Package 'LBLGXE'

April 15, 2016

Type Package

Title Bayesian Lasso for detecting Rare (or Common) Haplotype Association and their interactions with Environmental Covariates

Version 1.3

Date 2016-04-15

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

Maintainer Yuan Zhang <yxz112020@utdallas.edu>

Description This function takes a dataset of haplotypes and environmental covariates in which rows for individuals of uncertain phase have been augmented by ``pseudo-individuals" who carry the possible multilocus genotypes consistent with the single-locus phenotypes. 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. This version can also handle complex sampling data, in particular, frequency matched cases and controls with controls obtained using stratified sampling.

License GPL-3

LazyLoad yes

Archs x64

R topics documented:

`LBLGXE-package` *Bayesian Lasso for detecting Rare (or Common) Haplotype Association and their interactions with Environmental Covariates*

Description

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

Details

Package:	LBLGXE
Type:	Package
Version:	1.3
Date:	2016-04-15
License:	GPL-3
LazyLoad:	yes

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

Author(s)

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

Maintainer: Yuan Zhang <yxz112020@utdallas.edu>

References

Zhang Y, Hofmann J, Purdue M, Lin S, and Biswas S. Logistic Bayesian LASSO for Genetic Association Analysis of Data from Complex Sampling Designs. Manuscript.

Zhang Y, Lin S, and Biswas S. Detecting Rare Haplotype-Environment Interaction under Uncertainty of Gene-Environment Independence Assumption. Under review.

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.

See Also

<hapassoc> <pre.hapassoc>

Examples

```
#see ?LBL
```

LBL *Bayesian Lasso for detecting Rare (or Common) Haplotype Association and their interactions with Environmental Covariates*

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. This version can also handle complex sampling data, in particular, frequency matched cases and controls with controls obtained using stratified sampling. 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)
```

Arguments

`dat`	the non-SNP and SNP data as a data frame. If the complex.sampling option is set to be 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). The 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 levels 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 levels that are coded as 0 for each covariate.
`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.

`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.

`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.

`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".

`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.

`lambda` starting value of the lambda parameter described above; default is 1.

`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.

`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.

`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.

Value

BF	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	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	95% credible sets for the ORs. If CI.OR excludes 1, association may be concluded.
freq	A vector of posterior means of the haplotype frequencies.
CI.freq	95% credible sets for each haplotype frequency.
percentage.indep	Available only if the interaction.model option is set to be "u". Percentage of iterations in which independent model is choosen.
percentage.dep	Available only if the interaction.model option is set to be "u". Percentage of iterations in which dependent model is chosen.
CI.gamma	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	95% credible sets for the lambda parameter as described above.
CI.D	95% credible sets for D as described above.

Author(s)

Yuan Zhang, Shuang Xia, Swati Biswas, Shili Lin

References

Zhang Y, Hofmann J, Purdue M, Lin S, and Biswas S. Logistic Bayesian LASSO for Genetic Association Analysis of Data from Complex Sampling Designs. Manuscript.

Zhang Y, Lin S, and Biswas S. Detecting Rare Haplotype-Environment Interaction under Uncertainty of Gene-Environment Independence Assumption. Under review.

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', 'hapassoc', 'rGLM'

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)

# Install hapassoc and dummies package
library(hapassoc)
library(dummies)

# 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=100, num.it=1000)

## Analyzing LBL.ex1 under uncertainty of G-E independence assumption.
out.LBL<-LBL(LBL.ex1, numSNPs=5, interaction.model="u", burn.in=100, num.it=1000)

## 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=100, num.it=1000)
```

Index