LBL: Logistic and Quantitative Bayesian LASSO for Detecting Rare Haplotype Associations

Contents

1 Introduction

LBL (Logistic and Quantitative Bayesian LASSO) (Biswas and Lin (2012), Wang and Lin (2014), Zhou, Wang, and Lin (2019)) is a Bayesian genetic association test aimed at detecting association between rare *haplotypes* (which could be formed by common SNPs) and diseases. There are currently three software that handle different types of study designs for binary traits, and one software that is designed for quantitative traits while addressing for population stratification.

The three software that handle different types of study designs are: one for independent case-control data (LBL, Biswas and Lin (2012)), one for case-parent triad (family trio) data (famLBL, Wang and Lin (2014)) and one for a combination of data from the two designs (cLBL, Zhou, Wang, and Lin (2019)). The software designed for quantitative traits uses principal components (PCs) to adjust for population stratification.

LBLs take genotype and phenotype data as input, and provide statistical inferences of the effect of each haplotype on the phenotype based on the Markov Chain Monte Carlo samples from the posterior distribution.

The rest of the vignette is structured as follows:

- Methods section provides some technical details about the algorithm. Users can skip this section should they choose to.
- Using LBL section provides some detailed explanation of how to use the package.
- Example section has a step-by-step guide of applying LBL to a simulated dataset.

2 Methods

In this section, we provide a short description of the LBL methodology formulation. For binary traits, the likelihood portions are formulated with retrsopectively likelihoods, and are connected with disease model via a logistic link. For quantitative traits, we consider the prospective likelihood. The priors of LBL methods include a double exponential distribution on the haplotyic effect, penalizing the coefficients of non-effective haplotypes, so that effective haplotypes (such as the rare haplotypes with large size) will stand out in the association analysis. Monte Carlo Markov Chain algorithm (MCMC, Metropolis et al. (1953), Hastings (1970), and Geman and Geman (1984)) is used to sample from the posterior distribution.

In the following, we first discuss the likelihoods for all LBL methods, and then the priors, computation, and the inferences based on posterior samples. The likelihood formulations of LBLs share some similarities. All possible haplotypes compatible with observed genotypes are obtained from hapassoc. The priors for the parameters are the same for all LBL methods. The posterior samples of each LBL can be obtained via MCMC, and posterior inferences can be carried out once the chain has converged. For a more detailed discussion of each method, see the corresponding papers.

2.1 Likelihood

2.1.1 LBL

Consider a case-control design with n total individuals (n_1 cases and $n_2 = n - n_1$ controls), who are allgenetically independent of each other and ethnically homogeneous. For each individual i, let G_i be the observed genotype, Z_i be the unobserved halotype pair for the i-th individual (which can be inferred from G_i), Y_i be the binary case-control status of the i-th individual, then the complete retrospective likelihood is:

$$L_{cc} = \prod_{i=1}^{n_1} P(Z_i|Y_i = 1, \Psi) \prod_{i=n_1+1}^{n} P(Z_i|Y_i = 0, \Psi)$$

where ϕ is a collection of parameters, including individual haplotype effect (β), haplotype frequencies ($\mathbf{f}$) and other hyperparameters.

2.1.2 famLBL

FamLBL has a similar formulation to LBL. For each family $j, j = 1, 2, \ldots, m$, consider a "matched pair" design where each (affected child, father, mother) trio is decomposed into a pair of haplotypes transmitted to the offspring (Z_{jc}), and a pair of haplotypes not transmitted to the offspring (Z_{ju}). The pair not transmitted can be considered as a pseudo control. Similarly, let G_{jc} and G_{ju} be the corresponding genotypes transmitted or not transmitted. Let Y_{jc} denote the disease status of the offspring ($Y_{jc} = 1$). Then, the likelihood can be formulated as,

$$L_t = \prod_{j=1}^{m} P(Z_{jc}|Y_{jc} = 1, \Psi) \times P(Z_{ju}|\Psi)$$

2.1.3 cLBL

cLBL combines the independent case-control design and the case-parent triad design. With the same notations as in LBL and famLBL, the likelihood is the product of the two previous likelihoods:

$$L_{comb} = L_{cc}(\Psi) \times L_t(\Psi) = \prod_{i=1}^{n_1} P(Z_i|Y_i = 1, \Psi) \prod_{i=n_1+1}^{n} P(Z_i|Y_i = 0, \Psi) \prod_{j=1}^{m} \{P(Z_{jc}|Y_{jc} = 1, \Psi) \times P(Z_{ju}|\Psi)\}$$

