Package 'LBLGXE'

December 3, 2020

Type Package

Title Logistic Bayesian Lasso for Rare (or Common) Haplotype Association

Version 1.5

Author Xiaochen Yuan, Yuan Zhang, Shuang Xia, Swati Biswas, and Shili Lin

Maintainer Xiaochen Yuan <xxy142030@utdallas.edu>

Description This function takes a dataset of genotypes, at least one binary phenotype, and optionally environmental covariates. Bayesian lasso is used to find the posterior distributions of logistic regression coefficients, which are then used to calculate Bayes Factor and credible set to test for association with haplotypes, environmental covariates and interactions. The model can handle complex sampling data, in particular, frequency matched cases and controls with controls obtained using stratified sampling. This version can also be applied to a dataset with no environmental covariate and two correlated phenotypes (both can be binary or one is binary and the other one is continuous).
Yuan X, Biswas S (2019) <doi:10.1002/gepi.22258>.
Zhang Y, Hofmann J, Purdue M, Lin S, and Biswas S (2017) <doi:10.1038/jhg.2017.43>.
Zhang Y, Lin S, and Biswas S (2017) <doi:10.1111/biom.12567>.
Zhang Y and Biswas S (2015) <doi:10.4137/CIN.S17290>.
Biswas S, Xia S and Lin S (2014) <doi:10.1002/gepi.21773>.
Biswas S, Lin S (2012) <doi:10.1111/j.1541-0420.2011.01680.x>.
Burkett K, Graham J and McNeney B (2006) <doi:10.18637/jss.v016.i02>.

Depends R (>= 3.5), hapassoc

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

LBLGXE-package *Logistic Bayesian Lasso for Rare (or Common) Haplotype Association*

Description

The main function of this package is LBL. For details, see ?LBL.

Details

Package:	LBLGXE
Type:	Package
Version:	1.5
Date:	2020-11-30
License:	GPL-3
LazyLoad:	yes

Currently available functions: LBL. Type ?LBL for more details.

Author(s)

Xiaochen Yuan, Yuan Zhang, Shuang Xia, Swati Biswas, and Shili Lin

Maintainer: Xiaochen Yuan<xxy142030@utdallas.edu>

References

Yuan X and Biswas S (2020). Detecting rare haplotype association with two correlated phenotypes of binary and continuous types. Statistics in Medicine, revision submitted.

Yuan X and Biswas S (2019). Bivariate Logistic Bayesian LASSO for Detecting Rare Haplotype Association with Two Correlated Phenotypes. Genetic Epidemiology, 43(8):996-1017.

Zhang Y, Hofmann J, Purdue M, Lin S, and Biswas S. Logistic Bayesian LASSO for Genetic Association Analysis of Data from Complex Sampling Designs. Journal of Human Genetics, 62:819-829.

Zhang Y, Lin S, and Biswas S. Detecting Rare and common Haplotype-Environment Interaction under Uncertainty of Gene-Environment Independence Assumption. Biometrics, 73:344-355.

Zhang, Y. and Biswas, S (2015). An Improved Version of Logistic Bayesian LASSO for Detecting Rare Haplotype-Environment Interactions With Application to Lung Cancer, Cancer Informatics, 14(S2): 11-16.

Biswas S, Xia S and Lin S (2014). Detecting Rare Haplotype-Environment Interaction with Logistic Bayesian LASSO. Genetic Epidemiology, 38: 31-41.

Biswas S and Lin S (2012). Logistic Bayesian LASSO for Identifying Association with Rare Haplotypes and Application to Age-related Macular Degeneration. Biometrics, 68(2): 587-97.

Burkett K, Graham J and McNeney B (2006). hapassoc: Software for Likelihood Inference of Trait Associations with SNP Haplotypes and Other Attributes. Journal of Statistical Software, 16(2): 1-19.

