Package 'generalize'

July 5, 2017

Type Package

Title Test generalization of association analysis from study1 to study2 and display results

Version 1.2

Date 2017-07-05

Author Tamar Sofer

Maintainer Tamar Sofer <tsofer.uw@gmail.com>

Description Given multiple SNP-trait testing results from two studies, when one is considered the discovery study (study1) and the other (study2) is the follow-up, calculate r-values to quantify the strength of association between each of the SNPs and the trait. If effect sizes are available results can be displayed in a figure.

License GPL-2

LazyData true

Suggests ggplot2, gridExtra, RColorBrewer

R topics documented:

generalize-package 1
dat 3
matchEffectAllele 4
prepareGenResFigure 5
r.value 7
testGeneralization 8

Index **10**

`generalize-package` *Test generalization of association analysis from study1 to study2 and display results*

Description

Given multiple SNP-trait testing results from two studies, when one is considered the discovery study (study1) and the other (study2) is the follow-up, calculate r-values to quantify the strength of association between each of the SNPs and the trait. If effect sizes are available results can be displayed in a figure.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
Code used to perform generalization testing for a set of association from study 1 (the discovery study) to study 2 (the follow-up, generalization study).

Author(s)

Tamar Sofer

Maintainer:Tamar Sofer <tsofer.uw@gmail.com>

Examples

```
data(dat)
head(dat)
## first match effects on alleles:
dat.matched <- matchEffectAllele(dat$rsID,  study2.effect = dat$study2.beta,
study1.alleleA = dat$study1.alleleA, study2.alleleA = dat$study2.alleleA,
study1.alleleB = dat$study1.alleleB, study2.alleleB = dat$study2.alleleB)


dat$study2.beta <- dat.matched$study2.effect
dat$alleleA <- dat$study1.alleleA
dat$alleleB <- dat$study1.alleleB
dat$study1.alleleA <- dat$study1.alleleB <- dat$study2.alleleA <- dat$study2.alleleB <- NULL

## test for generalization:

gen.res <- testGeneralization(dat$rsID, dat$study1.pval, dat$study2.pval,
dat$study1.n.test[1], study1.effect = dat$study1.beta,
study2.effect = dat$study2.beta, directional.control = TRUE,
control.measure = "FDR" )
head(gen.res)

require(ggplot2)
require(gridExtra)
require(RColorBrewer)


## prepare figure
prepareGenResFigure(dat$rsID, dat$study1.beta, dat$study1.se, dat$study2.beta,
dat$study2.se, gen.res$generalized, gen.res$gen.rvals, dat$study1.n.test[1],
output.file = paste0(getwd(), "/Generalization_example.pdf"),
study1.name = "Study1", study2.name = "Study2")

## marking genome-wide significant hits in study 2 with red stars
## (here nothing is genome-wide significant so there will be no difference!)
prepareGenResFigure(dat$rsID, dat$study1.beta, dat$study1.se, dat$study2.beta,
dat$study2.se, gen.res$generalized, gen.res$gen.rvals, dat$study1.n.test[1], plot.width = 10,
output.file = paste0(getwd(), "/Generalization_example.pdf"), study1.name = "Study1",
study2.name = "Study2", add.stars = 5e-8)
```

dat *A data set of previouly published SNP-platelet count associations in Gieger et al., 2011, and association results for the same SNPs in the HCHS/SOL.*

Description

A data set with SNP associations reported for platelet counts, in study1 being Geiger et al., 2011, study2 the HCHS/SOL.

Usage

```
data("dat")
```

Format

A data frame with 42 observations on the following 15 variables.

`rsID` SNP identifier

`chromosome` chromosome

`position` position in human build 37

`study1.alleleA` effect Allele in study 1

`study1.alleleB` other Allele in study 1

`study1.beta` estimated effect size is study 1

`study1.se` SE of beta in study 1

`study1.pval` two-sided p-value in study 1

`study1.n.test` number of tested SNPs in study 1

`study2.alleleA` effect Allele in study 2

`study2.alleleB` other Allele in study 2

`study2.beta` estimated effect size is study 2

`study2.se` SE of beta in study 2

`study2.pval` two-sided p-value in study 2

`Ref` Reference for the platelet GWAS paper from which the study1 results are taken - Gieger,2011.

