Package ‘gwasurvivr’

April 22, 2024

Type Package

Title gwasurvivr: an R package for genome wide survival analysis

Version 1.21.0

Author Abbas Rizvi, Ezgi Karaesmen, Martin Morgan, Lara Sucheston-Campbell

Maintainer Abbas Rizvi <aarizv@gmail.com>

Description gwasurvivr is a package to perform survival analysis using Cox proportional hazard models on imputed genetic data.

VignetteBuilder knitr

Imports GWASTools, survival, VariantAnnotation, parallel, matrixStats, SummarizedExperiment, stats, utils, SNPRelate

Suggests BiocStyle, knitr, rmarkdown

Depends R (>= 3.4.0)

License Artistic-2.0

URL https://github.com/suchestoncampbelllab/gwasurvivr

biocViews GenomeWideAssociation, Survival, Regression, Genetics, SNP, GeneticVariability, Pharmacogenomics, BiomedicalInformatics

RoxygenNote 6.1.0

Encoding UTF-8

git_url https://git.bioconductor.org/packages/gwasurvivr

git_branch devel

git_last_commit 81de63f

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-04-22

Contents

gdsCoxSurv 2
impute2CoxSurv 4
michiganCoxSurv 6
plinkCoxSurv 8
sangerCoxSurv 10

Index **13**

`gdsCoxSurv` *Fit Cox survival to all variants from a standard IMPUTE2 output after genotype imputation*

Description

Performs survival analysis using Cox proportional hazard models on imputed genetic data from GDS files.

Usage

```
gdsCoxSurv(gdsfile, covariate.file, id.column, sample.ids = NULL,
  time.to.event, event, covariates, inter.term = NULL,
  print.covs = "only", out.file, chunk.size = 5000,
  maf.filter = 0.05, flip.dosage = TRUE, verbose = TRUE,
  clusterObj = NULL)
```

Arguments

`gdsfile`	path to .gds file. Location of the .gds file should also contain .snp.rdata and .scan.rdata files.
`covariate.file`	data.frame comprising phenotype information, all covariates to be added in the model must be numeric.
`id.column`	character giving the name of the ID column in covariate.file.
`sample.ids`	character vector of sample IDs to keep in survival analysis
`time.to.event`	character of column name in covariate.file that represents the time interval of interest in the analysis
`event`	character of column name in covariate.file that represents the event of interest to be included in the analysis
`covariates`	character vector with exact names of columns in covariate.file to include in analysis
`inter.term`	character string giving the column name of the covariate that will be added to the interaction term with SNP (e.g. `term*SNP`). See details.
`print.covs`	character string of either `"only"`, `"all"` or `"some"`, defining which covariate statistics should be printed to the output. See details.

`out.file`	character of output file name (do not include extension)
`chunk.size`	integer number of variants to process per thread
`maf.filter`	numeric to filter minor allele frequency (i.e. choosing 0.05 means filtering MAF>0.05). User can set this to `NULL` if no filtering is preffered. Default is 0.05.
`flip.dosage`	logical to flip which allele the dosage was calculated on, default `flip.dosage=TRUE`
`verbose`	logical for messages that describe which part of the analysis is currently being run
`clusterObj`	A cluster object that can be used with the `parApply` function. See details.

Details

Testing for SNP-covariate interactions: User can define the column name of the covariate that will be included in the interaction term. For example, for given covariates `a` and `b`, where `c` is defined as the `inter.term` the model will be: `~ a + b + c + SNP + c*SNP`.

Printing results of other covariates: `print.covs` argument controls the number of covariates will be printed as output. The function is set to `only` by default and will only print the SNP or if an interaction term is given, the results of the interaction term (e.g. `SNP*covariate`). Whereas, `all` will print results (coef, se.coef, p.value etc) of all covariates included in the model. `some` is only applicable if an interaction term is given and will print the results for SNP, covariate tested for interaction and the interaction term. User should be mindful about using the `all` option, as it will likely slow down the analysis and will increase the output file size.

User defined parallelization: This function uses `parApply` from `parallel` package to fit models to SNPs in parallel. User is not required to set any options for the parallelization. However, advanced users who wish to optimize it, can provide a cluster object generated by `makeCluster` family of functions that suits their need and platform.

Value

Saves two text files directly to disk: `.coxph` extension containing CoxPH survival analysis results. `.snps_removed` extension containing SNPs that were removed due to low variance or user-defined thresholds.

