Package 'BinomiRare'

October 6, 2016

Type Package

Title Test the association of rare variants with a disease

Version 1.0

Date 2016-10-06

Author Tamar Sofer

Maintainer Tamar Sofer <tsofer@uw.edu>

Description

Given estimated probabilities of disease obtained from an entire sample, for each rare variant it tests whether the number of diseased individuals is consistent with the expected number of diseased individuals. It performs meta-analysis across several studies as well.

License GPL-2

Imports poibin

R topics documented:

`BinomiRare-package` *Test the association of rare variants with a disease*

Description

Given estimated probabilities of disease obtained from an entire sample, for each rare variant it tests whether the number of diseased individuals is consistent with the expected number of diseased individuals. It performs meta-analysis across several studies as well.

Details

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

Index: This package was not yet installed at build time.
Functions to test whether the number of diseased individuals among carriers of rare variants is consistent with expectation, where the expectation is determined via any pre-specified disease probability model. It also included functions to export data to be used in meta-analysis, and to perform meta-analysis across several studies. In this version, not all possible quality check are implemented (e.g. the meta-analysis does not flip alleles, and uses information for a given variant between studies only if their effect alleles match).

Author(s)

Tamar Sofer

Maintainer: Tamar Sofer Tamar Sofer <tsofer@uw.edu>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

Examples

```
require(poibin)

##########  Example 1:  a single data set.
########## Simulate data
n <- 10000
effect.size <- 1
pop.risk <- -2.6

x <- rnorm(n, sd = 0.01)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ## one causal variant, x is a confounder
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)
names(d) <- paste0("person_", 1:nrow(G))


########### Now that we have outcome d, genotypes G and a covariate x:
########### Estimate disease probability model

prob.mod <- glm(d ~ x, family = binomial)
prob.d <- expit(predict(prob.mod))

system.time(res <- BRtest(d, prob.d, G)) ### super quick
```

```
######### If we wanted to contribute this data to meta-analysis
######### here we need annotation as well!
### simulate annotation:
variant.annot <- data.frame(variant = paste0("var_", 1:500), chromosome = 1,
position = 1:500, alleleA = sample(c("A", "C", "G", "T"), size = 500, replace = TRUE),
other.alleles = NA, stringsAsFactors = FALSE)

for (i in 1:nrow(variant.annot)){
variant.annot$other.alleles[i] <- sample(setdiff(c("A", "C", "G", "T"),
variant.annot$alleleA[i]), size = 1)
}

res <- prepareForMetaBRtest(d, prob.d, G, variant.annot, output.file =
"carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)

### CHECK: if we had missing annotation for variant 1:
#res <- prepareForMetaBRtest(d, prob.d, G, variant.annot[-1,], output.file =
#  "carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)
### CHECK: if the variant.annot data frame did not have a necessary column:
#res <- prepareForMetaBRtest(d, prob.d, G, variant.annot[,-1], output.file =
# "carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)


##########  Example 2:  simulate multiple data set, and then meta-analyze them.

########## First simulate data:
dir.create(file.path(getwd(), "Files_for_meta"))
setwd(file.path(getwd(), "Files_for_meta"))

n <- 10000
effect.size <- 0
n.in.pop <- n/5
pop.risk <- c(rep(-2.6, n.in.pop), rep(-2.2, n.in.pop), rep(-2, n.in.pop),
rep(-1.8, n.in.pop) , rep(-1.6, n.in.pop) )
pop.inds <- c(rep(1, n.in.pop), rep(2, n.in.pop), rep(3, n.in.pop),
rep(4, n.in.pop) , rep(5, n.in.pop) )

### for each population, prepare information for meta-analysis and write to file.
x <- rnorm(n, sd = 0.02)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ##
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)

for (pop in 1:5){
d.pop <- d[which(pop.inds == pop)]
G.pop <- G[which(pop.inds == pop),]
x.pop <- x[which(pop.inds == pop)]
prob.mod <- glm(d.pop ~ x.pop, family = binomial)
prob.d <- expit(predict(prob.mod))
prepareForMetaBRtest(d.pop, prob.d, G.pop, variant.annot, output.file =
```

```
file.path(getwd(), paste0("carriers_prob_study_", pop, ".csv")), test = FALSE,
return.result.object = FALSE)
}


############## Now meta-analyze results from files:
results.folder <- getwd()

res <- BRtestMeta(folder = results.folder, return.result.object = TRUE)
```