Examples

```
#see ?LBL
```

LBL *Logistic Bayesian Lasso for Rare (or Common) Haplotype Association*

Description

Bayesian LASSO is used to find the posterior distributions of logistic regression coefficients, which are then used to calculate Bayes Factor and credible set to test for association with haplotypes, environmental covariates, and interactions. It can handle complex sampling data, in particular, frequency matched cases and controls with controls obtained using stratified sampling. This version can also be applied to a dataset with no environmental covariate and two correlated phenotypes. The function first calls pre.hapassoc function from the hapassoc package, and some of the options such as "dat", "numSNPs", "maxMissingGenos" and "allelic" are used by pre.hapassoc. It takes as an argument a dataframe with non-SNP and SNP data. The rows of the input data frame should correspond to subjects. Missing single-locus genotypes, up to a maximum of maxMissingGenos (see below), are allowed, but subjects with missing data in more than maxMissingGenos, or with missing non-SNP data, are removed.

Usage

```
LBL(
  dat,
  numSNPs,
  maxMissingGenos = 1,
  allelic = TRUE,
  haplo.baseline = "missing",
  cov.baseline = "missing",
  complex.sampling = FALSE,
  n.stra = NULL,
  interaction.stra = TRUE,
  interaction.env = TRUE,
  interaction.model = "i",
  names.dep = "missing",
  a = 20,
  b = 20,
  start.beta = 0.01,
  gamma = 0.01,
  lambda = 1,
  D = 0,
  e = 0.1,
  seed = NULL,
  burn.in = NULL,
  num.it = NULL,
  twoPheno = "None",
  start.u = 0.01,
  sigma_sq_u = 1,
  sigma_sq_c = NULL,
  start.f00 = NULL,
```

```
  start.f10 = NULL,
  start.f01 = NULL,
  e_allHap = 0.4,
  print.freq.ci = FALSE,
  print.lambda.ci = FALSE,
  print.D.ci = FALSE,
  print.sigma_sq_u.ci = FALSE,
  print.sigma_sq_c.ci = FALSE
)
```

Arguments

dat	the non-SNP and SNP data as a data frame. If the twoPheno option is "None" (default) and the complex.sampling option is FALSE (default), the first column of the non-SNP data is the affection status, others (optional) are environmental covariates; if the complex.sampling option is set to be TRUE, the non-SNP data should consists of affection status, sampling weights, stratifying variables and environmental covariates (optional). If the twoPheno option is set to be "2B", then there should be no environmental covariate and the first two columns should be two binary phenotypes. If the twoPheno option is set to be "BC", then there should be no environmental covariate and the first column should be a binary phenotype while the second column be a continous phenotype. SNP data should comprise the last 2*numSNPs columns (allelic format) or last numSNPs columns (genotypic format). Missing allelic data should be coded as NA or "" and missing genotypic data should be coded as, e.g., "A" if one allele is missing and "" if both alleles are missing. Covariates (including stratifying variables) should be coded as dummy variables, e.g., 0, 1, etc.
numSNPs	number of SNPs per haplotype.
maxMissingGenos	maximum number of single-locus genotypes with missing data to allow for each subject. (Subjects with more missing data, or with missing non-SNP data are removed.) The default is 1.
allelic	TRUE if single-locus SNP genotypes are in allelic format and FALSE if in genotypic format; default is TRUE.
haplo.baseline	haplotype to be used for baseline coding; default is the most frequent haplotype according to the initial haplotype frequency estimates returned by pre.hapassoc.
cov.baseline	Needed only if the non-SNP data contains stratifying variables or environmental covariates. Indicates the baseline level(s) for the covariates (including stratifying variables). Note that they should be listed in the same order as in the actual data. The default is the level(s) that is coded as 0 for each covariate. This option is ignored if twoPheno = "2B" or "BC".
complex.sampling	whether complex sampling with frequency matching will be used; default is FALSE. Specifically, when this option is set to be TRUE, G-E and/or G-S dependence is assumed, which needs to be further specified by the names.dep option. This option is ignored if twoPheno = "2B" or "BC".
n.stra	Needed only if the complex.sampling option is set to be TRUE. Indicates number of stratifying variables.
interaction.stra	Needed only if the complex.sampling option is set to be TRUE. Indicates whether or not to model interaction between haplotypes and stratifying variables in the model; default is TRUE. This option is ignored if twoPheno = "2B" or "BC".

