The MR Data Challenge 2019: HDL and age-related macular degeneration

Chin Yang Shapland

19th of July 2019

Participants

Chin Yang Shapland[1,2] and Jack Bowden[1,2]

1. MRC Integrative Epidemiology Unit at the University of Bristol, U.K.
2. Population Health Sciences, University of Bristol, U.K.

Motivation

Age-related macular degeneration (AMD) is a painless eye-disease that eventually leads to vision loss. A GWAS have identified several rare and common variants located in gene regions that are associated with lipid levels (Fritsche et al. 2016). And later the reason for association was speculated and confirmed by separate studies (Leeuwen et al. 2018, J. Colijn et al. (2018)). Multivariable MR study showed evidence of a causal relationship between AMD and HDL cholesterol but not with LDL cholesterol and triglycerides (Burgess and Smith 2017). Following a Bayesian model averaging approach within a multivariable MR design, out of 30 metabolites, total cholesterol in extra-large HDL particles (XL.HDL.C) had the highest inclusion posterior probability as a risk factor for AMD (Zuber et al. 2019). With the same summary data used in (Zuber et al. 2019), we aim to find valid genetic instruments to infer causality between XL.HDL.C and AMD.

Data

Exposure: Total cholesterol in extra-large HDL particles (XL.HDL.C)

Outcome: Age-related macular degeneration (AMD)

Genetic instruments: 148, number of genetic variants available for both AMD (Fritsche et al. 2016) and XL.HDL.C (Kettunen et al. 2016)

Analysis methods

We have developed a method that aims to estimate the causal effect in the presence of pleiotropy with two-sample summary data MR. Specifically, we will make use of the technique of Bayesian model averaging and a Metropolis-Hastings algorithm to intelligently search the space of all 2^L models (L being the number of genetic instruments). Like all Bayesian approaches and incorporating the profile likelihood derived by Zhao and colleague (Zhao et al. 2018), our Bayesian model averaging algorithm is robust to weak instrument bias. Our technique naturally up-weights large sets of variants that furnish consistent, homogeneous estimates of causal effect, and down-weights sets of variants that provide heterogeneous estimates of causal effect.

For comparison, we will compare our method to inverse-variance weighted meta-analysis (IVW) and Robust Adjusted Profile Score (MR-RAPS) (Zhao et al. 2018), where the former acts as the baseline estimator and the latter is the classical counterpart to our method. IVW gives the pooled causal effect estimates from the Wald ratio of each instrument, weighted by their inverse-variance. MR-RAPS (Zhao et al. 2018) reduces the effect of outlying variants and robust to weak instrument bias but without the loss of precision. And achieves this by combining penalized weight function with profile-likelihood.

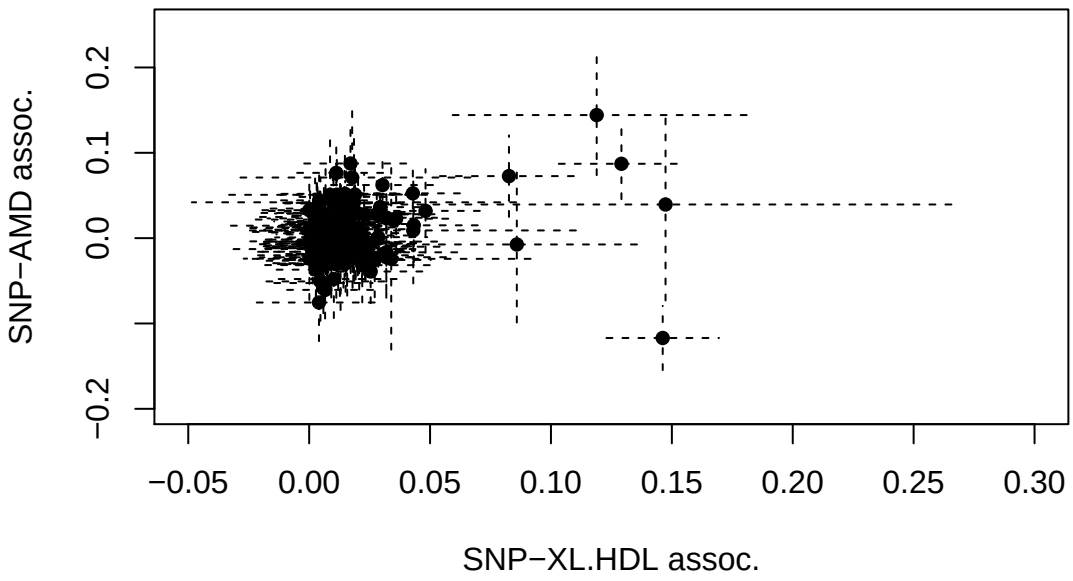

Figure 1: Scatter plot of the relationship between SNP-outcome and SNP-exposure association

