

LBL: Logistic Bayesian LASSO for Detecting Rare Haplotype Associations

Contents

1	Introduction	1
2	Methods	1
2.1	Likelihood	2
2.2	Parameters	2
2.3	Priors	3
2.4	Inferences on Posterior Samples	4
3	Using LBL	4
3.1	Data Input	5
3.2	Other Parameters	6
4	Example	6
4.1	LBL	6
4.2	famLBL	7
4.3	cLBL	9
	References	10

1 Introduction

LBL (Logistic Bayesian LASSO) ([1](#), [2](#), [3](#)) is a Bayesian genetic association test aimed at detecting association between rare *haplotypes* (which could be formed by common SNPs) and diseases. Currently there are three different LBLs that handle different types of study designs: one for independent case-control data (LBL, [1](#)), one for case-parent triad (family trio) data (famLBL, [2](#)) and one for a combination of data from the two designs (cLBL, [3](#)). LBLs take genotype and phenotype (binary traits) 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. The likelihood portions of all LBL methods are formulated with retrospectively likelihoods, and are connected with disease model via a logistic link. The priors of LBL methods include a double exponential distribution on the haplotypic 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, [4](#), [5](#), and [6](#)) is used to sample from the posterior distribution.

In the following, we first discuss the likelihoods for all three 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 three 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 $j, j = 1, 2, \dots, m$, 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:

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

2.2 Parameters

The aforementioned set of parameters ϕ include the following parameters:

- $\beta_l, l = 1, 2, \dots, k - 1$, the effect of haplotype l compared to the baseline haplotype. This is the parameter of interests.

- α , the effect of the baseline haplotype.
- $\mathbf{f} = (f_1, f_2, \dots, f_k)$, 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 population 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., $\theta = P(Y = 1 | Z) / P(Y = 0 | Z)$), 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 $\mathbf{f} = (f_1, f_2, \dots, f_k)$ 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:

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

where d is the within-population 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.

2.3 Priors

2.3.1 β

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

$$\pi(\beta_l | \lambda) = \frac{\lambda}{2} \exp(-\lambda |\beta_l|)$$

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 $\text{Gamma}(a, b)$ distribution with pdf

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

2.3.3 \mathbf{f} and d

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

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.

2.4 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, 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 distributions 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 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 $|\beta_l|$ exceed ϵ , then we set $\text{BF} = 999$ for computational considerations.

2.4.2 OR and CI

We also provide an odds ratio (OR) estimate based on posterior sample mean and a 95% credible interval (CI) estimate based posterior samples.

3 Using LBL

All three 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 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)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  1  0  0  1           1     0     1     1     0     1     0
#> 2     1  2  0  0  2           1     0     1     1     1     1     1
#> 3     1  3  1  2  2           2     1     0     0     1     0     1
#> 4     2  1  0  0  1           1     1     1     0     1     0     1
#> 5     2  2  0  0  2           1     0     1     1     0     1     0
#> 6     2  3  1  2  1           2     1     1     1     0     1     0
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       0       1       0       1
#> 2       0       0       0       0
#> 3       1       0       1       0
#> 4       1       0       1       0
#> 5       0       1       0       1
#> 6       0       1       0       1
head(cac)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  1  0  0  1           1     1     1     0     1     0     1
#> 2     2  1  0  0  1           1     1     1     1     0     1     0
#> 3     3  1  0  0  1           1     1     0     1     1     1     1
#> 4     4  1  0  0  1           1     1     1     1     1     1     1
#> 5     5  1  0  0  1           1     1     1     1     1     1     1
#> 6     6  1  0  0  1           1     1     1     0     0     0     0
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       1       1       1       1
```

```
#> 2      1      1      1      1
#> 3      0      0      0      0
#> 4      1      1      1      1
#> 5      0      1      0      1
#> 6      1      1      1      1
```

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

3.2 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.

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)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  1  0  0  1          1      1      1      0      1      0      1
#> 2     2  1  0  0  0  1          1      1      1      1      0      1      0
#> 3     3  1  0  0  0  1          1      1      0      1      1      1      1
#> 4     4  1  0  0  0  1          1      1      1      1      1      1      1
#> 5     5  1  0  0  0  1          1      1      1      1      1      1      1
#> 6     6  1  0  0  0  1          1      1      1      0      0      0      0
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       1       1       1       1
#> 2       1       1       1       1
```

```

#> 3      0      0      0      0
#> 4      1      1      1      1
#> 5      0      1      0      1
#> 6      1      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 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)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  0  0  1         1      0      1      1      0      1      0
#> 2     1  2  0  0  2         1      0      1      1      1      1      1
#> 3     1  3  1  2  2         2      1      0      0      1      0      1
#> 4     2  1  0  0  1         1      1      1      0      1      0      1
#> 5     2  2  0  0  2         1      0      1      1      0      1      0
#> 6     2  3  1  2  1         2      1      1      1      0      1      0
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       0       1       0       1
#> 2       0       0       0       0
#> 3       1       0       1       0
#> 4       1       0       1       0
#> 5       0       1       0       1
#> 6       0       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)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  0  0  1         1       1       1       0       1       0
#> 2     2  1  0  0  1         1       1       1       1       0       1
#> 3     3  1  0  0  1         1       1       0       1       1       1
#> 4     4  1  0  0  1         1       1       1       1       1       1
#> 5     5  1  0  0  1         1       1       1       1       1       1
#> 6     6  1  0  0  1         1       1       1       0       0       0
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       1       1       1       1
#> 2       1       1       1       1
#> 3       0       0       0       0
#> 4       1       1       1       1
#> 5       0       1       0       1
#> 6       1       1       1       1
head(fam)
#>   famID ID fID mID Sex Infection snp1.1 snp1.2 snp2.1 snp2.2 snp3.1 snp3.2
#> 1     1  1  0  0  1         1       0       1       1       0       1
#> 2     1  2  0  0  2         1       0       1       1       1       1
#> 3     1  3  1  2  2         2       1       0       0       1       0
#> 4     2  1  0  0  1         1       1       1       0       1       0
#> 5     2  2  0  0  2         1       0       1       1       0       1
#> 6     2  3  1  2  1         2       1       1       1       0       1
#>   snp4.1 snp4.2 snp5.1 snp5.2
#> 1       0       1       0       1
#> 2       0       0       0       0
#> 3       1       0       1       0
#> 4       1       0       1       0
#> 5       0       1       0       1
#> 6       0       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 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 ' '

```

References

1. Biswas, S., and Lin, S. (2012). Logistic Bayesian LASSO for identifying association with rare haplotypes and application to age-related macular degeneration. *Biometrics* 68, 587–597.
2. Wang, M., and Lin, S. (2014). FamLBL: detecting rare haplotype disease association based on common SNPs using case-parent triads. *Bioinformatics* 30, 2611–2618.
3. Zhou, X., Wang, M., and Lin, S. (2019). Detecting rare associated haplotypes with independent case, control and family trio data: combined logistic Bayesian LASSO. *Statistical Methods in Medical Research*.
4. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 1087–1092.
5. Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–109.
6. Geman, S., and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 721–741.