

Package ‘GWASinlps’

July 19, 2018

Type Package

Title Nonlocal Prior Based Iterative SNP Selection Tool for
Genome-Wide Association Studies

Version 1.2

Date 2018-07-19

Author Nilotpal Sanyal [aut, cre]

Maintainer Nilotpal Sanyal <nilotpal.sanyal@gmail.com>

Depends mombf, speedglm

Imports horseshoe

Suggests glmnet

Description

Performs variable selection with data from Genome-wide association studies (GWAS) combining, in an iterative variable selection framework, the computational efficiency of the screen-and-select approach based on some association learning and the parsimonious uncertainty quantification provided by the use of nonlocal priors, as described in Sanyal et al. (2018).

License GPL (>= 2)

URL <https://www.r-project.org>

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-19 20:40:20 UTC

R topics documented:

GWASinlps-package	2
GWASinlps	2
nlpLM	6

Index	9
--------------	----------

GWASinlps-package *Nonlocal prior based iterative SNP selection tool for genome-wide association study data*

Description

Performs variable selection with data from Genome-wide association studies (GWAS) combining, in an iterative variable selection framework, the computational efficiency of the screen-and-select approach based on some association learning and the parsimonious uncertainty quantification provided by the use of nonlocal priors, as described in Sanyal et al. (2018).

Details

Package: GWASinlps
Type: Package
Version: 1.1
Date: 2017-07-06
License: GPL (>= 2)

Author(s)

Nilotpal Sanyal <nilotpal.sanyal@gmail.com>

Maintainer: Nilotpal Sanyal <nilotpal.sanyal@gmail.com>

References

Sanyal et al. (2018), "GWASinlps: Nonlocal prior based iterative SNP selection tool for genome-wide association studies".

GWASinlps *Nonlocal prior based iterative SNP selection for Genome-wide association study data*

Description

Performs variable selection with data from Genome-wide association studies (GWAS) combining in an iterative variable selection framework, the computational efficiency of the screen-and-select approach based on some association learning and the parsimonious uncertainty quantification provided by the use of nonlocal priors, as described in Sanyal et al. (2018).

Usage

```
GWASinlps(y, x, family="normal", mmle=NULL, prior, tau, priorDelta = modelbbprior(1,1),
          k0, m, rxx, nskip = 3, niter = 2000, verbose = F, skip.return = F,
          seed = NULL, tau.hs.method = "halfCauchy", sigma.hs.method = "Jeffreys")
```

Arguments

y	The vector of the continuous or binary phenotype values.
x	The SNP genotype matrix with subjects represented by rows and SNPs represented by columns. The elements of x should be of numeric type. Missing values are currently not accepted.
family	"normal" for continuous data (default), "binomial" for binary data.
mmle	Optional, and only relevant if family=="binomial". If mmle is not input, it will be computed. Useful to input for the sake of time if running the function multiple times.
prior	"mom" for pMOM prior, "imom" for piMOM prior, "zellner" for Zellner's g-prior, "horseshoe" for horseshoe prior. When family=="binomial", only "pMOM" is available.
tau	the value of the scale parameter tau of the nonlocal prior.
priorDelta	Prior for model space. Defaults to modelbbprior(1,1), which is beta-binomial(1,1) prior.
k0	GWASinlps tuning parameter, denoting the number of leading SNPs (see References).
m	GWASinlps tuning parameter, denoting the maximum number of SNPs to be selected.
rxx	GWASinlps tuning parameter, denoting the correlation threshold to determine leading sets (see References)
nskip	GWASinlps tuning parameter, denoting the maximum allowed count of skipping an iteration selecting no SNPs (see References).
niter	Number of MCMC iterations for nonlocal prior based Bayesian variable selection. Defaults to 2000.
verbose	If TRUE, prints some details. FALSE by default.
skip.return	False by default. If True, returns the set of selected variables after each skipping, else returns the final set of selected variables.
seed	Optional. If supplied, the random seed is set to this value at the beginning for reproducibility.
tau.hs.method	Optional. Necessary only when prior=="horseshoe".
sigma.hs.method	Optional. Necessary only when prior=="horseshoe".