2.1.4 QBLbs

For individuals $i = 1, 2, \cdots, n$, let Y_i be the quantitative trait value, $\boldsymbol{C}_i$ be a vector of non-genetic covariates (such as age and sex) and $\boldsymbol{R}_i$ be a vector of PC scores of length L to be added to the model. The PCs are calculated from a large quantity of null markers to represent the genetic ancestry background for each individual. We consider the following complete data likelihood,

$$\begin{aligned} L(\boldsymbol{\Psi}) &= \prod_{i=1}^{n} P(Y_i, Z_i \mid \boldsymbol{G}_i, \boldsymbol{C}_i, \boldsymbol{R}_i, \boldsymbol{\Psi}) \\ &= \prod_{i=1}^{n} P(Y_i \mid Z_i, \boldsymbol{C}_i, \boldsymbol{R}_i, \boldsymbol{\beta}) P(Z_i \mid \boldsymbol{G}_i, \boldsymbol{\zeta}), \end{aligned} \tag{1}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\beta}, \boldsymbol{\zeta})$ consists of the model parameters $\boldsymbol{\beta}$ and parameters $\boldsymbol{\zeta}$ for modeling haplotype frequencies to be explained in the next section.

2.2 Parameters

2.2.1 Binary Traits

The aforementioned set of parameters ϕ include the following parameters:

- $\beta_l, l = 1, 2, \ldots, k-1$, the effect of haplotype l compared to the baseline haplotype. This is the parameter of interests.
- α, the effect of the baseline haplotype.
- λ, the hyperparameter controlling the degree of penalty for β_l.
- $\mathbf{f} = (f_1, f_2, \ldots, f_k)$, the frequency distribution of all haplotypes present in the dataset.
- $Z = (h_l, h_{l'})$, the unobserved haplotype pair an individual has.
- a_z, the probability of an individual having haplotype pair $Z = (h_l, h_{l'})$.
- d, the inbreeding coefficient. $d = 0$ would indicate the population is in Hardy-Weinberg Equilibrium. $d > 0$ would indicate an inbreeding populating and $d < 0$ indicate an outbreeding population.

Next we detail the parameters.

We connect β_l's with the likelihood through a logistic model. let θ be the odds of disease given a specific haplotype pair Z (i.e., $\theta = P(Y = 1 \mid Z)/P(Y = 0 \mid Z)$), then, we model the log odds ratio θ as,

$$\log \theta = \alpha + X\beta$$

where X is a row vector and each X is the design vector associated with haplotype pair Z, and α is log odds of the pre-selected baseline haplotype.

It is worth noting that each β_l measures the effect of haplotype l in contrast of the baseline haplotype. Therefore, choosing different baseline haplotypes might result in different β values. Only selecting a baseline that is not associated with the disease (i.e., $\alpha = 0$) will yield a correct interpretation. Choosing a haplotype that is associated with the disease might lead to loss of power in detecting other associated haplotypes and false positives. Therefore, one needs to take extra care when choosing the baseline haplotype. One way to avoid such scenarios is to use different baseline haplotypes. By default, the most frequent haplotype is chosen as the baseline.

Let $\mathbf{f} = (f_1, f_2, \ldots, f_k)$ denote frequency distribution of k distinct haplotypes. And let $a_z(\mathbf{f}, d)$, the frequency of an individual with a specific haplotype pair $Z = (h_l, h_{l'})$ be modelled as:

$$a_z(\mathbf{f}, d) = \begin{cases} f_l^2 + d f_l (1 - f_l) & \text{if } h_l = h_{l'} \\ 2(1 - d) f_l f_{l'} & \text{if } h_l \neq h_{l'} \end{cases},$$

where d is the within-populatin inbreeding coefficient. $d = 0$ denotes Hardy-Weinberg Equilibrium, $d > 0$ denotes excessive inbreeding and $d < 0$ denotes outbreeding. This allows us the freedom away from the assumption of Hardy-Weinberg equilibrium, as the effect of inbreeding/outbreeding can be modeled with d. When $d = 0$, the model is assuming HWE in the population.

2.2.2 Quantitative Traits

The set of parameters Ψ include the following parameters:

- $\beta_l, l = 1, 2, \ldots, q$, the effect of haplotypes excluding the baseline, covariates and PCs. This is the parameter of interests.
- α, the effect of the baseline haplotype.
- λ, the hyperparameter controlling the degree of penalty for β_l.
- $\mathbf{f} = (f_1, f_2, \ldots, f_k)$, the frequency distribution of all haplotypes present in the dataset.
- $Z = (h_l, h_{l'})$, the unobserved haplotype pair an individual has.
- d, the inbreeding coefficient. $d = 0$ would indicate the population is in Hardy-Weinberg Equilibrium. $d > 0$ would indicate an inbreeding populating and $d < 0$ indicate an outbreeding population.
- σ^2, the random error term for the linear model.

