Package 'comorbidPGS'

April 30, 2024

Title Assessing Predisposition Between Phenotypes using Polygenic Scores

Version 0.3.4

Description Using polygenic scores (PGS, or PRS/GRS for binary outcomes), this package allows to investigate shared predisposition between different conditions, and do fast association analysis, export plots and views of the PGS distribution using 'ggplot2' object.

Depends R (>= 3.5.0)

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.2

Imports ggplot2, stats, utils, MASS, nnet, parallel

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Vincent Pascat [aut, cre] (<https://orcid.org/0000-0002-6227-8812>)

Maintainer Vincent Pascat <vincent.pascat@univ-lille.fr>

Repository CRAN

Date/Publication 2024-04-30 09:00:08 UTC

R topics documented:

`assoc` *Association of a PGS distribution with a Phenotype*

Description

`assoc()` take a distribution of PGS, a Phenotype and eventual Confounders return a data frame showing the association of PGS on the Phenotype

Usage

```
assoc(
  df = NULL,
  prs_col = "SCORESUM",
  phenotype_col = "Phenotype",
  scale = TRUE,
  covar_col = NA,
  verbose = TRUE,
  log = ""
)
```

Arguments

`df`	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
`prs_col`	a character specifying the PGS column name
`phenotype_col`	a character specifying the Phenotype column name
`scale`	a boolean specifying if scaling of PGS should be done before testing
`covar_col`	a character vector specifying the covariate column names (facultative)
`verbose`	a boolean (TRUE by default) to write in the console/log messages.
`log`	a connection, or a character string naming the file to print to. If "" (by default), it prints to the standard output connection, the console unless redirected by sink.

Value

return a data frame showing the association of the PGS on the Phenotype with the following columns:

- PGS: the name of the PGS
- Phenotype: the name of Phenotype

- Phenotype_type: either 'Continuous', 'Ordered Categorical', 'Categorical' or 'Cases/Controls'
- Stat_method: association function detects what is the phenotype type and what is the best way to analyse it, either 'Linear regression', 'Binary logistic regression', 'Ordinal logistic regression' or 'Multinomial logistic regression'
- Covar: list all the covariates used for this association
- N_cases: if Phenotype_type is Cases/Controls, gives the number of cases
- N_controls: if Phenotype_type is Cases/Controls, gives the number of controls
- N: the number of individuals/samples
- Effect: if Phenotype_type is Continuous, it represents the Beta coefficient of linear regression; Otherwise, it is the OR of logistic regression
- SE: standard error of the Beta coefficient (if Phenotype_type is Continuous)
- lower_CI: lower confidence interval of the related Effect (Beta or OR)
- upper_CI: upper confidence interval of the related Effect (Beta or OR)
- P_value: associated P-value

Examples

```
results <- assoc(
  df = comorbidData,
  prs_col = "ldl_PGS",
  phenotype_col = "log_ldl",
  scale = TRUE,
  covar_col = c("age", "sex", "gen_array")
)
print(results)
```

---

assocplot *Multiple PGS Associations Plot*

Description

assocplot() take a data frame of associations, return plot of the associations from assoc() (ggplot2 object or list of ggplot object)

Usage

```
assocplot(score_table = NULL, axis = "vertical", pval = FALSE)
```

Arguments

`score_table` a dataframe with association results with at least the following columns:

- PGS: the name of the PGS
- Phenotype: the name of Phenotype
- Phenotype_type: either `'Continuous'`, `'Ordered Categorical'`, `'Categorical'` or `'Cases/Controls'`
- Effect: if Phenotype_type is Continuous, it represents the Beta coefficient of linear regression, OR of logistic regression otherwise
- lower_CI: lower confidence interval of the related Effect (Beta or OR)
- upper_CI: upper confidence interval of the related Effect (Beta or OR)
- P_value: associated P-value

`axis` a character, `'horizontal'` or `"vertical"` (the default) specifying the rotation of the plot

`pval` a parameter specifying information on how to display P-value

- if pval is FALSE, P-value does not appear on the plot
- if pval is TRUE, P-value always appears next to the signal
- if pval is a number, P-value will appear if the P-value is inferior to this given number.