Details

The GWASinlps method selects SNPs iteratively. For continuous response (phenotype), the procedure starts with an initial set of SNPs, a SNP genotype matrix x and a phenotype vector y . An iteration proceeds by determining the k_0 *leading SNPs* having the highest association with y . The measure of association is absolute value of the Pearson's correlation coefficient for continuous phenotype. These k_0 leading SNPs, in turn, determine k_0 *leading sets*, where each leading set consists of all SNPs with absolute correlation coefficient more than or equal to r_{xx} with the corresponding leading SNP. Then within each leading set, non-local prior based Bayesian variable selection is run (using package `mombf`) and the predictors appearing in the HPPM are considered selected in the current iteration. Thus, a single SNP can be selected from multiple leading sets. The selected predictors are regressed out from y . The predictors which are included in one or more *leading sets* but do not appear in any HPPM are dropped from further analysis. With updated y and SNP set, next iteration follows similarly. The procedure continues until the stopping point, determined by the GWASinlps tuning parameters m , r_{xx} , and n_{skip} , is reached. For more details, see the References.)

For binary response (phenotype), the procedure starts with an initial set of SNPs, a SNP genotype matrix x and a phenotype vector y . Unlike the continuous phenotype case, here, before proceeding into iterative selection, a pre-iteration step is conducted. In the pre-iteration step, we determine k_0_aux SNPs having the highest association with y when the measure of association is the absolute value of the maximum marginal likelihood coefficient, run non-local prior based Bayesian variable selection for binary phenotype (using package `mombf`) with those k_0_aux SNPs, and select those SNPs whose marginal posterior probability of inclusion is more than 0.5. If none of those k_0_aux SNPs is selected, we consider the next k_0_aux SNPs association-wise, and do the same. This process is repeated until at least one SNP is selected. The pre-iteration step ends when at least one SNP is selected. Then, we update y by the deviance residuals obtained from a glm regression of y on the SNPs selected in the pre-iteration step. These deviance residuals are continuous. With these updated continuous y , we conduct GWASinlps based variable selection for continuous phenotypes as mentioned before. The final selection of SNPs for the binary response case is the SNPs selected in pre-iteration step plus the SNPs selected based on deviance residuals. Currently, k_0_aux is set as 1.

Value

If `skip.return == F` (default), then

`selected` The vector of GWASinlps selected predictors, in the iterational order they were selected.

If `skip.return == T`, then a list

`selected` A list, whose length is equal to the number of times skipping took place, and whose elements are the vectors of selected predictors, in the iterational order they were selected, after every skipping.

Author(s)

Nilotpal Sanyal <nilotpal.sanyal@gmail.com>

References

Sanyal et al. (2018), "GWASinlps: Nonlocal prior based iterative SNP selection tool for genome-wide association studies".

Examples

```

p = 1000; n = 200
m = 10

# Generate SNP genotype matrix
set.seed(1)
f = runif( p, .1, .2 ) # simulate minor allele frequency
x = matrix( nrow = n, ncol = p )
colnames(x) = 1:p
for(j in 1:p)
  x[,j] = rbinom( n, 2, f[j] )

# Generate data
causal_snps = sample( 1:p, m )
beta = rep( 0, p )
set.seed(1)
beta[causal_snps] = rnorm(m, mean = 0, sd = 2 )
y = x %*% beta + rnorm(n, 0, 1)

# Fix scale parameter tau
tau = 0.2

# Perform GWASinlps
inlps = GWASinlps(y, x, prior = "mom", tau = tau, k0 = 1, m = 50,
  rxx = .2)
cat( "GWASinlps selected", length(inlps), "SNPs with", length(intersect(inlps, causal_snps)),
  "true positives." )

# Compare with LASSO
library(glmnet)
fit.cvlasso = cv.glmnet( x, y, alpha = 1 )
l.min = fit.cvlasso $lambda.min # lambda that gives minimum cvm
lasso_min = which( as.vector( coef( fit.cvlasso, s = l.min ) )[-1] != 0 )
cat( "LASSO with lambda.min selected", length(lasso_min), "SNPs with",
  length(intersect(lasso_min, causal_snps)), "true positives." )

l.1se = fit.cvlasso $lambda.1se # largest lambda such that error is within 1 se of the minimum
lasso_1se = which( as.vector( coef( fit.cvlasso, s = l.1se ) )[-1] != 0 )
cat( "LASSO with lambda.1se selected", length(lasso_1se), "SNPs with",
  length(intersect(lasso_1se, causal_snps)), "true positives." )