Details

This data set provides the top results from a platelet GWAS reported in Gieger et al., 2011, and the association testing results for the same SNP in the HCHS/SOL stuy. These results will soon be published in the American Journal of Human Genetics (as of Dec 2015), by Schick et al.

Source

Gieger, Christian, et al. "New gene functions in megakaryopoiesis and platelet formation." Nature 480.7376 (2011): 201-208, Schick, Ursula M., et al. "Genome-wide association study of platelet count identifies ancestry-specific loci in Hispanic/Latino Americans." The American Journal of Human Genetics 98.2 (2016): 229-242.

Examples

```
data(dat)
head(dat)
```

`matchEffectAllele` *check and update signs of estimated effect sizes in study 2 by matching alleles between study 1 and study 2*

Description

Compare effect alleles and other alleles and change the sign of the effect in study2 in cases were the effect alleles are different between studies.

Usage

```
matchEffectAllele(snpID, study2.effect, study1.alleleA, study2.alleleA,
study1.alleleB = NULL, study2.alleleB = NULL)
```

Arguments

`snpID`	a vector of snp identifiers (e.g. rsIDs)
`study2.effect`	Numeric vector of the SNP effects from study2, or their directions given as 1, -1.
`study1.alleleA`	Character vector providing the effect alleles corresponding to snpID in study1.
`study2.alleleA`	Character vector providing the effect alleles corresponding to snpID in study2.
`study1.alleleB`	Character vector providing the non-effect (other) alleles corresponding to snpID in study1
`study2.alleleB`	Character vector providing the non-effect (other) alleles corresponding to snpID in study2.

Details

Returns a data frame matching each snpID with its estimated effect size in study2, after making sure that both studies use the same effect alleles for all given SNPs. If there is enough information, strand-abgiuouos SNPs flag is also given.

Value

Returns a data frame with SNP identifiers, direction of effects of study2 after possibly flipping the directions of effects of study2 so the same effect allele is used in both studyes. The returned data frame also has an indicator of whether the direction of association was flipped based on allele matched. If it was, and you would like to provide the correct effect allele frequencies associated with the follow-up study effect sizes, make sure to change them (by eaf[which(flip)] <- 1-eaf[which(flip)]) for the flipped effect sizes. If effect estimates were given in study1.direction and study2.direction, then effect estimates are returned accordingly.

Author(s)

Tamar Sofer

Examples

```
data(dat)
head(dat)
## match effects on alleles:

dat.matched <- matchEffectAllele(dat$rsID, study2.effect = dat$study2.beta,
study1.alleleA = dat$study1.alleleA, study2.alleleA = dat$study2.alleleA,
study1.alleleB = dat$study1.alleleB, study2.alleleB = dat$study2.alleleB)

head(dat.matched)
```

`prepareGenResFigure` *Prepare a figure illustrating generalization testing results*

Description

The function takes vectors of effect sizes and standard errors from study1 and study2, and generalization testing results in the form of r-values and generalization status, corresponding to a set of SNPs selected for follow-up after association testing in study1. The function creates a figure to visualize these results, and the observed effect sizes and confidence intervals. The confidence intervals are scaled to control the false coverage rate (FCR) at the 95

Usage

```
prepareGenResFigure(snpID, study1.beta, study1.se, study2.beta, study2.se,
generalized, gen.rvals, study1.n.test, output.file, study1.name = "Study1",
study2.name = "Study2", make.title = FALSE, legend.position = c(0.9, 0.9),
CI.line.width = 1.7, x.ticks.size = 15, plot.width = 8, plot.height = 8,
trait = NULL, plot.title = NULL, add.stars = NULL, alpha.level.follow = 0.05)
```

Arguments

`snpID`	a vector of snp identifiers (e.g. rsIDs)
`study1.beta`	Numeric vector of the effect estimates from study1.
`study1.se`	Numeric vector of the SE of the effect estimates from study1.
`study2.beta`	Numeric vector of the effect estimates from study2.
`study2.se`	Numeric vector of the SE of the effect estimates from study2.
`generalized`	Generalization status, return by the function testGeneralization
`gen.rvals`	r-values from generalization testing. Returned by the function testGeneralization.
`study1.n.test`	Number of SNPs tested in study 1.
`output.file`	Name of file for the figure.
`study1.name`	A string, name of study1.
`study2.name`	A string, name of study2.
`make.title`	Should the plot have a title?
`legend.position`	Legend position.