Value

return either:

- a ggplot object representing the association results.
- a list of two ggplot objects, accessible by $continuous_phenotype and $discrete_phenotype, if there are both Continuous Phenotypes and Discrete Phenotypes (i.e. "Categorical" or "Cases/Controls")

`centileplot` *Centiles Plot from a PGS Association*

Description

`centileplot()` take a distribution of PGS, a Phenotype and eventual Confounders return a plot (ggplot2 object) with centiles (or deciles if not enough individuals) of PGS in x and Prevalence/Median/Mean of the Phenotype in y

Usage

```
centileplot(
  df = NULL,
  prs_col = "SCORESUM",
  phenotype_col = "Phenotype",
  decile = FALSE,
  continuous_metric = NA
)
```

Arguments

`df`	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
`prs_col`	a character specifying the PGS column name
`phenotype_col`	a character specifying the Phenotype column name
`decile`	a boolean specifying if centiles or deciles should be used
`continuous_metric`	a facultative character specifying what metric to use for continuous Phenotype, only three options: `NA`, `"median"` or `"mean"`

Value

return a figure of results in the format ggplot2 object

`comorbidData` *Mock dataset for comorbidPGS package*

Description

A dataset with sets of PGSs, Phenotypes and Covariates to demo the comorbidPGS package

Usage

```
comorbidData
```

Format

`who`**:**

A data frame with 10,000 rows (individuals) and 16 columns:

ID Individual's identifier, characters

sex Sex of the individuals, binary numeric values

age Age of the individuals, numeric value

gen_array The genotypic array used for those individuals, factor values

ethnicity The ethnicity of individuals, can be also used as Categorical Phenotype, factor values

brc_PGS, t2d_PGS, ldl_PGS Three distributions of PGS for Breast Cancer, Type 2 Diabetes and Hypertension respectively; numeric values

brc, t2d, hypertension Three Cases/Controls Phenotypes, representing Breast Cancer, Type 2 Diabetes and Hypertension respectively; binary values

ldl, bmi, sbp Three Continuous Phenotypes, representing low-density lipoprotein, body-mass index, and systolic blood pressure respectively; numeric values

log_ldl A continuous Phenotype, based on log(ldl) to have a normal distribution; numeric values

sbp_cat An Ordered Categorical Phenotype, with 3 possible outcomes: low, normal or high systolic blood pressure; factor values

Source

https://github.com/VP-biostat/comorbidPGS

`decileboxplot` *Deciles BoxPlot from a PGS Association with a Continuous Phenotype*

Description

`decileboxplot()` take a distribution of PGS, a Continuous Phenotype return a plot with deciles of PGS in x and Boxplot of the Phenotype in y

Usage

```
decileboxplot(df = NULL, prs_col = "SCORESUM", phenotype_col = "Phenotype")
```

Arguments

`df`	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
`prs_col`	a character specifying the PGS column name
`phenotype_col`	a character specifying the Continuous Phenotype column name

Value

return a ggplot object (ggplot2)

densityplot *Density Plot from a PGS Association*

Description

densityplot() take a distribution of PGS, a Phenotype and eventual Confounders return a plot with density of PGS in x by Categories of the Phenotype

Usage

```
densityplot(
  df = NULL,
  prs_col = "SCORESUM",
  phenotype_col = "Phenotype",
  scale = TRUE,
  threshold = NA
)
```

Arguments

df	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
prs_col	a character specifying the PGS column name
phenotype_col	a character specifying the Phenotype column name
scale	a boolean specifying if scaling of PGS should be done before plotting
threshold	a facultative numeric specifying for Continuous Phenotype the Threshold to consider individuals as Cases/Controls as following: • Phenotype > Threshold = Case • Phenotype < Threshold = Control

Value

return a ggplot object (ggplot2)

multiassoc *Multiple PGS Associations from a Data Frame*

Description

multiassoc() take a data frame with distribution(s) of PGS and Phenotype(s), and a table of associations to make from this data frame

return a data frame showing the association results

Usage

```
multiassoc(
  df = NULL,
  assoc_table = NULL,
  scale = TRUE,
  covar_col = NA,
  verbose = TRUE,
  log = "",
  parallel = FALSE,
  num_cores = NA
)
```

