or character vector) indicating annotation types to be used (e.g. "Nonsynonymous", "Start_Loss", "Stop_loss", "Essential_Splice_Site")
`beta.par`	two positive numeric shape parameters in the beta distribution to assign weights for each genetic variant as a function of MAF in the default weights function (see Details). Default = c(1, 25).
`weights.function`	a function of minor allele frequency (MAF) to assign weights for each genetic variant. By default, the weights will be calculated using the beta distribution (see Details).
`method`	the method for computing P value. 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).
`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. Depending on the method option chosen, either Kuonen or Davies method is used to calculate P values from the score statistics. 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.

Value

A list with values:

`results`	a data frame containing P values, numbers of variants and polymorphic variants for each of analyzed regions. If `return.variance.explained = TRUE` it contains also the column with marginal amounts of variance explained by each region. If `reml = FALSE` the new estimates of heritability (h2) and total variance with corresponding total log-likelihood are also returned.
`nullmod`	an object containing the estimates of the null model parameters: heritability (h2), total variance (total.var), estimates of fixed effects of covariates (alpha), the gradient (df), and the total log-likelihood (logLH).
`sample.size`	the sample size after omitting NAs.
`time`	If `return.time = TRUE` a list with running times for null model, regional analysis and total analysis is returned. See `proc.time()` for output format.

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.

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.

Examples

```
## Run SKAT with example files:
VCFfileName <- system.file("testfiles/CFH.scores.anno.vcf.gz",
package = "sumFREGAT")
cor.path <- system.file("testfiles/", package = "sumFREGAT")
out <- SKAT(VCFfileName, region = 'CFH', cor.path = cor.path)
```

Index