CI.line.width	the width of the line of the confidence intervals.
x.ticks.size	The size of the snp identifiers strings given at the bottom of the figure (at the X-axis).
plot.width	Figure width.
plot.height	Figure height.
trait	An optional string with the name of the trait associated (or not) with the SNPs. Used in the figure title if given.
plot.title	An optional string of a spefic title
add.stars	An optional numerical value, corrsponding to a p-value threshold. If a SNP from study2 has a two-sided p-value that is smaller than this threshold, than a star is plotted next to the estimated effect of the appropriate SNP.
alpha.level.follow	Significance level for plotting confidence interval. Defult level is 0.05 (plots 95% confidence intervals)

Details

A figure is created in output.file.

Examples

```
data(dat)
head(dat)

## first match effects on alleles:
dat.matched <- matchEffectAllele(dat$rsID, study2.effect = dat$study2.beta,
study1.alleleA = dat$study1.alleleA, study2.alleleA = dat$study2.alleleA,
study1.alleleB = dat$study1.alleleB, study2.alleleB = dat$study2.alleleB)

head(dat.matched)

dat$study2.beta <- dat.matched$study2.effect

## test for generalization:

gen.res <- testGeneralization(dat$rsID, dat$study1.pval, dat$study2.pval,
dat$study1.n.test[1], study1.effect = dat$study1.beta,
study2.effect = dat$study2.beta, directional.control = TRUE,
control.measure = "FDR" )

## prepare figure
require(ggplot2)
require(gridExtra)
require(RColorBrewer)

## prepare figure
prepareGenResFigure(dat$rsID, dat$study1.beta, dat$study1.se, dat$study2.beta,
dat$study2.se, gen.res$generalized, gen.res$gen.rvals, dat$study1.n.test[1],
output.file = paste0(getwd(), "/Generalization_example.pdf"),
study1.name = "Study1", study2.name = "Study2")

## marking genome-wide significant hits in study 2 with red stars
## (here nothing is genome-wide significant so there will be no difference!)
prepareGenResFigure(dat$rsID, dat$study1.beta, dat$study1.se, dat$study2.beta,
```

```
dat$study2.se, gen.res$generalized, gen.res$gen.rvals, dat$study1.n.test[1],
output.file = paste0(getwd(), "/Generalization_example.pdf"),
study1.name = "Study1", study2.name = "Study2", add.stars = 5e-8)
```

`r.value` *r-value computation*

Description

The function computes r-values given two vectors of p-values from primary and follow-up studies. r-values assess the False Discovery Rate (FDR) of repilcability claims across the primary and follow-up studies. This is a function from Ruth Heller, adapted to compute FWER r-values in addition to FDR r-values.

Usage

```
r.value(p1, p2, m, c2 = 0.5, control.measure = "FDR", l00 = 0.8 , variation = c("none","use.m.star"
```

Arguments

`p1`	Numeric vector of the p-values from study 1
`p2`	Numeric vector of the p-values from study 2
`m`	Number of features examined in the primary study.
`c2`	Parameter for relative boost to the p-values from the primary study. 0.5 (default) is recommended, since was observed in simulations to yield similar power to procedure with the optimal value (which is unknown for real data).
`control.measure`	A sting, either FDR or FWER, depending on the desired measure of control on false generalizations.
`l00`	Lower bound of the fraction of features (out of m) with true null hypotheses in both studies. For example, for GWAS on the whole genome, the choice of 0.8 is conservative in typical applications.
`variation`	When 'use.m.star' is selected m* is used. m* is defined as follows: $m^* = m\sum_{i=1}^{m}\frac{1}{i}$. When 'use.t' is selected c1 is computed given the threshold tt. Both variations guarantee that the procedure that decleares all r-values below q as replicability claims, controls the FDR at level q, for any type of dependency of the p-values in the primary study. default is 'none'.
`tt`	The selection rule threshold for p-values from the primary study. must be supplied when variation 'use.t' is selected.
`Q`	The level or false generalization (e.g. control FDR at the q level or FWER at the q level).