Examples

```
gdsfile <- system.file(package="gwasurvivr",
                       "extdata",
                       "gds_example.gds")
covariate.file <- system.file(package="gwasurvivr",
                              "extdata",
                              "simulated_pheno.txt")
covariate.file <- read.table(covariate.file,
                             sep=" ",
                             header=TRUE,
                             stringsAsFactors = FALSE)
covariate.file$SexFemale <- ifelse(covariate.file$sex=="female", 1L, 0L)
sample.ids <- covariate.file[covariate.file$group=="experimental",]$ID_2
gdsCoxSurv(gdsfile=gdsfile,
```

```
covariate.file=covariate.file,
id.column="ID_2",
sample.ids=sample.ids,
time.to.event="time",
event="event",
covariates=c("age", "SexFemale", "DrugTxYes"),
inter.term=NULL,
print.covs="only",
out.file="impute_example",
chunk.size=50,
maf.filter=0.005,
flip.dosage=TRUE,
verbose=TRUE,
clusterObj=NULL)
```

`impute2CoxSurv` *Fit Cox survival to all variants from a standard IMPUTE2 output after genotype imputation*

Description

Performs survival analysis using Cox proportional hazard models on imputed genetic data from IMPUTE2 output

Usage

```
impute2CoxSurv(impute.file, sample.file, chr, covariate.file, id.column,
  sample.ids = NULL, time.to.event, event, covariates,
  inter.term = NULL, print.covs = "only", out.file,
  chunk.size = 10000, maf.filter = 0.05, exclude.snps = NULL,
  flip.dosage = TRUE, verbose = TRUE, clusterObj = NULL,
  keepGDS = FALSE)
```

Arguments

`impute.file`	character of IMPUTE2 file
`sample.file`	character of sample file affiliated with IMPUTE2 file
`chr`	numeric denoting chromosome number
`covariate.file`	data.frame comprising phenotype information, all covariates to be added in the model must be numeric.
`id.column`	character giving the name of the ID column in covariate.file.
`sample.ids`	character vector of sample IDs to keep in survival analysis
`time.to.event`	character of column name in covariate.file that represents the time interval of interest in the analysis
`event`	character of column name in covariate.file that represents the event of interest to be included in the analysis

covariates	character vector with exact names of columns in covariate.file to include in analysis
inter.term	character string giving the column name of the covariate that will be added to the interaction term with SNP (e.g. term*SNP). See details.
print.covs	character string of either "only", "all" or "some", defining which covariate statistics should be printed to the output. See details.
out.file	character of output file name (do not include extension)
chunk.size	integer number of variants to process per thread
maf.filter	numeric to filter minor allele frequency (i.e. choosing 0.05 means filtering MAF>0.05). User can set this to NULL if no filtering is preffered. Default is 0.05.
exclude.snps	a character vector listing the rsIDs of SNPs that will be excluded from analyses
flip.dosage	logical to flip which allele the dosage was calculated on, default flip.dosage=TRUE
verbose	logical for messages that describe which part of the analysis is currently being run
clusterObj	A cluster object that can be used with the parApply function. See details.
keepGDS	logical to keep GDS files (compressed IMPUTE2 files) after the analysis. Defaults to FALSE.

Details

Testing for SNP-covariate interactions: User can define the column name of the covariate that will be included in the interaction term. For example, for given covariates a and b, where c is defined as the inter.term the model will be: ~ a + b + c + SNP + c*SNP.

Printing results of other covariates: print.covs argument controls the number of covariates will be printed as output. The function is set to only by default and will only print the SNP or if an interaction term is given, the results of the interaction term (e.g. SNP*covariate). Whereas, all will print results (coef, se.coef, p.value etc) of all covariates included in the model. some is only applicable if an interaction term is given and will print the results for SNP, covariate tested for interaction and the interaction term. User should be mindful about using the all option, as it will likely slow down the analysis and will increase the output file size.

User defined parallelization: This function uses parApply from parallel package to fit models to SNPs in parallel. User is not required to set any options for the parallelization. However, advanced users who wish to optimize it, can provide a cluster object generated by makeCluster family of functions that suits their need and platform.

Value

Saves two text files directly to disk: .coxph extension containing CoxPH survival analysis results. .snps_removed extension containing SNPs that were removed due to low variance or user-defined thresholds.