`interaction.env`
: Needed only if the non-SNP data contains environmental covariates. Indicates whether or not to model interaction between haplotypes and environmental covariates in the model; default is TRUE. This option is ignored if twoPheno = "2B" or "BC".

`interaction.model`
: Needed only if the complex.sampling option is set to be FALSE and the interaction.cov option is set to be TRUE. Indicates whether G-E independence is assumed or not for fitting haplotype-environment interactions. "i" represents G-E independent model, "d" represents G-E dependent model, and "u" represents uncertainty about G-E independence, i.e., allows possibility of both models. The default is "i". This option is ignored if twoPheno = "2B" or "BC".

`names.dep`
: Needed only if the complex.sampling option is set to be TRUE or interaction.model option is set to be "d" or "u". Indicates the covariates that are believed to cause G-E dependence. The default is a vector consisting of all covariates, however, if the number of covariates is large, then this will lead to a very large and complicated G-E dependence model so a judicious choice of covariates for this model is recommended in that case.

`a`
: first hyperparameter of the prior for regression coefficients, beta. The prior variance of beta is 2/lambda^2 and lambda has Gamma(a,b) prior. The Gamma parameters a and b are such that the mean and variance of the Gamma distribution are a/b and a/b^2. The default is 20.

`b`
: b parameter of the Gamma(a,b) distribution described above; default is 20.

`start.beta`
: starting value of all regression coefficients, beta; default is 0.01.

`gamma`
: starting value of the gamma parameters (slopes), which are used to model G-E dependence through a multinomial logistic regression model; default is 0.01. This option is ignored if twoPheno = "2B" or "BC".

`lambda`
: starting value of the lambda parameter described above; when twoPheno="BC", it is the starting value of the lambda_b and lambda_c parameters; default is 1.

`D`
: starting value of the D parameter, which is the within-population inbreeding coefficient; default is 0.

`e`
: a (small) number epsilon in the null hypothesis of no association, H0: |beta| <= epsilon. Changing e from default of 0.1 may need choosing a different threshold for Bayes Factor (one of the outputs) to infer association. The default is 0.1.

`seed`
: the seed to be used for the MCMC in Bayesian Lasso; default is a random seed. If exactly same results need to be reproduced, seed should be fixed to the same number.

`burn.in`
: burn-in period of the MCMC sampling scheme; default is 20000 for model with a single univariate phenotype and 50000 for model with two binary phenotypes.

`num.it`
: total number of MCMC iterations including burn-in. When the complex.sampling option is set to be FALSE, default is 50000 if there are no covariates or interaction.model = "i"; default values are 70000 and 100000, respectively, if interaction.model = "d" and "u". When the complex.sampling option is set to be TRUE, the default value of num.it is 120000. When the twoBinaryPheno option is set to be TRUE, the default value of num.it is 200000.

`twoPheno`
: has three options: "2B" means two binary correlated phenotypes will be used (no environmental covariate allowed), "BC" means one binary and one continuous correlated phenotypes will be used (no environmental covariate allowed), and "None" means bivariate methods will not be used; when being "2B" or "BC",

the options of "complex.sampling", "interaction.stra", "interaction.env", and "interaction.model" will be ignored; default is "None".

`start.u` Needed only if twoPheno="2B" or "BC". Starting value of u (subject-specific latent variables); ui induces correlation between the two phenotypes of i-th individual; default is 0.01.

`sigma_sq_u` Needed only if twoPheno="2B" or "BC" or BiBC=TRUE. Starting value of sigma_sq_u parameter, which is the variance of u elements; ui is assumed to follow N(0, sigma_sq_u) distribution; default is 1.

`sigma_sq_c` Needed only if twoPheno="BC". Starting value of sigma_sq_c parameter, which is the varance of the error term in the continous phenotype; default is NULL.

`start.f00` Needed only if twoPheno="2B". Starting value of the f00 parameter vector, which consists of haplotype frequencies in the population of controls for both phenotypes; if it set to be None (default), the initFreq returned by pre.hapassoc when applied to the corresponding sample will be used.

