Package 'GhostKnockoff'

October 12, 2022

Type Package

Title The Knockoff Inference Using Summary Statistics

Version 0.1.0

Date 2021-11-20

Author Zihuai He <zihuai@stanford.edu>

Maintainer Zihuai He <zihuai@stanford.edu>

Description Functions for multiple knockoff inference using summary statistics, e.g. Z-scores. The knockoff inference is a general procedure for controlling the false discovery rate (FDR) when performing variable selection. This package provides a procedure which performs knockoff inference without ever constructing individual knockoffs (GhostKnockoff). It additionally supports multiple knockoff inference for improved stability and reproducibility. Moreover, it supports meta-analysis of multiple overlapping studies.

License GPL-3

Encoding UTF-8

Depends R(>= 3.5.0), Matrix, CVXR, Rdsdp, gtools, seqminer

Imports RSpectra, corpcor

NeedsCompilation no

Repository CRAN

Date/Publication 2021-11-23 08:20:02 UTC

R topics documented:

GhostKnockoff.filter 2
GhostKnockoff.fit 3
GhostKnockoff.GetCorStudy 5
GhostKnockoff.prelim 6
GhostKnockoff.prelim.Meta 7

Index **9**

GhostKnockoff.filter *GhostKnockoff filter*

Description

This function calculates Q-values to select variants associated with the outcome, given the feature importance scores

Usage

```
GhostKnockoff.filter(T_0,T_k)
```

Arguments

T_0	A p*1 matrix. The feature importance score for the original variants.
T_k	A p*M matrix where M is the number of hypothetical knockoffs. The knockoff feature importance scores.

Value

q	A vector of length p, where each element is a Q-value. Variants with q <= target FDR will be selected.
kappa	A vector of length p. Multiple knockoff statistics kappa.
tau	A vector of length p. Multiple knockoff statistics tau.

Examples

```
# We use genetic data as an example
library(GhostKnockoff)

# load example vcf file from package "seqminer", this serves as the reference panel
vcf.filename = system.file("vcf/1000g.phase1.20110521.CFH.var.anno.vcf.gz", package = "seqminer")

## this is how the actual genotype matrix from package "seqminer" looks like
example.G <- t(readVCFToMatrixByRange(vcf.filename, "1:196621007-196716634",annoType='')[[1]])
example.G <- example.G[,apply(example.G,2,sd)!=0]
example.G <- example.G[,1:100]

# compute correlation among variants
cor.G<-matrix(as.numeric(corpcor::cor.shrink(example.G)), nrow=ncol(example.G))

# fit null model
fit.prelim<-GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)

### if only one study is involved
Zscore_0<-as.matrix(rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
```

```
n.study<-c(5000)

# knockoff scores for each block, this can be run in parallel too
GK.stat<-GhostKnockoff.fit(Zscore_0,n.study,fit.prelim,gamma=1,weight.study=NULL)

# knockoff filter
GK.filter<-GhostKnockoff.filter(GK.stat$T_0,GK.stat$T_k)
GK.filter$q

### if multiple studies are involved, for a meta-analysis

# compute study correlation
Zscore_0<-cbind(rnorm(nrow(cor.G)),rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
cor.study<-GhostKnockoff.GetCorStudy(Zscore_0,fit.prelim)

# note that all steps above can be run in parallel for nonoverlapping blocks of the genome.
# Then the overall study correlation can be computed by averaging cor.study of all blocks.

# compute optimal weights and study dependency using overall study correlaton
n.study<-c(5000,7500)
Meta.prelim<-GhostKnockoff.prelim.Meta(cor.study, n.study)
gamma<-Meta.prelim$gamma
weight.study<-Meta.prelim$w_opt

# knockoff scores for each block, this can be run in parallel too
GK.stat<-GhostKnockoff.fit(Zscore_0,n.study,fit.prelim,gamma=gamma,weight.study=weight.study)

# knockoff filter
GK.filter<-GhostKnockoff.filter(GK.stat$T_0,GK.stat$T_k)
GK.filter$q
```

`GhostKnockoff.fit` *Feature importance score generator*

Description

Generate original and knockoff feature importance scores given original Z-scores from multiple studies.