Examples

```
impute.file <- system.file(package="gwasurvivr",
                           "extdata",
                           "impute_example.impute2.gz")
sample.file <- system.file(package="gwasurvivr",
                           "extdata",
                           "impute_example.impute2_sample")
covariate.file <- system.file(package="gwasurvivr",
                              "extdata",
                              "simulated_pheno.txt")
covariate.file <- read.table(covariate.file,
                             sep=" ",
                             header=TRUE,
                             stringsAsFactors = FALSE)
covariate.file$SexFemale <- ifelse(covariate.file$sex=="female", 1L, 0L)
sample.ids <- covariate.file[covariate.file$group=="experimental",]$ID_2
impute2CoxSurv(impute.file=impute.file,
               sample.file=sample.file,
               chr=14,
               covariate.file=covariate.file,
               id.column="ID_2",
               sample.ids=sample.ids,
               time.to.event="time",
               event="event",
               covariates=c("age", "SexFemale", "DrugTxYes"),
               inter.term=NULL,
               print.covs="only",
               out.file="impute_example",
               chunk.size=50,
               maf.filter=0.005,
               exclude.snps=NULL,
               flip.dosage=TRUE,
               verbose=TRUE,
               clusterObj=NULL,
               keepGDS=FALSE)
```

`michiganCoxSurv` *Fit Cox survival to all variants in a .vcf.gz file from Michigan imputation server*

Description

Performs survival analysis using Cox proportional hazard models on imputed genetic data stored in compressed VCF files

Usage

```
michiganCoxSurv(vcf.file, covariate.file, id.column, sample.ids = NULL,
```

```
time.to.event, event, covariates, inter.term = NULL,
print.covs = "only", out.file, maf.filter = 0.05, r2.filter = NULL,
chunk.size = 5000, verbose = TRUE, clusterObj = NULL)
```

Arguments

`vcf.file`	character(1) path to VCF file.
`covariate.file`	matrix(1) comprising phenotype (time, event) and additional covariate data.
`id.column`	character(1) providing exact match to sample ID column from `covariate.file`
`sample.ids`	character vector with sample ids to include in analysis
`time.to.event`	character(1) string that matches time column name in pheno.file
`event`	character(1) string that matches event column name in pheno.file
`covariates`	character vector with matching column names in pheno.file of covariates of interest
`inter.term`	character(1) string giving the column name of the covariate that will be added to the interaction term with SNP (e.g. `term*SNP`). See details.
`print.covs`	character(1) string of either `"only"`, `"all"` or `"some"`, defining which covariate statistics should be printed to the output. See details.
`out.file`	character(1) string with output name
`maf.filter`	integer(1) filter out minor allele frequency below threshold (i.e. 0.005 will filter MAF > 0.005)
`r2.filter`	integer(1) of imputation quality score filter (i.e. 0.7 will filter r2 > 0.7)
`chunk.size`	integer(1) number of variants to process per thread
`verbose`	logical(1) for messages that describe which part of the analysis is currently being run
`clusterObj`	A cluster object that can be used with the `parApply` function. See details.

Details

Testing for SNP-covariate interactions: User can define the column name of the covariate that will be included in the interaction term. For example, for given covariates `a` and `b`, where `c` is defined as the `inter.term` the model will be: `~ a + b + c + SNP + c*SNP`.

Printing results of other covariates: `print.covs` argument controls the number of covariates will be printed as output. The function is set to `only` by default and will only print the SNP or if an interaction term is given, the results of the interaction term (e.g. `SNP*covariate`). Whereas, `all` will print results (coef, se.coef, p.value etc) of all covariates included in the model. `some` is only applicable if an interaction term is given and will print the results for SNP, covariate tested for interaction and the interaction term. User should be mindful about using the `all` option, as it will likely slow down the analysis and will increase the output file size.

User defined parallelization: This function uses `parApply` from `parallel` package to fit models to SNPs in parallel. User is not required to set any options for the parallelization. However, advanced users who wish to optimize it, can provide a cluster object generated by `makeCluster` family of functions that suits their need and platform.

Value

Saves two text files directly to disk: `.coxph` extension containing CoxPH survival analysis results. `.snps_removed` extension containing SNPs that were removed due to low variance or user-defined thresholds.