`start.f10` Needed only if twoPheno="2B". Starting value of the f10 parameter vector, which consists of haplotype frequencies in the population of cases for the first phenotype and controls for the second phenotype; if it set to be None (default), the initFreq returned by pre.hapassoc when applied to the corresponding sample will be used.

`start.f01` Needed only if twoPheno="2B". Starting value of the f01 parameter vector, which consists of haplotype frequencies in the population of controls for the first phenotype and cases for the second phenotype; if it set to be None (default), the initFreq returned by pre.hapassoc when applied to the corresponding sample will be used.

`e_allHap` Needed only if twoPheno="2B" or "BC". Epsilon in the null hypothesis for testing all haplotypes together in a block, H0: |beta| <= epsilon for all beta coefficients corresponding to all haplotypes in a block and both diseases. The default is 0.4.

`print.freq.ci` Needed only if twoPheno="2B" or "BC". Whether the 95% credible sets for f00, f10, and f01 are to be printed. The default is FALSE.

`print.lambda.ci`
Needed only if twoPheno="2B" or "BC". Whether the 95% credible set for lambda is to be printed. The default is FALSE.

`print.D.ci` Needed only if twoPheno="2B" or "BC". Whether the 95% credible set for D is to be printed. The default is FALSE.

`print.sigma_sq_u.ci`
Needed only if twoPheno="2B" or "BC". Whether the 95% credible set for sigma_sq_u (Bivariate LBL-2B) or sigma_u (Bivariate LBL-BC) is to be printed. The default is FALSE.

`print.sigma_sq_c.ci`
Needed only if twoPheno="BC". Whether the 95% credible set for sigma_sq_c is to be printed. The default is FALSE.

Value

`BF` For single phenotype. A vector of Bayes Factors for all regression coefficients. If BF exceeds a certain threshold (e.g., 2 or 3) association may be concluded.

`OR` For single phenotype. A vector of estimated odds ratios of the corresponding haplotype against the reference haplotype (haplo.baseline). This is the exponential of the posterior means of the regression coefficients.

`CI.OR` For single phenotype. 95% credible sets for the ORs. If CI.OR excludes 1, association may be concluded.

`freq` For single phenotype or twoPheno = "BC". A vector of posterior means of the haplotype frequencies.

`CI.freq` For single phenotype or twoPheno = "BC". 95% credible sets for each haplotype frequency. It is optional when twoPheno = "BC", only shown if print.freq.ci=TRUE.

`percentage.indep` For single phenotype. Available only if the interaction.model option is set to be "u". Percentage of iterations in which independent model is choosen.

`percentage.dep` For single phenotype. Available only if the interaction.model option is set to be "u". Percentage of iterations in which dependent model is chosen.

`CI.gamma` For single phenotype. Available only if the interaction.model option is set to be "d" or "u". 95% credible sets for the gamma parameters as described above.

`CI.lambda` For single phenotype or twoPheno = "2B". 95% credible sets for the lambda parameter as described above. When twoPheno = "2B", it is optional, only shown if print.lambda.ci=TRUE.

`CI.lambda_b` For twoPheno = "BC". 95% credible sets for the lambda_b parameter. It is optional, only shown if print.lambda.ci=TRUE.

`CI.lambda_c` For twoPheno = "BC". 95% credible sets for the lambda_c parameter. It is optional, only shown if print.lambda.ci=TRUE.

`CI.D` 95% credible sets for D as described above. When twoPheno = "2B" or "BC", it is optional, only shown if print.D.ci=TRUE.

`BF_bivariate_hap` For twoPheno = "2B" or "BC". A vector of Bayes Factors for testing association of each haplotype with both phenotypes jointly. If a BF exceeds a certain threshold, the corresponding haplotype may be associated with at least one of the two phenotypes.

`BF_bivariate_allHap` For twoPheno = "2B" or "BC". The joint Bayes Factor for testing association of all haplotypes in a block together with both phenotypes jointly. If joint BF exceeds a certain threshold, then at least one of the haplotypes may be associated with at least one of the two phenotypes.