Usage

```
GhostKnockoff.fit(Zscore_0,n.study,fit.prelim,gamma=1,weight.study=NULL)
```

Arguments

`Zscore_0`	A p*K Z-score matrix, where p is the number of variants and K is the number of studies. Variants not observed in the study should be coded as NA.

n.study	A vector of length K, where each element is the study sample size.
fit.prelim	The output of function "GhostKnockoff.prelim".
gamma	The estimated study dependency. See functon "GhostKnockoff.GetGamma".
weight.study	The weights to combine multiple studies. The default is based on sample size. The optimal weights can be estimated using function "GhostKnockoff.prelim.Meta".

Value

GK.Zscore_0	A vector of length p, where each element is weighted combination of original Z-scores
GK.Zscore_k	A p*M matrix, where each column is a knockoff copy of GK.Zscore_0.
T_0	Feature importance score of original data: square of GK.Zscore_0.
T_k	Feature importance score of knockoff copies: square of GK.Zscore_k.

Examples

```
# We use genetic data as an example
library(GhostKnockoff)

# load example vcf file from package "seqminer", this serves as the reference panel
vcf.filename = system.file("vcf/1000g.phase1.20110521.CFH.var.anno.vcf.gz", package = "seqminer")

## this is how the actual genotype matrix from package "seqminer" looks like
example.G <- t(readVCFToMatrixByRange(vcf.filename, "1:196621007-196716634",annoType='')[[1]])
example.G <- example.G[,apply(example.G,2,sd)!=0]
example.G <- example.G[,1:100]

# compute correlation among variants
cor.G<-matrix(as.numeric(corpcor::cor.shrink(example.G)), nrow=ncol(example.G))

# fit null model
fit.prelim<-GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)

### if only one study is involved
Zscore_0<-as.matrix(rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
n.study<-c(5000)

# knockoff scores for each block, this can be run in parallel too
GK.stat<-GhostKnockoff.fit(Zscore_0,n.study,fit.prelim,gamma=1,weight.study=NULL)

### if multiple studies are involved, for a meta-analysis

# compute study correlation
Zscore_0<-cbind(rnorm(nrow(cor.G)),rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
cor.study<-GhostKnockoff.GetCorStudy(Zscore_0,fit.prelim)

# note that all steps above can be run in parallel for nonoverlapping blocks of the genome.
```

```
# Then the overall study correlation can be computed by averaging cor.study of all blocks.

# compute optimal weights and study dependency using overall study correlaton
n.study<-c(5000,7500)
Meta.prelim<-GhostKnockoff.prelim.Meta(cor.study, n.study)
gamma<-Meta.prelim$gamma
weight.study<-Meta.prelim$w_opt

# knockoff scores for each block, this can be run in parallel too
GK.stat<-GhostKnockoff.fit(Zscore_0,n.study,fit.prelim,gamma=gamma,weight.study=weight.study)
```

GhostKnockoff.GetCorStudy *Calculate study correlation*

Description

This function computes correlation among studies given Z-scores and the output of GhostKnockoff.prelim.

Usage

```
GhostKnockoff.GetCorStudy(Zscore_0, fit.prelim)
```

Arguments

Zscore_0	A p*K Z-score matrix, where p is the number of variants and K is the number of studies. Variants not observed in the study should be coded as NA.
fit.prelim	The output of function "GhostKnockoff.prelim".

Value

cor.study	The correlation among studies.

Examples

```
# We use genetic data as an example
library(GhostKnockoff)

# load example vcf file from package "seqminer", this serves as the reference panel
vcf.filename = system.file("vcf/1000g.phase1.20110521.CFH.var.anno.vcf.gz", package = "seqminer")

## this is how the actual genotype matrix from package "seqminer" looks like
example.G <- t(readVCFToMatrixByRange(vcf.filename, "1:196621007-196716634",annoType='')[[1]])
example.G <- example.G[,apply(example.G,2,sd)!=0]
example.G <- example.G[,1:100]
```

```
# compute correlation among variants
cor.G<-matrix(as.numeric(corpcor::cor.shrink(example.G)), nrow=ncol(example.G))

# fit null model
fit.prelim<-GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)

# compute study correlation
Zscore_0<-cbind(rnorm(nrow(cor.G)),rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
cor.study<-GhostKnockoff.GetCorStudy(Zscore_0,fit.prelim)
```

`GhostKnockoff.prelim` *Preliminary data management for GhostKnockoff*

Description

This function does the preliminary data management and pre-computes matrices for GhostKnockoff inference, given pre-computed correlation matrix of the variants (e.g. using reference panel in genetic studies). The output will be passed to the other functions.