BRtest *Tests the association between a set of variants and disease status.*

Description

Tests a set of variants represented in the numberical matrix G, given disease status and previously estimated probabilities of disease.

Usage

```
BRtest(d, probs, G)
```

Arguments

d	An n vector of disease status
probs	An n vector of (estimated) disease probabilities corresponding to the individuals with disease status.
G	An nxp matrix of counts of (assumed rare) p variants genotpyed on n individuals. Columns correspond to variants, rows to individuals.

Details

For each variant, calculates the expected number of disease individuals among the carriers, and calculates p-value from the BinomiRare test of association between the variant and the disease status.

Value

The function returns a data frame with variant name, the number of carriers of the variant, number of diseased carriers, the expected number of diseased carriers (according to the supplied probabilities), and p-value.

Author(s)

Tamar Sofer

Examples

```
require(poibin)

##########  Example 1:  a single data set.
########## Simulate data
n <- 10000
effect.size <- 1
pop.risk <- -2.6

x <- rnorm(n, sd = 0.01)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ## one causal variant, x is a confounder
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)
names(d) <- paste0("person_", 1:nrow(G))


########### Now that we have outcome d, genotypes G and a covariate x:
########### Estimate disease probability model

prob.mod <- glm(d ~ x, family = binomial)
prob.d <- expit(predict(prob.mod))

system.time(res <- BRtest(d, prob.d, G)) ### super quick
```

`BRtestMeta` *Applies BinomiRare (BR) test in meta-analysis based on previously prepared files from multiple studies.*

Description

Given a folder with files of data prepared for BR meta-analysis, merge the files together and applies the BR test.

Usage

```
BRtestMeta(folder, recursive = TRUE, output.file = "BR_meta_results.csv",
error.log.file = "BR_meta_inconsistencies.txt", return.result.object = FALSE)
```

Arguments

`folder`	A folder in which the files from the various studies are at. Only .csv files are considered.
`recursive`	Should folders with the main folders be looked at?
`output.file`	File with meta-analysis test results.
`error.log.file`	File for which the meta-analysis function prints identified errors (e.g. non-matching alleles, etc).

return.result.object
: Should an R data.frame with results be returned, in addition the printing of the results to file?

Value

prints results to file, and if return.result.object == TRUE, also returns a data.frame with columns: variant (variant name), n.carrier (total across studies), n.D.carrier (total across studies), chromosome, position, alleleA, other.alleles, n.studies (the number of studied that had individuals carrying the variant), pval (BR p-value), and expected.n.D.carrier (the expected number of diseased individuals among the carriers).

Note

other.allele is still not used. It is assume that the same effect allele (alleleA) is tested by all studies that privde it. If there's no match in annotation for a given vairant, information about the variant is thrown away.

Author(s)