Given $Z_i, \boldsymbol{C}_i, \boldsymbol{R}_i$ and $\boldsymbol{\beta}$, Y_i is assumed to follow the normal distribution,

$$Y_i \mid Z_i, \boldsymbol{C}_i, \boldsymbol{R}_i, \boldsymbol{\beta} \sim N(\boldsymbol{X}_i^T \boldsymbol{\beta}, \sigma_e^2), \tag{2}$$

where $\boldsymbol{X}_i^T = (1, \boldsymbol{x}_{Hi}^T, \boldsymbol{x}_{Ci}^T, \boldsymbol{x}_{Ri}^T)$ is a row vector of model covariates. $\boldsymbol{x}_{Hi}^T = (x_1, \cdots, x_{S-1})$ is a vector of haplotype copies, where x_s denotes the number of copies for haplotype h_l. $\boldsymbol{x}_{Ci}^T$ and $\boldsymbol{x}_{Ri}^T$ are vectors for covariates and PCs, respectively. If discrete covariate exists, $\boldsymbol{x}_{Ci}^T$ includes a collection of dummy variables corresponding to that discrete covariate. Suppose the length of $\boldsymbol{X}_i^T$ is $q + 1$.

For the second term $P(Z_i \mid \boldsymbol{G}_i, \boldsymbol{\zeta})$, notice that $P(Z_i \mid \boldsymbol{G}_i, \boldsymbol{\zeta}) = 0$ if a haplotype pair is not compatible with the observed genotypes. Therefore, we only need to consider all possible haplotype pairs that are compatible with $\boldsymbol{G}_i$. For those compatible haplotype pairs, $P(Z_i \mid \boldsymbol{G}_i, \boldsymbol{\zeta})$ does not depend on $\boldsymbol{G}_i$ and is modeled by,

$$P(Z_i = h_l/h_l' \mid \boldsymbol{\zeta}) = \delta_{ll'} d f_l + (2 - \delta_{ll'})(1 - d) f_l f_{l'}, \tag{3}$$

where $\boldsymbol{\zeta} = (\boldsymbol{f}, d)$, $\boldsymbol{f}$ is a vector of haplotype frequencies, $d \in (-1, 1)$ is the within-population inbreeding coefficient, and $\delta_{ll'} = 1(0)$ if $h_l = h_l'(h_l \neq h_l')$. Then the complete likelihood can be expressed as,

$$L(\boldsymbol{\Psi}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\Big\{ - \frac{(Y_i - \boldsymbol{X}_i^T \boldsymbol{\beta})^2}{2\sigma_e^2} \Big\} \big\{ \delta_{ll'} d f_l + (2 - \delta_{ll'})(1 - d) f_l f_{l'} \big\}. \tag{4}$$

2.3 Priors

2.3.1 β

To penalize unassociated haplotype or PC effects and reduce dimension, double exponential (Laplace) distribution is used as the prior distribution for each β_l,

$$\pi(\beta_l \mid \lambda) = \frac{\lambda}{2} \exp\left(-\lambda \mid \beta_l \mid\right)$$

The hyperparameter λ controls the level of shrinkage. A larger value of λ indicates more shrinkage.

2.3.2 λ

Instead of picking a fixed λ, we let λ follow a Gamma(a, b) distribution with pdf

$$\pi(\lambda) = b^a \Gamma(a)^{-1} \lambda^{a-1} \exp\left(-b\lambda\right)$$

2.3.3 f and d

For the parameters involved in frequency calculation, we use Dirichlet(1,1,. . . ,1) distribution as the prior distribution for haplotype frequency distribution parameters $\mathbf{f}$. The prior distribution for the inbreeding coefficient d is set as unif$(\max_l\{-f_l/(1-f_l)\}, 1)$.

2.3.4 Z

For each individual i, we assign discrete uniform priors to all haplotypes compatible with the observed genotype. Therefore during each iteration, the haplotype will get updated according to the likelihood of each compatible haplotype pair.

2.3.5 σ^2

For QBLbs only, we consider the Inverse-Gamma(a_e, b_e) prior for σ^2 with pdf

$$\pi(\sigma^2) = \frac{b_e^{a_e}}{\Gamma(a_e)} \sigma^{2(a_e-1)} \exp(-\frac{b_e}{\sigma^2})$$

2.4 Inferences on Posterior Samples

Once the Markov Chain has converged, one can carry out inference based on posterior samples. The package includes built-in functions for inference based on posterior samples of β, providing estimates for OR (binary traits only), CI and Bayes Factor.

2.4.1 Bayes Factor

Bayes Factor is defined as the ratio between posterior odds and prior odds.