Inconsistent estimation of the causal effect in balanced pleiotropy can happen if InSIDE assumption is violated (Instrument Strength Independent of Direct Effect). InSIDE assumes the genetic association with the exposure is independent of the direct genetic effect on outcome (Kolesár et al. 2015 Bowden, Davey Smith, and Burgess (2015)). Therefore we have extended our method to allow for violation to InSIDE by estimating two effects; one is the true causal effect estimated from valid instruments and the other is the biased causal effect estimated from instruments that violates the InSIDE assumption.

Results

Assuming InSIDE is valid

These 148 instruments have mean F-statistics of 4. Figure 2 shows convergence for the causal effect estimate was reached after 500,000 iterations and 100,000 burn-ins for both DL (65 secs) and full Bayesian (89 secs) approach. All the estimators have found a significant positive effect of XL.HDL.C on AMD, except IVW. But the from simulations and previous studies have shown that IVW is affected by weak instrument bias. Our approach have estimated stronger positive effects in comparison to MR-RAPS, however their 95% intervals do overlap (Figure 3). For inclusion posterior probability for each instrument, Figure 4, DL estimate have down weighted more instruments than the full Bayesian approach, this is due to the τ^2 are estimated rather being allowed to be any value within the gamma distribution.

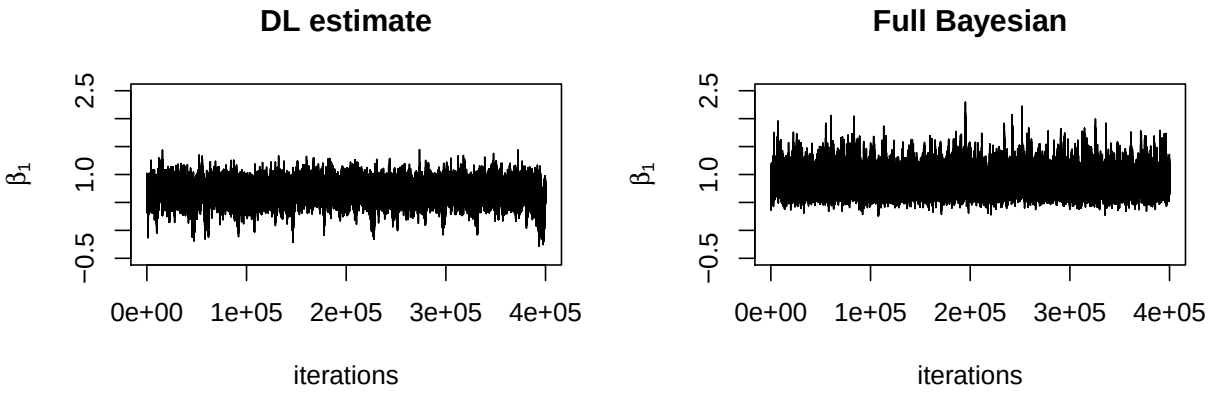

Figure 2: Convergence of causal effect estimate

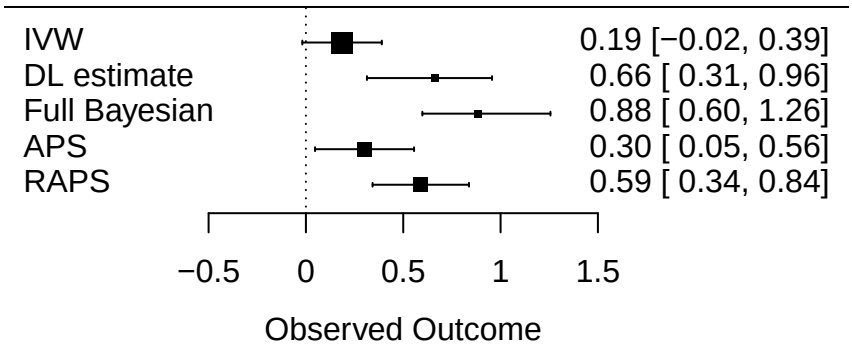

Figure 3: Forest plot for the summary of results from each estimator

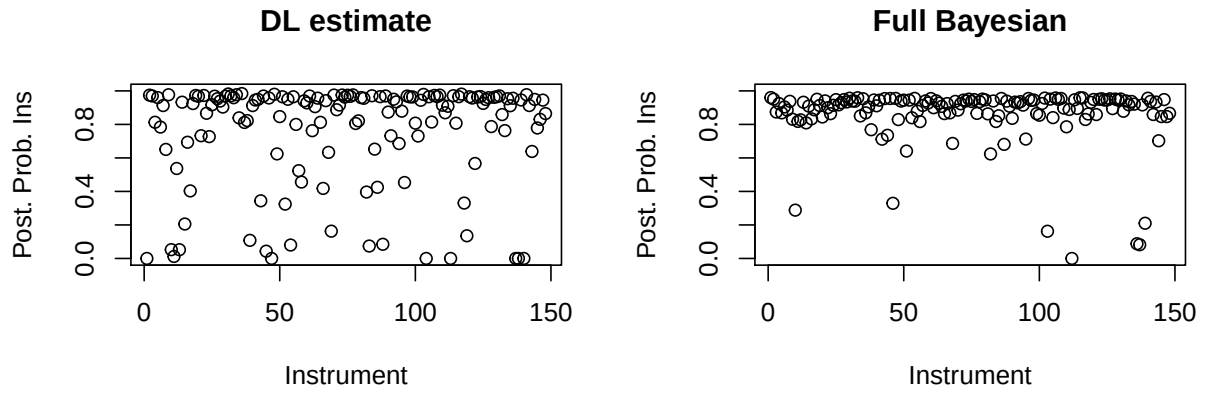

Figure 4: Inclusion Posterior Probaility

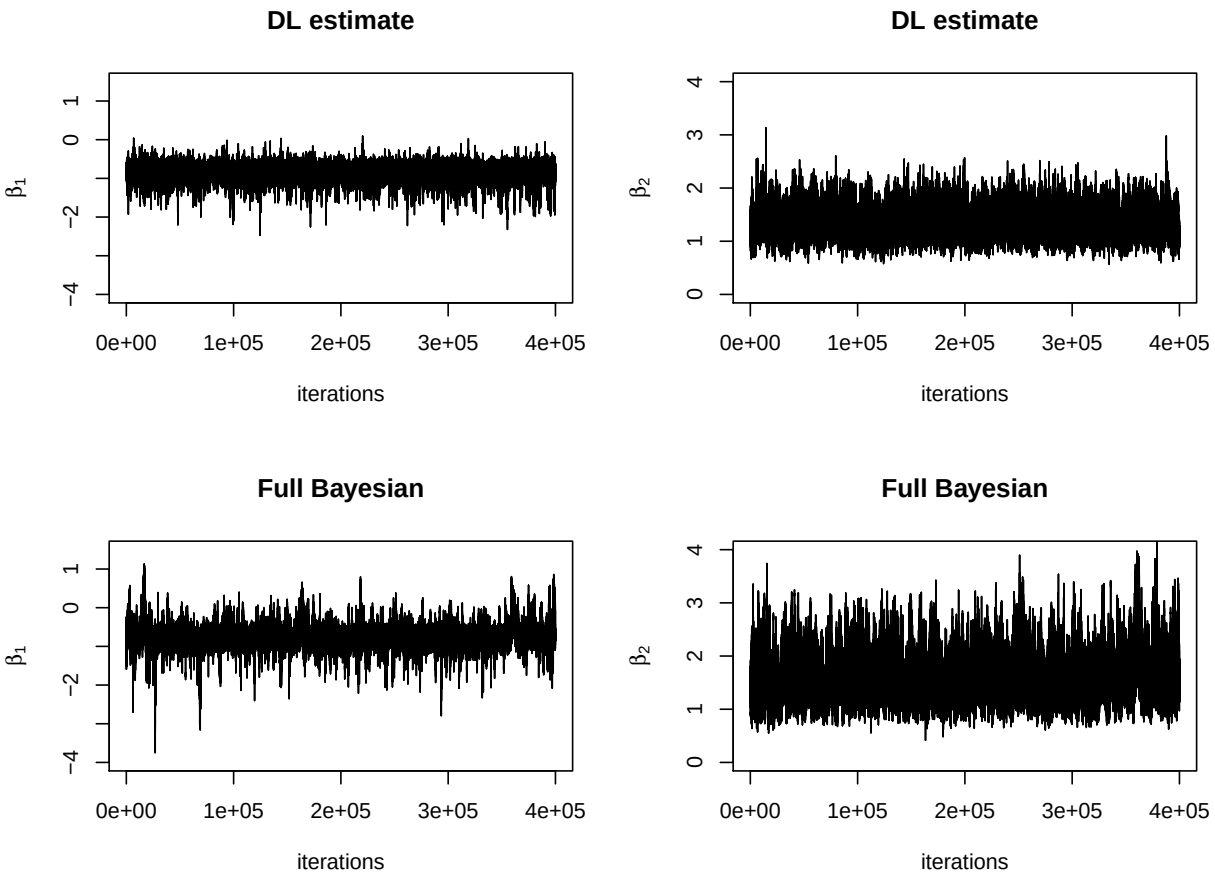

Figure 5: Convergence of causal effect estimate for violation of InSIDE

Assuming InSIDE is invalid

Convergence were reached show for both effect estimates after 500,000 iterations and 100,000 burn-ins in DL and full Bayesian approach, Figure 5. Our approach have estimated 2 opposite effects (Figure 6), demonstrating that some instruments have violated the InSIDE assumption. Aprroximately 25 instruments have strong inclusion posterior probability (> 0.8) to each of the 2 clusters, and 4 instruments belonging to neither, see Figure 7 and Figure 8. A little less for full Bayesian approach, as we have seen in the previous section that the Bayesian approach penalises outlier less harshly. Even though there is evidence for InSIDE violation, we cannot tell which is the true causal effect estimate, and will require external biological knowledge to why there are 2 distinct clusters of instruments estimating 2 opposite effects.

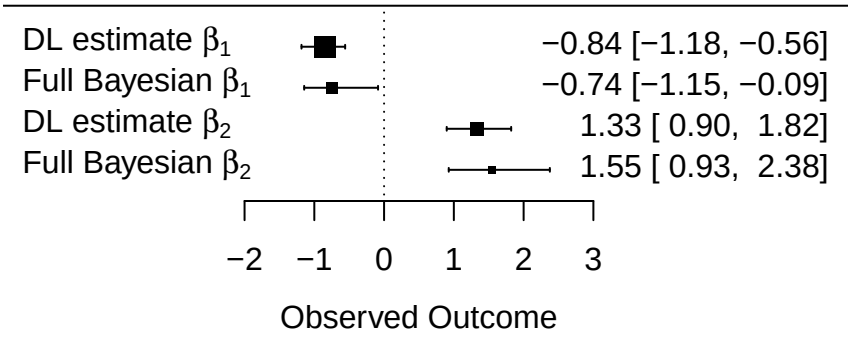