Tamar Sofer

Examples

```
require(poibin)
dir.create(file.path(getwd(), "Files_for_meta"))
setwd(file.path(getwd(), "Files_for_meta"))


variant.annot <- data.frame(variant = paste0("var_", 1:500), chromosome = 1,
position = 1:500, alleleA = sample(c("A", "C", "G", "T"),
size = 500, replace = TRUE), other.alleles = NA, stringsAsFactors = FALSE)

for (i in 1:nrow(variant.annot)){
variant.annot$other.alleles[i] <- sample(setdiff(c("A", "C", "G", "T"),
variant.annot$alleleA[i]), size = 1)
}


n <- 10000
effect.size <- 0
n.in.pop <- n/5
pop.risk <- c(rep(-2.6, n.in.pop), rep(-2.2, n.in.pop), rep(-2, n.in.pop),
rep(-1.8, n.in.pop) , rep(-1.6, n.in.pop) )
pop.inds <- c(rep(1, n.in.pop), rep(2, n.in.pop), rep(3, n.in.pop),
rep(4, n.in.pop) , rep(5, n.in.pop) )

### for each population, prepare information for meta-analysis and write to file.
x <- rnorm(n, sd = 0.02)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ##
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
```

```
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)

for (pop in 1:5){
d.pop <- d[which(pop.inds == pop)]
G.pop <- G[which(pop.inds == pop),]
x.pop <- x[which(pop.inds == pop)]
prob.mod <- glm(d.pop ~ x.pop, family = binomial)
prob.d <- expit(predict(prob.mod))
prepareForMetaBRtest(d.pop, prob.d, G.pop, variant.annot, output.file =
file.path(getwd(), paste0("carriers_prob_study_", pop, ".csv")),
test = FALSE, return.result.object = FALSE)
}


############# Now meta-analyze results from files:
results.folder <- getwd()

res <- BRtestMeta(folder = results.folder, return.result.object = TRUE)
```

`expit` *The expit function*

Description

returns the expit of a variable x expit(x) = exp(x)/(1 + exp(x))

Usage

```
expit(x)
```

Arguments

`x`	a numerical vector

Details

applies the expit transformation

Value

expit(x)

Note

Useful for calculating disease probabilities from logistic regression.

Author(s)

Tamar Sofer

Examples

```
n <- 10000
effect.size <- 1
pop.risk <- -2.6

x <- rnorm(n, sd = 0.01)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ## one causal variant, x is a confounder
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)
names(d) <- paste0("person_", 1:nrow(G))


########### Now that we have outcome d, genotypes G and a covariate x:
########### Estimate disease probability model

prob.mod <- glm(d ~ x, family = binomial)
prob.d <- expit(predict(prob.mod))
```

---

logit *The logit function*

Description

Given a numerical variable x, returns it logit transfomation (inverse of expit)

Usage

```
logit(x)
```

Arguments

x	A numerical vector

Details

logit(x) = log(x/(1-x)) This is related to the logistic model of disease probability

Value

a numerical vector

Author(s)

Tamar Sofer

Examples

```
x <- rnorm(100)
all(abs(logit(expit(x)) - x) < 1e-7)
```

poibin.midp *Calculated the mid-p values based on the Poisson-Binomial distribution*

Description

Given a vector of disease probabilities and a number of diseased individual, Calculated the mid-p values based on the Poisson-Binomial distribution

Usage

```
poibin.midp(n.carrier, n.D.carrier, prob.vec)
```

Arguments

n.carrier	The number of carriers of a rare variant
n.D.carrier	The number of diseased carriers of a rare variant. n.D.carrier cannot be larger than n.carrier.
prob.vec	vector of disease probabilities of the carriers.

Value

a single numeric variable - a p-value for the test that the number n.D.carrier is consistent with prob.vec.

Note

Althouth n.carrier is not strictly needed, but is useful for quality checks, especially when meta-analyzing.

Author(s)

Tamar Sofer

Examples

```
require(poibin)
n <- 100

### simulation under the null
g <- rbinom(n, 2, 0.1)
x <- rnorm(n)
p <- expit(-2.3 + x)
d <- rbinom(n, size = 1, prob = p)

mod <- glm(d ~ x, family = "binomial")
prob.d <- expit(predict(mod))
poibin.midp(n.carrier = sum(g >0 ), n.D.carrier = sum(g*d > 0),
prob.vec = prob.d[which(g>0)] )

##### under the alternative:
p <- expit(-2.3 + x + g)
d <- rbinom(n, size = 1, prob = p)
```

```
mod <- glm(d ~ x, family = "binomial")
prob.d <- expit(predict(mod))
poibin.midp(n.carrier = sum(g >0 ), n.D.carrier = sum(g*d > 0),
prob.vec = prob.d[which(g>0)] )
```

`prepareForMetaBRtest` *Prepares a file with information required for meta-analysis of Binomi-Rare test.*

Description

Based on vectors of disease status, pre-computed disease probabilities, a matrix of genetic variants, and variant annotation, writes a file with information for meta-anlaysis.