Examples

```
vcf.file <- system.file(package="gwasurvivr",
                        "extdata",
                      "michigan.chr14.dose.vcf.gz")
pheno.fl <- system.file(package="gwasurvivr",
                        "extdata",
                      "simulated_pheno.txt")
pheno.file <- read.table(pheno.fl,
                         sep=" ",
                         header=TRUE,
                         stringsAsFactors = FALSE)
pheno.file$SexFemale <- ifelse(pheno.file$sex=="female", 1L, 0L)
sample.ids <- pheno.file[pheno.file$group=="experimental",]$ID_2
michiganCoxSurv(vcf.file=vcf.file,
               covariate.file=pheno.file,
               id.column="ID_2",
               sample.ids=sample.ids,
               time.to.event="time",
               event="event",
               covariates=c("age", "SexFemale", "DrugTxYes"),
               inter.term=NULL,
               print.covs="only",
               out.file="michigan_example",
               r2.filter=0.3,
               maf.filter=0.005,
               chunk.size=50,
               verbose=TRUE,
               clusterObj=NULL)
```

---

plinkCoxSurv — *Fit Cox survival to all variants from PLINK binary files (.BED, .BIM, .FAM)*

Description

Performs survival analysis using Cox proportional hazard models on directly typed data in PLINK format

Usage

```
plinkCoxSurv(bed.file, covariate.file, id.column, sample.ids = NULL,
  time.to.event, event, covariates, inter.term = NULL,
```

```
  print.covs = "only", out.file, chunk.size = 10000,
  maf.filter = 0.005, flip.dosage = TRUE, verbose = TRUE,
  clusterObj = NULL)
```

Arguments

`bed.file`	character of name of plink files without extension
`covariate.file`	data.frame comprising phenotype information, all covariates to be added in the model must be numeric.
`id.column`	character giving the name of the ID column in covariate.file.
`sample.ids`	character vector of sample IDs to keep in survival analysis
`time.to.event`	character of column name in covariate.file that represents the time interval of interest in the analysis
`event`	character of column name in covariate.file that represents the event of interest to be included in the analysis
`covariates`	character vector with exact names of columns in covariate.file to include in analysis
`inter.term`	character string giving the column name of the covariate that will be added to the interaction term with SNP (e.g. `term*SNP`). See details.
`print.covs`	character string of either `"only"`, `"all"` or `"some"`, defining which covariate statistics should be printed to the output. See details.
`out.file`	character of output file name (do not include extension)
`chunk.size`	integer number of variants to process per thread
`maf.filter`	numeric to filter minor allele frequency (i.e. choosing 0.05 means filtering MAF>0.05). User can set this to `NULL` if no filtering is preffered. Default is 0.05.
`flip.dosage`	logical to flip which allele the dosage was calculated on, default `flip.dosage=TRUE`
`verbose`	logical for messages that describe which part of the analysis is currently being run
`clusterObj`	A cluster object that can be used with the `parApply` function. See details.

Details

Testing for SNP-covariate interactions: User can define the column name of the covariate that will be included in the interaction term. For example, for given covariates `a` and `b`, where `c` is defined as the `inter.term` the model will be: `~ a + b + c + SNP + c*SNP`.

Printing results of other covariates: `print.covs` argument controls the number of covariates will be printed as output. The function is set to `only` by default and will only print the SNP or if an interaction term is given, the results of the interaction term (e.g. `SNP*covariate`). Whereas, `all` will print results (coef, se.coef, p.value etc) of all covariates included in the model. `some` is only applicable if an interaction term is given and will print the results for SNP, covariate tested for interaction and the interaction term. User should be mindful about using the `all` option, as it will likely slow down the analysis and will increase the output file size.

User defined parallelization: This function uses `parApply` from `parallel` package to fit models to SNPs in parallel. User is not required to set any options for the parallelization. However, advanced users who wish to optimize it, can provide a cluster object generated by `makeCluster` family of functions that suits their need and platform.

Value

Saves two text files directly to disk: `.coxph` extension containing CoxPH survival analysis results. `.snps_removed` extension containing SNPs that were removed due to low variance or user-defined thresholds.

Examples

```
bed.file <- system.file(package="gwasurvivr",
                        "extdata",
                        "plink_example.bed")
covariate.file <- system.file(package="gwasurvivr",
                              "extdata",
                              "simulated_pheno.txt")
covariate.file <- read.table(covariate.file,
                             sep=" ",
                             header=TRUE,
                             stringsAsFactors = FALSE)
covariate.file$SexFemale <- ifelse(covariate.file$sex=="female", 1L, 0L)
sample.ids <- covariate.file[covariate.file$group=="experimental",]$ID_2
plinkCoxSurv(bed.file=bed.file,
             covariate.file=covariate.file,
             id.column="ID_2",
             sample.ids=sample.ids,
             time.to.event="time",
             event="event",
             covariates=c("age", "SexFemale", "DrugTxYes"),
             inter.term=NULL,
             print.covs="only",
             out.file="impute_example",
             chunk.size=50,
             maf.filter=0.005,
             flip.dosage=TRUE,
             verbose=TRUE,
             clusterObj=NULL)
```

`sangerCoxSurv` *Fit Cox survival to all variants in a .vcf.gz file from Sanger imputation server*

Description

Performs survival analysis using Cox proportional hazard models on imputed genetic data stored in compressed VCF files.