Usage

```
GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)
```

Arguments

`cor.G`	The pre-computed correlation matrix of the variants.
`M`	Hypothetical number of knockoffs per variant. The default is 5 for multiple knockoff inference.
`method`	Either "sdp" or "asdp" (default: "asdp"). This determines the method that will be used to minimize the correlation between the original variables and the knockoffs.
`max.size`	The maximum number in each cluster in the "asdp" method. The default is 500. It will be ignored for "sdp".

Value

It returns a list that will be passed to function GhostKnockoff.fit().

Examples

```
# We use genetic data as an example
library(GhostKnockoff)

# load example vcf file from package "seqminer", this serves as the reference panel
vcf.filename = system.file("vcf/1000g.phase1.20110521.CFH.var.anno.vcf.gz", package = "seqminer")
```

```
## this is how the actual genotype matrix from package "seqminer" looks like
example.G <- t(readVCFToMatrixByRange(vcf.filename, "1:196621007-196716634",annoType='')[[1]])
example.G <- example.G[,apply(example.G,2,sd)!=0]
example.G <- example.G[,1:100]

# compute correlation among variants
cor.G<-matrix(as.numeric(corpcor::cor.shrink(example.G)), nrow=ncol(example.G))

# fit null model
GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)
```

`GhostKnockoff.prelim.Meta` *Additional preliminary data management for GhostKnockoff if multiple studies are involved*

Description

This function compute study dependency gamma and the optimal weights to combine multiple studies.

Usage

```
GhostKnockoff.prelim.Meta(cor.study, n.study)
```

Arguments

`cor.study`	The correlation among studies.
`n.study`	A vector of length K, where each element is the study sample size.

Value

`w_opt`	Optimal weights to combine multiple studies.
`gamma`	study dependency.

Examples

```
# We use genetic data as an example
library(GhostKnockoff)

# load example vcf file from package "seqminer", this serves as the reference panel
vcf.filename = system.file("vcf/1000g.phase1.20110521.CFH.var.anno.vcf.gz", package = "seqminer")

## this is how the actual genotype matrix from package "seqminer" looks like
example.G <- t(readVCFToMatrixByRange(vcf.filename, "1:196621007-196716634",annoType='')[[1]])
example.G <- example.G[,apply(example.G,2,sd)!=0]
example.G <- example.G[,1:100]
```

```
# compute correlation among variants
cor.G<-matrix(as.numeric(corpcor::cor.shrink(example.G)), nrow=ncol(example.G))

# fit null model
fit.prelim<-GhostKnockoff.prelim(cor.G,M=5,method='asdp',max.size=500)

# compute study correlation
Zscore_0<-cbind(rnorm(nrow(cor.G)),rnorm(nrow(cor.G))) # hypothetical Z-scores
Zscore_0<-Zscore_0+rbinom(nrow(cor.G),size=2,0.1) # set causal
cor.study<-GhostKnockoff.GetCorStudy(Zscore_0,fit.prelim)

# compute optimal weights and study dependency
n.study<-c(5000,7500)
Meta.prelim<-GhostKnockoff.prelim.Meta(cor.study, n.study)
```

Index

∗ **Feature selection**
GhostKnockoff.filter, 2
∗ **Knockoff inference**
GhostKnockoff.fit, 3
∗ **Meta analysis**
GhostKnockoff.prelim.Meta, 7
∗ **Study correlation**
GhostKnockoff.GetCorStudy, 5
∗ **preliminary work**
GhostKnockoff.prelim, 6

GhostKnockoff.filter, 2
GhostKnockoff.fit, 3
GhostKnockoff.GetCorStudy, 5
GhostKnockoff.prelim, 6
GhostKnockoff.prelim.Meta, 7