```

nlpsLM	<i>Nonlocal prior based single-step SNP selection for Genome-wide association study data with continuous phenotype</i>
--------	--

Description

Performs variable selection for continuous phenotypes in a single iteration, combining the computational efficiency of the screen-and-select approach based on some association learning and the parsimonious uncertainty quantification provided by the use of nonlocal priors, as described in Sanyal et al. (2018)

Usage

```
nlpsLM(y, x, prior, tau, priorDelta = modelbbprior(1,1), k0, rxx,
       niter = 2000, verbose = F,
       tau.hs.method = "halfCauchy", sigma.hs.method = "Jeffreys" )
```

Arguments

y	The vector of the continuous phenotype values.
x	The SNP genotype matrix with subjects represented by rows and SNPs represented by columns. The elements of x should be of numeric type. Missing values are currently not accepted.
prior	"mom" for pMOM prior, "imom" for piMOM prior, "zellner" for Zellner's g-prior, "horseshoe" for horseshoe prior.
tau	the value of the scale parameter tau of the nonlocal prior.
priorDelta	Prior for model space. Defaults to modelbbprior(1,1), which is beta-binomial(1,1) prior.
k0	GWASinlps tuning parameter, denoting the number of leading SNPs (see References).
rxx	GWASinlps tuning parameter, denoting the correlation threshold to determine leading sets (see References).
niter	Number of MCMC iterations for nonlocal prior based Bayesian variable selection. Defaults to 2000.
verbose	If TRUE, prints some details. FALSE by default.
tau.hs.method	Optional. Necessary only when prior=="horseshoe".
sigma.hs.method	Optional. Necessary only when prior=="horseshoe".

Details

The nlpsLM function performs SNP selection in one iteration for continuous phenotypes. The GWASinlps function repeatedly calls the nlpsLM function. The nlpsLM procedure starts by determining the k_0 leading SNPs having the highest association with y . The measure of association is absolute value of the Pearson's correlation coefficient. These k_0 leading SNPs, in turn, determine the k_0 leading sets, where each leading set consists of all SNPs with absolute correlation coefficient more than or equal to r_{xx} with the corresponding leading SNP. Then non-local prior based Bayesian variable selection is run within each leading set (using package mombf), and the predictors appearing in the HPPM are considered selected. For more details, see the References. For horseshoe prior, package horseshoe is used.

Value

A list with elements

hppm	The set of predictors appearing in the HPPM of at least one leading set.
not.selected	The set of predictors appearing in at least one leading set, but in none of the HPPMs.

Author(s)

Nilotpal Sanyal <nilotpal.sanyal@gmail.com>

References

Sanyal et al. (2018), "GWASinlps: Nonlocal prior based iterative SNP selection tool for genome-wide association studies".

See Also

[GWASinlps](#)

Examples

```
p = 1000; n = 100
m = 10
# Generate SNP genotype matrix
set.seed(1)
f = runif( p, .1, .2 ) # simulate minor allele frequency
x = matrix( nrow = n, ncol = p )
colnames(x) = 1:p
for(j in 1:p)
  x[,j] = rbinom( n, 2, f[j] )
# Generate data
causal_snps = sample( 1:p, m )
beta = rep( 0, p )
set.seed(1)
beta[causal_snps] = rnorm(m, mean = 0, sd = 2 )
y = x %*% beta + rnorm(n, 0, 1)
```

```
# Fix scale parameter tau
tau = 0.022
# Perform GWASinlps
out = nlpLM(y, x, prior = "mom", tau = tau, k0 = 10, rxx = .5, niter = 10000, verbose = TRUE)
out
```


Index

GWASinlps, [2](#), [7](#)

GWASinlps-package, [2](#)

nlpLM, [6](#)