Usage

```
sangerCoxSurv(vcf.file, covariate.file, id.column, sample.ids = NULL,
  time.to.event, event, covariates, inter.term = NULL,
  print.covs = "only", out.file, maf.filter = 0.05,
  info.filter = NULL, chunk.size = 5000, verbose = TRUE,
  clusterObj = NULL)
```

Arguments

`vcf.file`	character(1) path to VCF file.
`covariate.file`	matrix(1) comprising phenotype (time, event) and additional covariate data.
`id.column`	character(1) providing exact match to sample ID column from `covariate.file`
`sample.ids`	character vector with sample ids to include in analysis
`time.to.event`	character(1) string that matches time column name in pheno.file
`event`	character(1) string that matches event column name in pheno.file
`covariates`	character vector with matching column names in pheno.file of covariates of interest
`inter.term`	character(1) string giving the column name of the covariate that will be added to the interaction term with SNP (e.g. `term*SNP`). See details.
`print.covs`	character(1) string of either `"only"`, `"all"` or `"some"`, defining which covariate statistics should be printed to the output. See details.
`out.file`	character(1) string with output name
`maf.filter`	integer(1) filter out minor allele frequency below threshold (i.e. 0.005 will filter MAF > 0.005)
`info.filter`	integer(1) of imputation quality score filter (i.e. 0.7 will filter info > 0.7)
`chunk.size`	integer(1) number of variants to process per thread
`verbose`	logical(1) for messages that describe which part of the analysis is currently being run
`clusterObj`	A cluster object that can be used with the `parApply` function. See details.

Details

Testing for SNP-covariate interactions: User can define the column name of the covariate that will be included in the interaction term. For example, for given covariates `a` and `b`, where `c` is defined as the `inter.term` the model will be: `~ a + b + c + SNP + c*SNP`.

Printing results of other covariates: `print.covs` argument controls the number of covariates will be printed as output. The function is set to `only` by default and will only print the SNP or if an interaction term is given, the results of the interaction term (e.g. `SNP*covariate`). Whereas, `all` will print results (coef, se.coef, p.value etc) of all covariates included in the model. `some` is only applicable if an interaction term is given and will print the results for SNP, covariate tested for interaction and the interaction term. User should be mindful about using the `all` option, as it will likely slow down the analysis and will increase the output file size.

User defined parallelization: This function uses `parApply` from `parallel` package to fit models to SNPs in parallel. User is not required to set any options for the parallelization. However, advanced users who wish to optimize it, can provide a cluster object generated by `makeCluster` family of functions that suits their need and platform.

Value

Saves two text files directly to disk: `.coxph` extension containing CoxPH survival analysis results. `.snps_removed` extension containing SNPs that were removed due to low variance or user-defined thresholds.

Examples

```
vcf.file <- system.file(package="gwasurvivr",
                        "extdata",
                        "sanger.pbwt_reference_impute.vcf.gz")
pheno.fl <- system.file(package="gwasurvivr",
                         "extdata",
                       "simulated_pheno.txt")
pheno.file <- read.table(pheno.fl,
                          sep=" ",
                          header=TRUE,
                          stringsAsFactors = FALSE)
pheno.file$SexFemale <- ifelse(pheno.file$sex=="female", 1L, 0L)
sample.ids <- pheno.file[pheno.file$group=="experimental",]$ID_2
sangerCoxSurv(vcf.file=vcf.file,
              covariate.file=pheno.file,
              id.column="ID_2",
              sample.ids=sample.ids,
              time.to.event="time",
              event="event",
              covariates=c("age", "SexFemale", "DrugTxYes"),
              inter.term=NULL,
              print.covs="only",
              out.file="sanger_example",
              info.filter=0.3,
              maf.filter=0.005,
              chunk.size=50,
              verbose=TRUE,
              clusterObj=NULL)
```

Index

gdsCoxSurv, 2

impute2CoxSurv, 4

michiganCoxSurv, 6

plinkCoxSurv, 8

sangerCoxSurv, 10