Since the prior and posterior distirbutions for all β_l's are both continuous, we cannot directly calculate the prior or posterior odds of $|\beta_l| = 0$. So, we opt to test $H_0 : |\beta_l| \leq \epsilon$ where ϵ is a pre-defined a small number. The odds is calculated as $P(|\beta| > \epsilon)/P(|\beta| \leq \epsilon)$ for both posterior and prior distributions. Then BF is the ratio between the two odds.

If all posterior $\mid \beta_l \mid$ exceed ϵ, then we set BF = 999 for computational considerations.

2.4.2 OR and CI

We also provide an odds ratio (OR) estimate based on posterior sample mean for binary traits, and a 95% credible interval (CI) estimate based posterior samples for all types of traits.

3 Using LBL

All LBL algorithms take some common input (genotypes, phenotypes, starting parameters, etc). First we detail those parameters, and then we follow up with examples for all three algorithms with a simulated dataset.

3.1 Binary Traits

3.1.1 Data Input

LBL takes data in pedigree format, regardless of the type of the design. The objects should be either a matrix or a data frame, consisting of n rows (n = number of individuals) and $6 + 2 \times p$ columns (p= number of SNPs). The first 6 columns of the data describe the pedigree relationship and the phenotype of the individual, and the last $2 \times p$ columns describe the genotype information of the individual, with each marker taking up 2 columns. The genotype data can be either alphabetic or numeric.

The first 6 columns of the dataset should consist of:

- Family ID: A number denoting to which family this individual belongs. Related individuals should share the same family ID, while unrelated individuals should have different family IDs.
- Individual ID: The unique identifier of this individual. This ID should be unique within each family.
- Father ID: ID of the father of this individual. If the father is unknown, or the individual is a founder, then father ID = 0.
- Mother ID: ID of the mother of this individual. If the mother is unknown, or the individual is a founder, then mother ID = 0.
- Sex: the gender of an individual. Male =1 and female = 2.
- Phenotype: affection status of an individual. A case should have the value of 2 and a control should have the value of 1. Individuals with unknown status should have the value of 0 and they are treated as controls in the analysis.

More information about the format can be found here.

The LBL package includes two example datasets: `fam` includes 250 case-parent trios, while `cac` includes 250 independent cases and 250 independent controls. Both datasets consist of 5 no-recombining SNPs. Below is a look of the beginning of these datasets.

```
library(LBL)
data(cac)
data(fam)
head(fam)
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        0
#> 2        1        2        0        0        2        1        0
#> 3        1        3        1        2        2        2        1
#> 4        2        1        0        0        1        1        1
#> 5        2        2        0        0        2        1        0
#> 6        2        3        1        2        1        2        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        1         0         1         0         0
#> 2        1        1         1         1         1         0
#> 3        0        0         1         0         1         1
#> 4        1        0         1         0         1         1
#> 5        1        1         0         1         0         0
#> 6        1        1         0         1         0         0
#>   column 14 column 15 column 16
#> 1         1         0         1
#> 2         0         0         0
#> 3         0         1         0
#> 4         0         1         0
#> 5         1         0         1
#> 6         1         0         1
head(cac)
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        1
#> 2        2        1        0        0        1        1        1
#> 3        3        1        0        0        1        1        1
#> 4        4        1        0        0        1        1        1
#> 5        5        1        0        0        1        1        1
#> 6        6        1        0        0        1        1        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        0         1         0         1         1
#> 2        1        1         0         1         0         1
#> 3        0        1         1         1         1         0
#> 4        1        1         1         1         1         1
#> 5        1        1         1         1         1         0
#> 6        1        0         0         0         0         1
#>   column 14 column 15 column 16
#> 1         1         1         1
#> 2         1         1         1
#> 3         0         0         0
#> 4         1         1         1
#> 5         1         0         1
#> 6         1         1         1
```

Note that for case-control data, father ID and mother ID are both 0.

3.1.2 Other Parameters

There are some other parameters that need to be specified for the MCMC algorithm. They are:

- starting values: providing a starting values for the MCMC.
- a, b: the hyperparameters for λ, which controls the shrinkage effect of β. See Priors section for details. Different values of a and b have some effect on the outcome, the details can be found in paper.
- e: the number ϵ used as a cutoff as if β can be treated as 0. The default is 0.1. See Inferences on Posterior Samples section for details.

3.2 Quantitative Traits

The QBLbs function is designed for rare haplotype effects estimation for quantitative traits. The data input should be either a matrix or data frame, consisting of n rows ($n =$ number of individuals) and $1+n.cov+2\times p$ (allelic format) or $1+n.cov+p$ (genotypic format) columns ($n.cov=$ number of covariates and PCs, $p=$ number of SNPs). The first column is the trait, followed by the columns of covariates and PCs. If in allelic format, the other 2*p columns are allels with one column for each allele of the single-locus genotypes. If in genotypic format, the other p columns are genotypes with one column for each SNP.