Figure 6: Forest plot for violation of InSIDE

Technical appendix

Description of the model

Suppose that we have data from a Mendelian randomisation study, where it consists of N individuals, each subject k has L independent genetic variants (G_{kj}), an exposure (X_k) and an outcome (Y_k) of interest. We assume the confounders (U) between X and Y are unknown. Then their relationship can be described by the following linear model

$$X_k = \sum_{j=1}^{L} \gamma_j G_{kj} + U_k + \epsilon_k^X \tag{1}$$

$$Y_k = \sum_{j=1}^{L} \alpha_j G_{kj} + \beta X_k + U_k + \epsilon_k^Y \tag{2}$$

where ϵ_k^X and ϵ_k^Y are independent error term for X and Y respectively. We can substitute (1) into (2) to have the reduced-form equation of Y

$$Y_k = \sum_{j=1}^{L} (\alpha_j + \beta\gamma_j) G_{kj} + U_k' + \epsilon_k'^Y \tag{3}$$

$$= \sum_{j=1}^{L} \Gamma_j G_{kj} + U_k' + \epsilon_k'^Y \tag{4}$$

where $U_k' = (1+\beta)U_k$ and $\epsilon_k'^Y = \epsilon_k^Y + \beta\epsilon_k^X$. When the exclusion restriction is satisfied ($\alpha_j = 0$), i.e. the genetic variant does not have an effect on Y not through X. We can use the Wald ratio to estimate the causal effect $\beta = \dfrac{\beta\gamma_j}{\gamma_j} = \dfrac{\Gamma_j}{\gamma_j}$. However, if $\alpha_j \neq 0$, the causal effect estimate will be biased; $\beta_j = \beta + \dfrac{\alpha_j}{\gamma_j}$.