Usage

```
prepareForMetaBRtest(d, probs, G, variant.annot, output.file = "carriers_prob_dat.csv",
test = FALSE, return.result.object = FALSE)
```

Arguments

`d`	An n vector of disease status
`probs`	An n vector of (estimated) disease probabilities corresponding to the individuals with disease status.
`G`	An nxp matrix of counts of (assumed rare) p variants genotpyed on n individuals. Columns correspond to variants, rows to individuals.
`variant.annot`	A data-frame with annotation of the genetic variants under considerations. It has to have column named "variant" (variant name), "chromosome", "position", "alleleA" (effect allele), and "other.alleles" (the alleles that aren't tested).
`output.file`	Name of file to print information to.
`test`	In addition to printing information to file, should the BR test be performed?
`return.result.object`	If return.result.object is true, R will return a data.frame with results, in addition to printing them to file.

Value

a matrix is printed to the file output.file. In addition, if requested a data frame is returned. Returned/printed columns are variant, chromosome, position, alleleA, other.alleles, n.carrier, n.D.carrier, and if test == TRUE, also expected.n.D.carrier (expected number of diseased carriers), and pval (the p-value from the BinomiRare test).

Note

other.alleles are currently not used, but in the future there will be added functionality that may flip alleles if needed, or return information and alleles do not match, or when triallelic SNPs are used.

Author(s)

Tamar Sofer

Examples

```
require(poibin)

##########  Example 1:  a single data set.
########## Simulate data
n <- 10000
effect.size <- 1
pop.risk <- -2.6

x <- rnorm(n, sd = 0.01)
x <- pmax(x, 0)

g <- rbinom(n, size = 2, prob = x)  ## one causal variant, x is a confounder
G <- matrix(rbinom(n*100, size = 1, prob = 0.001), nrow = n)  ### another 100 null variants
G <- cbind(g, G)
colnames(G) <- paste0("var_", 1:ncol(G))
rownames(G) <- paste0("person_", 1:nrow(G))
p <- expit(pop.risk +   g*effect.size + 20*x)
d <- rbinom(n, 1, p)
names(d) <- paste0("person_", 1:nrow(G))


############ Now that we have outcome d, genotypes G and a covariate x:
############ Estimate disease probability model

prob.mod <- glm(d ~ x, family = binomial)
prob.d <- expit(predict(prob.mod))

system.time(res <- BRtest(d, prob.d, G)) ### super quick


######### If we wanted to contribute this data to meta-analysis
######### here we need annotation as well!
### simulate annotation:
variant.annot <- data.frame(variant = paste0("var_", 1:500), chromosome = 1,
position = 1:500, alleleA = sample(c("A", "C", "G", "T"), size = 500, replace = TRUE),
other.alleles = NA, stringsAsFactors = FALSE)

for (i in 1:nrow(variant.annot)){
variant.annot$other.alleles[i] <- sample(setdiff(c("A", "C", "G", "T"),
variant.annot$alleleA[i]), size = 1)
}

res <- prepareForMetaBRtest(d, prob.d, G, variant.annot, output.file =
"carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)
### CHECK: if we had missing annotation for variant 1:
# res <- prepareForMetaBRtest(d, prob.d, G, variant.annot[-1,],
# output.file = "carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)
### CHECK: if the variant.annot data frame did not have a necessary column:
#res <- prepareForMetaBRtest(d, prob.d, G, variant.annot[,-1], output.file =
# "carriers_prob_dat.csv", test = TRUE, return.result.object = TRUE)
```

Index