`beta1` For twoPheno = "2B". A vector of estimated posterior means of the regression coefficients for the first phenotype.

`beta2` For twoPheno = "2B". A vector of estimated posterior means of the regression coefficients for the second phenotype.

`beta_b` For twoPheno = "BC". A vector of estimated posterior means of the regression coefficients for the binary phenotype.

`beta_c` For twoPheno = "BC". A vector of estimated posterior means of the regression coefficients for the continous phenotype.

`CI.beta1` For twoPheno = "2B". 95% credible sets for the beta1s. These are based on marginal distribution of each beta1 coefficient and should not be used for inference about association with the two phenotypes jointly.

`CI.beta2` For twoPheno = "2B". 95% credible sets for the beta2s. These are based on marginal distribution of each beta2 coefficient and should not be used for inference about association with the two phenotypes jointly.

`CI.beta_b` For twoPheno = "BC". 95% credible sets for the beta_b's. These are based on marginal distribution of each beta_b coefficient and should not be used for inference about association with the two phenotypes jointly.

`CI.beta_c` For twoPheno = "BC". 95% credible sets for the beta_c's. These are based on marginal distribution of each beta_c coefficient and should not be used for inference about association with the two phenotypes jointly.

`freq00` For twoPheno = "2B". A vector of posterior means of the haplotype frequencies in the population of controls for both phenotypes.

`freq10` For twoPheno = "2B". A vector of posterior means of the haplotype frequencies in the population of cases for the first phenotype and controls for the second phenotype.

`freq01` For twoPheno = "2B". A vector of posterior means of the haplotype frequencies in the population of controls for the first phenotypes and cases for the second phenotype.

`CI.freq00` For twoPheno = "2B". 95% credible sets for each haplotype frequency in the population of controls for both phenotypes. Optional, only shown if print.freq.ci=TRUE.

`CI.freq10` For twoPheno = "2B". 95% credible sets for each haplotype frequency in the population of cases for the first phenotype and controls for the second phenotype. Optional, only shown if print.freq.ci=TRUE.

`CI.freq01` For twoPheno = "2B". 95% credible sets for each haplotype frequency in the population of controls for the first phenotypes and cases for the second phenotype. Optional, only shown if print.freq.ci=TRUE.

`CI.sigma_sq_u` For twoPheno = "2B". 95% credible sets for sigma_sq_u as described above. Optional, only shown if print.sigma_sq_u.ci=TRUE.

`CI.sigma_u` For twoPheno = "BC". 95% credible sets for sigma_u, which is the square root of sigma_sq_u as described above. Optional, only shown if print.sigma_sq_u.ci=TRUE.

`CI.sigma_sq_c` For twoPheno = "BC". 95% credible sets for sigma_sq_c as described above. Optional, only shown if print.sigma_sq_c.ci=TRUE.

Author(s)

Xiaochen Yuan, Yuan Zhang, Shuang Xia, Swati Biswas, Shili Lin

References

Yuan X and Biswas S (2020). Detecting rare haplotype association with two correlated phenotypes of binary and continuous types. Statistics in Medicine, revision submitted.

Yuan X and Biswas S (2019). Bivariate Logistic Bayesian LASSO for Detecting Rare Haplotype Association with Two Correlated Phenotypes. Genetic Epidemiology, 43(8):996-1017.

Zhang Y, Hofmann J, Purdue M, Lin S, and Biswas S. Logistic Bayesian LASSO for Genetic Association Analysis of Data from Complex Sampling Designs. Journal of Human Genetics, 62:819-829.

Zhang Y, Lin S, and Biswas S. Detecting Rare and common Haplotype-Environment Interaction under Uncertainty of Gene-Environment Independence Assumption. Biometrics, 73:344-355.

Zhang, Y. and Biswas, S (2015). An Improved Version of Logistic Bayesian LASSO for Detecting Rare Haplotype-Environment Interactions With Application to Lung Cancer, Cancer Informatics, 14(S2): 11-16.

Biswas S, Xia S and Lin S (2014). Detecting Rare Haplotype-Environment Interaction with Logistic Bayesian LASSO. Genetic Epidemiology, 38: 31-41.