Now consider a two-sample summary data MR analysis. The first study provides L estimates of G-X associations, $\hat{\gamma}_j$ with standard error σ_{Xj}. The second study provides L estimates of G-Y associations, $\hat{\Gamma}_j$ with standard error σ_{Yj}. Specifically, we assume that

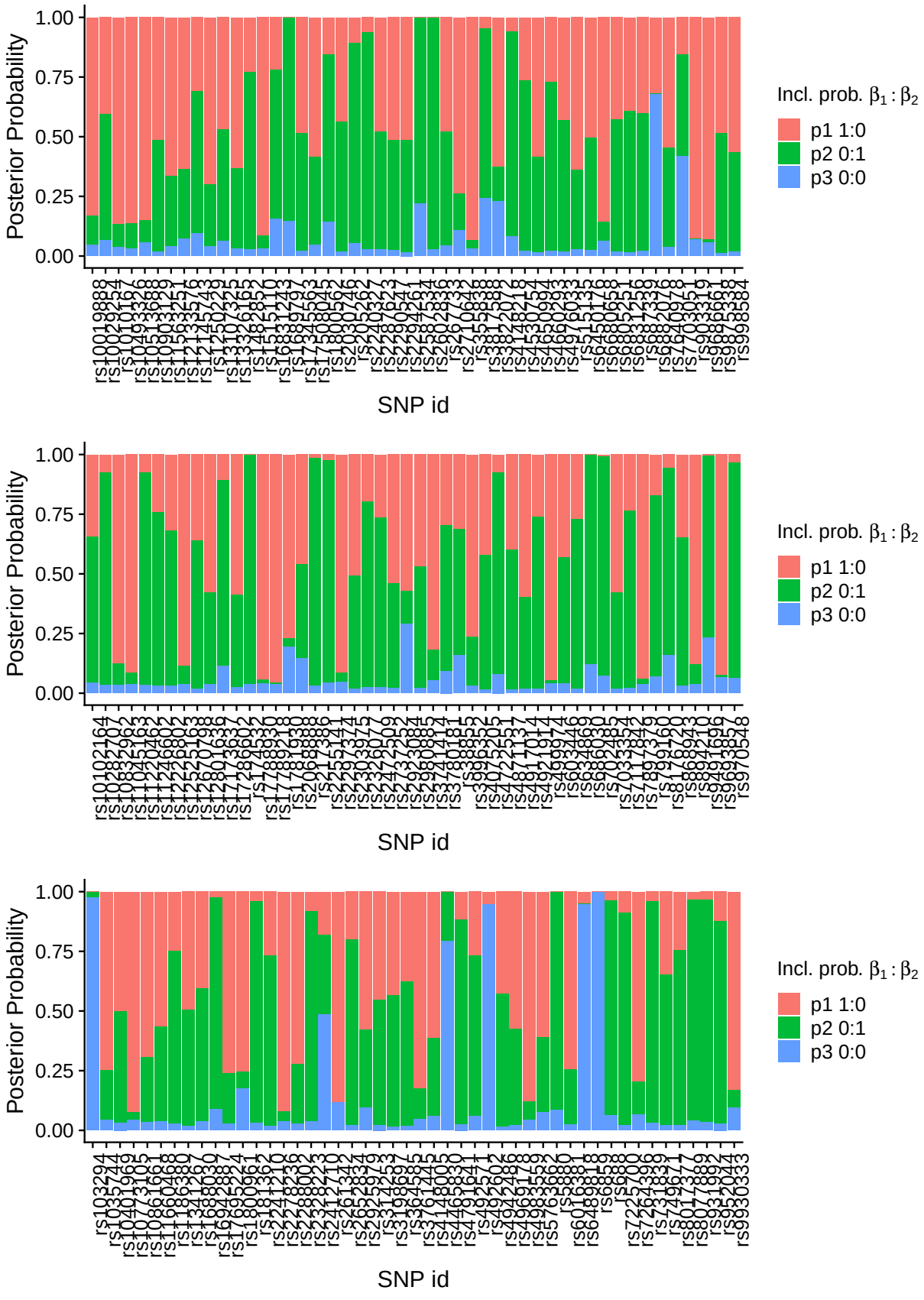

Figure 7: DL estimation instrument inclusion posterior probaility for violation of InSIDE