The LBL package includes one example dataset: QBLbsData consisting of 5 PCs and 5 SNPs for 960 individuals. Below is a look of the beginning of these datasets.

```
library(LBL)
data(QBLbsData)
head(QBLbsData)
#>             Y        PC1        PC2        PC3         PC4        PC5
#> 2  0.9116686 -0.2702250 -0.3334982 -0.2942433 -0.10333472 -0.1570381
#> 3  0.2997905 -0.2703233 -0.3345437 -0.2985917 -0.08317809 -0.1607751
#> 4 -0.6068688 -0.2703051 -0.3336549 -0.3006124 -0.11480781 -0.1971382
#> 5 -1.1412648 -0.2702159 -0.3330559 -0.2952761 -0.11915872 -0.1752794
#> 6  0.2330924 -0.2703144 -0.3341000 -0.2995938 -0.09901067 -0.1789419
#> 7 -1.3094843 -0.2707457 -0.3328562 -0.3014475 -0.11490629 -0.1682322
#>   1_loc1 1_loc2 2_loc1 2_loc2 3_loc1 3_loc2 4_loc1 4_loc2 5_loc1
#> 2      0      0      0      0      1      1      0      0      0
#> 3      0      0      0      0      1      1      0      0      0
#> 4      0      0      0      0      1      1      0      0      0
#> 5      0      0      0      0      1      1      0      0      0
#> 6      0      0      0      0      1      1      0      0      0
#> 7      0      0      0      0      1      1      0      0      0
#>   5_loc2
#> 2      0
#> 3      0
#> 4      0
#> 5      0
#> 6      0
#> 7      0
```

4 Example

4.1 LBL

LBL is the original version of logistic Bayesian LASSO that detects association between common diseases and rare haplotypes. It analyzes independent case-control data. In the LBL package, the corresponding function is LBL.

The procedure below provides a simple example of running LBL on dataset cac. cac is a sample input composed of case-control data. Note that the data is in pedigree format where the first 6 columns are: family ID, individual ID, father ID, mother ID, sex, and phenotype. Since the cases and controls are required to be independent, the family IDs of the individuals are all different. The last $2 \times p$ columns represent the genotype information of the p SNPs. In this example, $p = 5$.

By default, the LBL function will return a list of haplotype names (haplotypes), haplotype frequencies (freq), odds ratios (OR), credible intervals of odds ratio (OR.CI), and Bayes factors (BF). For haplotypes and freq, the last value corresponds to the baseline haplotype whose OR, OR.CI, and BF cannot be calculated. If better output summary is preferred, the user can save the outcome list from LBL and call the print_LBL_summary function. Significant haplotypes will be indicated with *+ (risk) or *- (protective).

LBL can also return the entire posterior samples for all parameters. To acquire the entire samples, just set the summary parameter of LBL to be FALSE.

```
library(LBL)
head(cac)
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        1
#> 2        2        1        0        0        1        1        1
#> 3        3        1        0        0        1        1        1
#> 4        4        1        0        0        1        1        1
#> 5        5        1        0        0        1        1        1
#> 6        6        1        0        0        1        1        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        0         1         0         1         1
#> 2        1        1         0         1         0         1
#> 3        0        1         1         1         1         0
#> 4        1        1         1         1         1         1
#> 5        1        1         1         1         1         0
#> 6        1        0         0         0         0         1
#>   column 14 column 15 column 16
#> 1         1         1         1
#> 2         1         1         1
#> 3         0         0         0
#> 4         1         1         1
#> 5         1         0         1
#> 6         1         1         1
set.seed(1)
LBL.obj<-LBL(cac,burn.in = 40000,num.it = 70000,summary = T)
#> running LBL...
LBL.obj
#> $haplotypes
#> [1] "h01100" "h10100" "h11011" "h11100" "h11111" "h10011"
#>
#> $freq
#> [1] 0.284182893 0.004871725 0.010559994 0.137745431 0.092719217
#> [6] 0.469920740
#>
#> $OR
#> [1] 0.9558737 1.3156901 2.3043371 1.2984247 1.8394852
#>
#> $OR.CI
#>           2.5%    97.5%
#> [1,] 0.7074538 1.277637
#> [2,] 0.3334896 5.947696
#> [3,] 0.8976443 7.526092
#> [4,] 0.9177489 1.855987
#> [5,] 1.2351585 2.796245
#>
#> $BF
#> [1]  0.1076655  0.6294617  1.8826335  0.4864193 17.5588280
print_LBL_summary(LBL.obj)
#>      Hap        Freq        OR  OR Lower OR Upper         BF
#> 1 h01100 0.284182893 0.9558737 0.7074538 1.277637  0.1076655
#> 2 h10100 0.004871725 1.3156901 0.3334896 5.947696  0.6294617
#> 3 h11011 0.010559994 2.3043371 0.8976443 7.526092  1.8826335
#> 4 h11100 0.137745431 1.2984247 0.9177489 1.855987  0.4864193
#> 5 h11111 0.092719217 1.8394852 1.2351585 2.796245 17.5588280 *+
#> 6 h10011 0.469920740        NA        NA       NA         NA
#> ---
#> Signif.codes: Risk '*+' Protective '*-' Not significant ' '
```