Biswas S, Lin S (2012). Logistic Bayesian LASSO for Identifying Association with Rare Haplotypes and Application to Age-related Macular Degeneration. Biometrics, 68(2): 587-97.

Burkett K, Graham J and McNeney B (2006). hapassoc: Software for Likelihood Inference of Trait Associations with SNP Haplotypes and Other Attributes. Journal of Statistical Software, 16(2): 1-19.

See Also

pre.hapassoc

Examples

```
# Load example datasets.
# This dataset consists of affection status, a binary environmental covariate, and SNP data.
data(LBL.ex1)
# This dataset consists of affection status, complex sampling weights, a binary stratifying
# variable, a binary environmental covariate, and SNP data.
data(LBL.ex2)
# This dataset consists of two correlated affection statuses, no environmental covariate,
#and SNP data.
data(LBL.ex3)
# This dataset consists of one binary and one continuous correlated affection statuses,
#no environmental covariate, and SNP data.
data(LBL.ex4)
# Install hapassoc package.
library(hapassoc)
# Run LBL to make inference on haplotype associations and interactions. Note the default
# setting for burn.in and num.it are larger in the LBL function. However, you may want to
# use smaller numbers for a quick check to make sure the package is loaded properly. With
# such shorts runs, the results may not be meaningful.
## Analyzing LBL.ex1 under G-E independence assumption.
out.LBL<-LBL(LBL.ex1, numSNPs=5, burn.in=0, num.it=5)

## Analyzing LBL.ex1 under uncertainty of G-E independence assumption.
out.LBL<-LBL(LBL.ex1, numSNPs=5, interaction.model="u", burn.in=0, num.it=5)

## Analyzing LBL.ex2 which comes from complex sampling design with frequency matching.
out.LBL<-LBL(LBL.ex2, numSNPs=5, complex.sampling=TRUE, n.stra=1, names.dep="stra",
burn.in=0, num.it=5)

## Analyzing LBL.ex3 using the bivariate LBL-2B method.
out.LBL<-LBL(LBL.ex3, numSNPs=5, twoPheno="2B", burn.in=0, num.it=5)

## Analyzing LBL.ex4 using the bivariate LBL-BC method.
out.LBL<-LBL(LBL.ex4, numSNPs=5, twoPheno="BC", burn.in=0, num.it=5)
```

LBL.ex1 *This dataset consists of affection status, a binary environmental covariate, and SNP data.*

Description

This dataset consists of affection status, a binary environmental covariate, and SNP data.

Usage

```
LBL.ex1
```

Format

A data frame with variables: affected, cov, M1.1, M1.2, M2.1, M2.2, M3.1, M3.2, M4.1, M4.2, M5.1, M5.2.

LBL.ex2 — *This dataset consists of affection status, complex sampling weights, a binary stratifying variable, a binary environmental covariate, and SNP data.*

Description

This dataset consists of affection status, complex sampling weights, a binary stratifying variable, a binary environmental covariate, and SNP data.

Usage

```
LBL.ex2
```

Format

A data frame with variables: affected, wt, stra, cov, M1.1, M1.2, M2.1, M2.2, M3.1, M3.2, M4.1, M4.2, M5.1, M5.2.

LBL.ex3 — *This dataset consists of two correlated affection statuses, no environmental covariate, and SNP data.*

Description

This dataset consists of two correlated affection statuses, no environmental covariate, and SNP data.

Usage

```
LBL.ex3
```

Format

A data frame with variables: Y1, Y2, M1.1, M1.2, M2.1, M2.2, M3.1, M3.2, M4.1, M4.2, M5.1, M5.2.

LBL.ex4 *This dataset consists of one binary and one continuous traits, no environmental covariate, and SNP data.*

Description

This dataset consists of one binary and one continuous traits, no environmental covariate, and SNP data.

Usage

```
LBL.ex4
```

Format

A data frame with variables: Yb, Yc, M1.1, M1.2, M2.1, M2.2, M3.1, M3.2, M4.1, M4.2, M5.1, M5.2.

Index