Arguments

df	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
assoc_table	a dataframe or matrix specifying the associations to make from df, with 2 columns: PGS and Phenotype (in this order)
scale	a boolean specifying if scaling of PGS should be done before testing
covar_col	a character vector specifying the covariate column names (facultative)
verbose	a boolean (TRUE by default) to write in the console/log messages.
log	a connection, or a character string naming the file to print to. If "" (by default), it prints to the standard output connection, the console unless redirected by sink. If parallel = TRUE, the log will be incomplete
parallel	a boolean, if TRUE, multiassoc() parallelise the association analysis to run it faster (no log available with this option, does not work with Windows machine) If FALSE (default), the association analysis will not be parallelised (useful for debugging process)
num_cores	an integer, if parallel = TRUE (default), multiassoc() parallelise the association analysis to run it faster using num_cores as the number of cores. If nothing is provided, it detects the number of cores of the machine and use num_cores-1

Value

return a data frame showing the association of the PGS(s) on the Phenotype(s) with the following columns:

- PGS: the name of the PGS
- Phenotype: the name of Phenotype
- Phenotype_type: either `'Continuous'`, `'Ordered Categorical'`, `'Categorical'` or `'Cases/Controls'`
- Stat_method: association function detects what is the phenotype type and what is the best way to analyse it, either `'Linear regression'`, `'Binary logistic regression'`, `'Ordinal logistic regression'` or `'Multinomial logistic regression'`
- Covar: list all the covariates used for this association
- N_cases: if Phenotype_type is Cases/Controls, gives the number of cases
- N_controls: if Phenotype_type is Cases/Controls, gives the number of controls
- N: the number of individuals/samples
- Effect: if Phenotype_type is Continuous, it represents the Beta coefficient of linear regression, OR of logistic regression otherwise
- SE: standard error of the related Effect (Beta or OR)
- lower_CI: lower confidence interval of the related Effect (Beta or OR)
- upper_CI: upper confidence interval of the related Effect (Beta or OR)
- P_value: associated P-value

Examples

```
assoc_table <- expand.grid(
  c("t2d_PGS", "ldl_PGS"),
  c("ethnicity","brc","t2d","log_ldl","sbp_cat")
)
results <- multiassoc(
  df = comorbidData,
  assoc_table = assoc_table,
  covar_col = c("age", "sex", "gen_array"),
  parallel = FALSE,
  verbose = FALSE
)
print(results)
```

`multiphenassoc` *Multiple PGS Associations from different Phenotypes*

Description

`multiphenassoc()` take a distribution of PGS and multiple Phenotypes and eventual confounders return a data frame showing the association results

Usage

```
multiphenassoc(
  df = NULL,
  prs_col = "SCORESUM",
  phenotype_col = "Phenotype",
  scale = TRUE,
  covar_col = NA,
  verbose = TRUE,
  log = ""
)
```

Arguments

df	a dataframe with individuals on each row, and at least the following columns: • one ID column, • one PGS column, with numerical continuous values following a normal distribution, • one Phenotype column, can be numeric (Continuous Phenotype), character, boolean or factors (Discrete Phenotype)
prs_col	a character specifying the PGS column name
phenotype_col	a character vector specifying the Phenotype column names
scale	a boolean specifying if scaling of PGS should be done before testing
covar_col	a character vector specifying the covariate column names (facultative)
verbose	a boolean (TRUE by default) to write in the console/log messages.
log	a connection, or a character string naming the file to print to. If "" (by default), it prints to the standard output connection, the console unless redirected by sink.

Value

return a data frame showing the association of the PGS on the Phenotypes with the following columns:

- PGS: the name of the PGS
- Phenotype: the name of Phenotype
- Phenotype_type: either `'Continuous'`, `'Ordered Categorical'`, `'Categorical'` or `'Cases/Controls'`
- Stat_method: association function detects what is the phenotype type and what is the best way to analyse it, either `'Linear regression'`, `'Binary logistic regression'`, `'Ordinal logistic regression'` or `'Multinomial logistic regression'`
- Covar: list all the covariates used for this association
- N_cases: if Phenotype_type is Cases/Controls, gives the number of cases
- N_controls: if Phenotype_type is Cases/Controls, gives the number of controls
- N: the number of individuals/samples
- Effect: if Phenotype_type is Continuous, it represents the Beta coefficient of linear regression; Otherwise, it is the OR of logistic regression

- SE: standard error of the Beta coefficient (if Phenotype_type is Continuous)
- lower_CI: lower confidence interval of the related Effect (Beta or OR)
- upper_CI: upper confidence interval of the related Effect (Beta or OR)
- P_value: associated P-value

Index