4.2 famLBL

famLBL is the logistic Bayesian LASSO that uses case-parent triad (family trio) data to detect rare haplotype effects. In the LBL package, the corresponding function is famLBL.

The procedure below provides a simple example of running famLBL on dataset fam. fam is a sample input composed of case-parent triad data. Again, the data is in pedigree format where the first 6 columns are: family ID, individual ID, father ID, mother ID, sex, and phenotype. Since the data are of case-parent triad, every three individuals share the same family ID. Within the same family, the affected child's father ID will be the father's individual ID; the affected child's mother ID will be the mother's individual ID. Again, the last $2 \times p$ columns represent the genotype information of the p SNPs. In this example, $p = 5$.

By default, the famLBL function will return a list of haplotype names (haplotypes), haplotype frequencies (freq), odds ratios (OR), credible intervals of odds ratio (OR.CI), and Bayes factors (BF). For haplotypes and freq, the last value corresponds to the baseline haplotype whose OR, OR.CI, and BF cannot be calculated. If better output summary is preferred, the user can save the outcome list from famLBL and call the print_LBL_summary function. Significant haplotypes will be indicated with *+ (risk) or *- (protective).

famLBL can also return the entire posterior samples for all parameters. To acquire the entire samples, just set the summary parameter of famLBL to be FALSE.

```
library(LBL)
head(fam)
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        0
#> 2        1        2        0        0        2        1        0
#> 3        1        3        1        2        2        2        1
#> 4        2        1        0        0        1        1        1
#> 5        2        2        0        0        2        1        0
#> 6        2        3        1        2        1        2        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        1         0         1         0         0
#> 2        1        1         1         1         1         0
#> 3        0        0         1         0         1         1
#> 4        1        0         1         0         1         1
#> 5        1        1         0         1         0         0
#> 6        1        1         0         1         0         0
#>   column 14 column 15 column 16
#> 1         1         0         1
#> 2         0         0         0
#> 3         0         1         0
#> 4         0         1         0
#> 5         1         0         1
#> 6         1         0         1
set.seed(1)
famLBL.obj<-famLBL(fam,burn.in = 40000,num.it = 70000,summary = T)
#> A total of 250 families are in the study
#> running famLBL...
famLBL.obj
#> $haplotypes
#> [1] "h01100" "h10100" "h11011" "h11100" "h11111" "h10011"
#>
#> $freq
#> [1] 0.30659192 0.01289241 0.01404503 0.13737667 0.11584437 0.41324960
#>
#> $OR
#> [1] 1.2998064 0.4108729 1.6021311 1.3286153 2.3342274
#>
#> $OR.CI
#>             2.5%    97.5%
#> [1,] 0.97233977 1.749926
#> [2,] 0.06402784 1.412998
#> [3,] 0.67895485 4.351326
#> [4,] 0.91288777 1.953164
#> [5,] 1.61831847 3.373616
#>
#> $BF
#> [1]   0.6322156   1.1696694   0.7129140   0.5806387 999.0000000
```

```
print_LBL_summary(famLBL.obj)
#>      Hap       Freq        OR   OR Lower OR Upper          BF
#> 1 h01100 0.30659192 1.2998064 0.97233977 1.749926   0.6322156
#> 2 h10100 0.01289241 0.4108729 0.06402784 1.412998   1.1696694
#> 3 h11011 0.01404503 1.6021311 0.67895485 4.351326   0.7129140
#> 4 h11100 0.13737667 1.3286153 0.91288777 1.953164   0.5806387
#> 5 h11111 0.11584437 2.3342274 1.61831847 3.373616 999.0000000 *+
#> 6 h10011 0.41324960        NA         NA       NA          NA
#> ---
#> Signif.codes: Risk '*+' Protective '*-' Not significant ' '
```

4.3 cLBL

cLBL is the latest logistic Bayesian LASSO that detects association between common diseases and rare haplotypes. It analyzes case-control and case-parent triad data simultaneously and thus take advantage of the larger sample size from the combined data. In the LBL package, the corresponding function is cLBL.