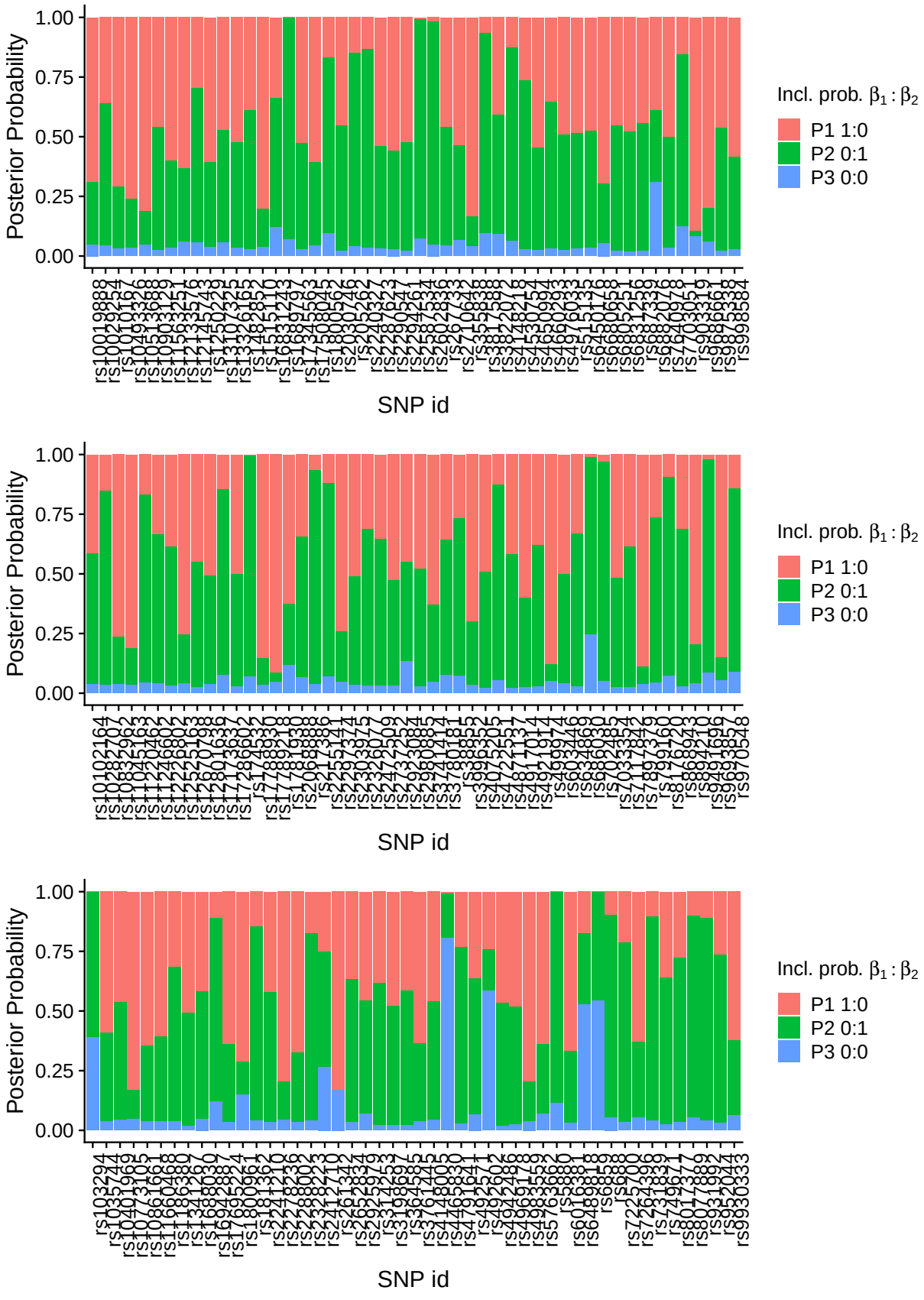

Figure 8: Full Bayesian instrument inclusion posterior probaility for violation of InSIDE

$$\hat{\Gamma}_j = \alpha_j + \beta\hat{\gamma}_j + \sigma_{Yj}\epsilon_j \tag{5}$$

where $\alpha_j \sim N(\mu_\alpha, \tau^2)$ and $\epsilon_j \sim N(0,1)$. We assume the InSIDE assumption is satisfied, i.e. γ_j is independent of α_j. And we also assume that σ_{Xj} and σ_{Yj} are fixed and known quantities.