Author(s)

Ruth Heller, Shay Yaacoby (shay66@gmail.com), small adaptation by Tamar Sofer.

Examples

```
# General example from Ruth Heller's website:
  pv <- read.csv("http://www.math.tau.ac.il/~ruheller/Software/CrohnExample.csv")
  rv <- r.value(p1=pv$p1,p2=pv$p2,m=635547,c2=0.5,l00=0.8)
  rv2 <- r.value(p1=pv$p1,p2=pv$p2,m=635547,c2=0.5,l00=0.8,variation="use.t",tt=1e-5)

#### in this package, the function is called by testGenerelization.
```

`testGeneralization` *Test for generalization of associations from study1 to study2*

Description

The function calls another function that computes r-values given two vectors of p-values from primary (study1) and follow-up (study2) studies and determine generalization (TRUE/FALSE) for each of the associations. The computes either FDR or FWER r-values. Directional control is applied by default, so that one-sided p-values are computed and used.only and associations with the same sign in the two studies are generalized.

Usage

```
testGeneralization(snpID, study1.pval, study2.pval, study1.n.test,
study1.effect = NULL,  study2.effect = NULL, directional.control = TRUE,
control.measure = "FDR", q = 0.05, l00 = 0.8, c2 = 0.5,
variation = c("none","use.m.star","use.t"), tt = NULL,  verbose = FALSE)
```

Arguments

`snpID`	a vector of snp identifiers (e.g. rsIDs)
`study1.pval`	Numeric vector of the two sided p-values from study1.
`study2.pval`	Numeric vector of the two sided p-values from study2.
`study1.n.test`	Number of SNPs tested in study 1.
`study1.effect`	An optional numeric vector of SNP effects from study1, or their directions as 1, -1.\ Required for directional control using 1-sided p-values.
`study2.effect`	An optional numeric vector of SNP effects from study2, or their directions as 1, -1.\ Required for directional control using 1-sided p-values.
`directional.control`	should directional controlled by applied? i.e., a SNP could be generalized only if it has the same direction of association in both study1 and study2.
`control.measure`	Either "FDR" or "FWER", according to the desired control of false generalizations.
`q`	The level or false generalization (e.g. control FDR at the q level or FWER at the q level).
`l00`	Lower bound of the fraction of features (out of m) with true null hypotheses in both studies. For example, for GWAS on the whole genome, the choice of 0.8 is conservativeeffect in typical applications.

c2 — Parameter for relative boost to the p-values from the primary study. 0.5 (default) is recommended, since was observed in simulations to yield similar power to procedure with the optimal value (which is unknown for real data).

variation — When 'use.m.star' is selected m* is used. m* is defined as follows: $m^* = m \sum_{i=1}^{m} \frac{1}{i}$. When 'use.t' is selected c1 is computed given the threshold tt. Both variations guarantee that the procedure that decleares all r-values below q as replicability claims, controls the FDR at level q, for any type of dependency of the p-values in the primary study. default is 'none'.

tt — The selection rule threshold for p-values from the primary study. must be supplied when variation 'use.t' is selected.

verbose

Details

Returns a data frame matching each snpID with its computed r-value and generalization status.

Value

a data frame matching each snpID with its computed r-value and generalization status.

Author(s)

Tamar Sofer

Examples

```
data(dat)
head(dat)
## first match effects on alleles:

dat.matched <- matchEffectAllele(dat$rsID,  study2.effect = dat$study2.beta,
study1.alleleA = dat$study1.alleleA, study2.alleleA = dat$study2.alleleA,
study1.alleleB = dat$study1.alleleB, study2.alleleB = dat$study2.alleleB)


head(dat.matched)

dat$study2.beta <- dat.matched$study2.effect

## test for generalization:

gen.res <- testGeneralization(dat$rsID, dat$study1.pval, dat$study2.pval, dat$study1.n.test[1],
study1.effect = dat$study1.beta, study2.effect = dat$study2.beta,
directional.control = TRUE, control.measure = "FDR" )
```

Index

∗Topic **datasets**
dat, 3

dat, 3

generalize *(*generalize-package*)*, 1
generalize-package, 1

matchEffectAllele, 4

prepareGenResFigure, 5

r.value, 7

testGeneralization, 8