The procedure below provides a simple example of running cLBL on dataset cac and fam. The first and the second parameters required from cLBL are case-parent triad and case-control data, respectively. These two dataset should be both in pedigree format. The rest parameter settings of cLBL are similar to those of LBL and famLBL.

By default, the cLBL function will return a list of haplotype names (haplotypes), haplotype frequencies (freq), odds ratios (OR), credible intervals of odds ratio (OR.CI), and Bayes factors (BF). For haplotypes and freq, the last value corresponds to the baseline haplotype whose OR, OR.CI, and BF cannot be calculated. If better output summary is preferred, the user can save the outcome list from cLBL and call the print_LBL_summary function. Significant haplotypes will be indicated with *+ (risk) or *- (protective).

cLBL can also return the entire posterior samples for all parameters. To acquire the entire samples, just set the summary parameter of cLBL to be FALSE.

```
library(LBL)
head(cac)
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        1
#> 2        2        1        0        0        1        1        1
#> 3        3        1        0        0        1        1        1
#> 4        4        1        0        0        1        1        1
#> 5        5        1        0        0        1        1        1
#> 6        6        1        0        0        1        1        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        0         1         0         1         1
#> 2        1        1         0         1         0         1
#> 3        0        1         1         1         1         0
#> 4        1        1         1         1         1         1
#> 5        1        1         1         1         1         0
#> 6        1        0         0         0         0         1
#>   column 14 column 15 column 16
#> 1         1         1         1
#> 2         1         1         1
#> 3         0         0         0
#> 4         1         1         1
#> 5         1         0         1
#> 6         1         1         1
head(fam)
```

```
#>   column 1 column 2 column 3 column 4 column 5 column 6 column 7
#> 1        1        1        0        0        1        1        0
#> 2        1        2        0        0        2        1        0
#> 3        1        3        1        2        2        2        1
#> 4        2        1        0        0        1        1        1
#> 5        2        2        0        0        2        1        0
#> 6        2        3        1        2        1        2        1
#>   column 8 column 9 column 10 column 11 column 12 column 13
#> 1        1        1         0         1         0         0
#> 2        1        1         1         1         1         0
#> 3        0        0         1         0         1         1
#> 4        1        0         1         0         1         1
#> 5        1        1         0         1         0         0
#> 6        1        1         0         1         0         0
#>   column 14 column 15 column 16
#> 1         1         0         1
#> 2         0         0         0
#> 3         0         1         0
#> 4         0         1         0
#> 5         1         0         1
#> 6         1         0         1
set.seed(1)
cLBL.obj<-cLBL(fam,cac,burn.in = 40000,num.it = 70000,summary = T)
#> A total of 250 families are in the study
#> running cLBL...
cLBL.obj
#> $haplotypes
#> [1] "h01100" "h10100" "h11011" "h11100" "h11111" "h10011"
#>
#> $freq
#> [1] 0.296088636 0.007483365 0.010292924 0.136248603 0.102943013
#> [6] 0.446943460
#>
#> $OR
#> [1] 1.133830 0.591277 2.225438 1.346486 2.160047
#>
#> $OR.CI
#>           2.5%    97.5%
#> [1,] 0.9311700 1.394997
#> [2,] 0.1651112 1.530622
#> [3,] 1.0523130 5.221890
#> [4,] 1.0411274 1.756443
#> [5,] 1.6207662 2.861191
#>
#> $BF
#> [1]   0.1605677   0.7165455   3.8500719   1.3223980 999.0000000
print_LBL_summary(cLBL.obj)
#>      Hap        Freq       OR  OR Lower OR Upper          BF
#> 1 h01100 0.296088636 1.133830 0.9311700 1.394997   0.1605677
#> 2 h10100 0.007483365 0.591277 0.1651112 1.530622   0.7165455
#> 3 h11011 0.010292924 2.225438 1.0523130 5.221890   3.8500719 *+
#> 4 h11100 0.136248603 1.346486 1.0411274 1.756443   1.3223980
#> 5 h11111 0.102943013 2.160047 1.6207662 2.861191 999.0000000 *+
#> 6 h10011 0.446943460        NA        NA        NA          NA
#> ---
#> Signif.codes: Risk '*+' Protective '*-' Not significant ' '
```

4.4 QBLbs

QBLbs is the Quantitative Bayesian LASSO that detects rare haplotype effects with a quantitative trait while using PCs to adjust for population stratification. In the LBL package, the corresponding function is QBLbs.

The procedure below provides a simple example of running QBLbs on dataset QBLbsData. QBLbsData is a sample input consisting of a quantitative trait, 5 PC scores and phenotype for 960 individuals. Again, the last $2 \times p$ columns represent the genotype information of the p SNPs. In this example, $p = 5$.

The QBLbs function will return a list of BF (BF), coefficient estimates (beta), credible intervals of coefficient estimates (CI.beta), credible interval of λ (CI.lambda), credible interval of d (CI.D) and haplotype frequencies (freq). For freq, the last value corresponds to the baseline haplotype whose effects are not of interest.

```
library(LBL)
head(QBLbsData)
#>            Y        PC1        PC2        PC3         PC4        PC5
#> 2  0.9116686 -0.2702250 -0.3334982 -0.2942433 -0.10333472 -0.1570381
#> 3  0.2997905 -0.2703233 -0.3345437 -0.2985917 -0.08317809 -0.1607751
#> 4 -0.6068688 -0.2703051 -0.3336549 -0.3006124 -0.11480781 -0.1971382
#> 5 -1.1412648 -0.2702159 -0.3330559 -0.2952761 -0.11915872 -0.1752794
#> 6  0.2330924 -0.2703144 -0.3341000 -0.2995938 -0.09901067 -0.1789419
#> 7 -1.3094843 -0.2707457 -0.3328562 -0.3014475 -0.11490629 -0.1682322
#>   1_loc1 1_loc2 2_loc1 2_loc2 3_loc1 3_loc2 4_loc1 4_loc2 5_loc1
#> 2      0      0      0      0      1      1      0      0      0
#> 3      0      0      0      0      1      1      0      0      0
#> 4      0      0      0      0      1      1      0      0      0
#> 5      0      0      0      0      1      1      0      0      0
#> 6      0      0      0      0      1      1      0      0      0
#> 7      0      0      0      0      1      1      0      0      0
#>   5_loc2
#> 2      0
#> 3      0
#> 4      0
#> 5      0
#> 6      0
#> 7      0
set.seed(1)
QBLbs.obj<-out<-QBLbs(QBLbsData, numSNPs=5, cov=5)
QBLbs.obj
#> $BF
#>    h00100    h00000    h00101    h01000    h01010    h10000
#>  "0.1137"  "6.5341"  "0.0751"  "0.0276"  "0.0686"    ">100"
#>       PC1       PC2       PC3       PC4       PC5
#>  "0.4795"  "0.4071"  "0.4438" "39.7289"    ">100"
#>
#> $beta
#>      h00100      h00000      h00101      h01000      h01010
#> -0.10188464 -0.26877298 -0.08425849 -0.02736908  0.01664008
#>      h10000         PC1         PC2         PC3         PC4
#>  0.27439935  0.05165756 -0.03258599  0.23363646  1.10655920
#>         PC5
#> -2.06506355
#>
#> $CI.beta
#>      Name      Lower       Upper
#> 1  h00100 -0.1829568 -0.02086597
#> 2  h00000 -0.4186951 -0.11466952
#> 3  h00101 -0.2264155  0.05825464
#> 4  h01000 -0.1734455  0.11973106
#> 5  h01010 -0.2174381  0.25238832
#> 6  h10000  0.1894435  0.36126855
#> 7     PC1 -1.0185756  1.12027819
#> 8     PC2 -1.0103140  0.94729752
#> 9     PC3 -0.5851943  1.14286180
#> 10    PC4  0.3506757  1.85027164
#> 11    PC5 -2.7454833 -1.36928048
#>
#> $CI.lambda
#>  Lower  Upper
#> 0.8013 1.6721
#>
#> $CI.D
#>  Lower  Upper
#> 0.9933 1.0000
#>
#> $freq
#> h00000 h00101 h01000 h01010 h10000 h00100
#> 0.0475 0.0478 0.0475 0.0167 0.2487 0.5918
```

References

Biswas, Swati, and Shili Lin. 2012. "Logistic Bayesian LASSO for Identifying Association with Rare Haplotypes and Application to Age-Related Macular Degeneration." *Biometrics* 68 (2): 587–97.

Geman, Stuart, and Donald Geman. 1984. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6: 721–41.

Hastings, W Keith. 1970. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika* 57 (1): 97–109.

Metropolis, Nicholas, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics* 21 (6): 1087–92.

Wang, Meng, and Shili Lin. 2014. "FamLBL: detecting Rare Haplotype Disease Association Based on Common SNPs Using Case-Parent Triads." *Bioinformatics* 30 (18): 2611–18.

Zhou, Xiaofei, Meng Wang, and Shili Lin. 2019. "cLBL: Combined logistic Bayesian LASSO for Detecting Rare Associated Haplotypes Using Independent Case, Control and Family Trio Data." *Manuscript*.