Bayesian model averaging

We wish to use Bayesian model averaging to search the space of all possible models defined by each of the 2^L subsets in the data. Let I_L represent the vector of prior probabilities of G inclusion, our parameters of interest is then $\theta = (\beta, \tau^2, I)$ and the data, D, will contain the estimates $\hat{\gamma}_j$ and $\hat{\Gamma}_j$ and, their standard errors σ_{Xj} and σ_{Yj}. Then the joint posterior is

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{6}$$

where $P(D|\theta)$ is the likelihood and $P(\theta)$ is user specified prior for each of the parameters.

Robust profile likelihood

For a likelihood that is efficient with pleiotropic and weak instrument bias, we adapted the profile likelihood of (Zhao et al. 2018); we can express the likelihood for (β, τ^2) given the data $(\hat{\gamma}, \hat{\Gamma})$ profiled over the parameters $\gamma_1, ..., \gamma_L$ as

$$\begin{aligned} l(\beta, \tau^2|\hat{\gamma}, \hat{\Gamma}) &= \text{Max}_{\boldsymbol{\gamma}} l(\beta, \tau^2|\hat{\gamma}, \hat{\Gamma}) \\ &= \log f(\hat{\gamma}, \hat{\Gamma}|\beta, \tau^2) \\ &= -\frac{L}{2}log(2\pi) - \frac{1}{2}\sum_{j=1}^{L}\left\{log(\sigma_{Yj}^2 + \tau^2) + \left(\frac{(\hat{\Gamma}_j - \beta\hat{\gamma}_j)^2}{\beta^2\sigma_{Xj}^2 + \sigma_{Yj}^2 + \tau^2}\right)\right\} \end{aligned}$$

To include a vector of G inclusion, I, the likelihood is modified to

$$\begin{aligned} l(\beta, \tau^2, I|\hat{\gamma}, \hat{\Gamma}) =& \text{Max}_{\boldsymbol{\gamma}} l(\beta, \tau^2|\hat{\gamma}, \hat{\Gamma}) \\ =& \log f(\hat{\gamma}, \hat{\Gamma}|\beta, \tau^2) \\ =& -\frac{\sum_{j=1}^{L} I_j}{2}log(2\pi) \\ & - \frac{1}{2}\sum_{j=1}^{L} I_j\left\{log(\sigma_{Yj}^2 + \tau^2) + \left(\frac{(\hat{\Gamma}_j - \beta\hat{\gamma}_j)^2}{\beta^2\sigma_{Xj}^2 + \sigma_{Yj}^2 + \tau^2}\right)\right\} \end{aligned} \tag{7}$$

Note that this likelihood is an increasing function of number of instruments included and decreasing function of increasing τ^2. Hence, this likelihood will favour the largest sets of instruments with homogeneous causal effect estimates, satisfying the consistency criterion (Kang et al. 2016) or ZEro Modal Pleiotropy Assumption (Hartwig, Davey Smith, and Bowden 2017).

Metropolis-Hastings algorithm

We will use random walk Metropolis-Hastings (MH) algorithm to sequentially update the parameter values. Unlike Gibbs sampling, the MH algorithm does not directly sample from the conditional posterior distribution, but instead it requires a proposal distribution for each parameter. Let $\theta_i = (\beta_i, \tau_i^2, I_i)$ be the current ith value of the parameter vector θ. $P(\theta)$ as the joint posterior density of θ. θ_i is updated to θ_{i+1}, by first simulating a candidate value θ^* from proposal density, then calculate the acceptance probability (*prob*), which is the probability of whether θ^* is accepted as θ_{i+1}. Note that if the proposal density for a given parameter is 'symmetric' - that is if $C(\theta_i|\theta_{i+1}) = C(\theta_{i+1}|\theta_i)$ then the proposal density can be omitted from the calculation of *prob*, such is the case for β and I. Here we describe the detail of MH algorithm for Full Bayesian and DL approach:

Full Bayesian approach

- **Update** β
  1. Sample $\beta^* \sim \beta_i + h_\beta \times N(0,1)$, where h_β is a user defined tuning constant.
  2. Accept $\beta_{i+1} = \beta^*$ with probability:
  $$prob = \min\left\{1, \frac{P(\beta^*, \tau_i^2, I_i)}{P(\beta_i, \tau_i^2, I_i)}\right\}$$
  otherwise set $\beta_{i+1} = \beta_i$.
- **Update** $Prec$ **(**$\tau^2 = 1/Prec$**)**
  1. Sample
  $$\begin{aligned} Prec^* &\sim U(\mathrm{LB}_{Prec^*}, \mathrm{UB}_{Prec^*}) \\ \mathrm{LB}_{Prec^*} &= \max(\mathrm{LL}, Prec_i - h_{Prec}) \\ \mathrm{UB}_{Prec^*} &= \min(\mathrm{UL}, Prec_i + h_{Prec}) \end{aligned}$$
  where LL and UL is user defined lower and upper limit for $Prec$ respectively, and h_{Prec} is a user defined tuning constant.
  2. Accept $Prec_{i+1} = Prec^*$ with probability:
  $$prob = \min\left\{1, \frac{U(\mathrm{LB}_{Prec_i}, \mathrm{UB}_{Prec_i})P(\beta_{i+1}, Prec^*, I_i)}{U(\mathrm{LB}_{Prec^*}, \mathrm{UB}_{Prec^*})P(\beta_{i+1}, Prec_i, I_i)}\right\}$$
  where U(,) is the proposal density. Otherwise set $Prec_{i+1} = Prec_i$.
- **Update** I
  1. Generate a random number between 1 and L, define it as I_q^*, which is the qth element of I^*
  2. Set $I_d^* = I_{id}$ for all $d \neq q$.
  3. Set $I_q^* = (I_{iq} - 1)^2$ (this defines the proposed and current model to differ by the inclusion or the exclusion of one instrument.)
  4. Check if $\sum_{j=1}^{L} I_j \geq 5$ is true, otherwise start from beginning again where another random number is generated.
  5. Accept $I_{i+1} = I^*$ with probability:
  $$prob = \min\left\{1, \frac{P(\beta_{i+1}, \tau_{i+1}^2, I^*)}{P(\beta_{i+1}, \tau_{i+1}^2, I_i)}\right\}$$
  otherwise set $I_{i+1} = I_i$.

DerSimonian-Laird (DL) approach

- **Update** β
  1. Sample $\beta^* \sim \beta_i + h_\beta \times N(0,1)$, where h_β is a user defined tuning constant.
  2. Accept $\beta_{i+1} = \beta^*$ with probability:
  $$prob = \min\left\{1, \frac{P(\beta^*, \hat{\tau}_i^2, I_i)}{P(\beta_i, \hat{\tau}_i^2, I_i)}\right\}$$
  where $\hat{\tau}^2$ is estimated from DL heterogeneity statistics [@DerSimonian1986]. Otherwise set $\beta_{i+1} = \beta_i$.
- **Update** L
  1. Generate a random number between 1 and L, define it as I_q^*, which is the qth element of I^*
  2. Set $I_d^* = I_{id}$ for all $d \neq q$.
  3. Set $I_q^* = (I_{iq} - 1)^2$
  4. Check if $\sum_{j=1}^{L} I_j \geq 10$ is true, otherwise start from beginning again where another random number is generated.
  5. Accept $I_{i+1} = I^*$ with probability:
  $$prob = \min\left\{1, \frac{P(\beta_{i+1}, \hat{\tau}^{2*}, I^*)}{P(\beta_{i+1}, \hat{\tau}_i^2, I_i)}\right\}$$
  where $\hat{\tau}^{2*}$ is estimated from DL heterogeneity statistics [@DerSimonian1986] with I^*. Otherwise set $I_{i+1} = I_i$.

h_β and h_{Prec} acts as tuning parameters for acceptance rate, i.e. proportion of iterations that θ^* has been accepted as θ_{i+1}. Acceptance rate have been recommended to be between 0.25 and 0.45 for random walk MH algorithm (Albert 2009).

Violation of InSIDE assumption

In this section we show how, by invoking two sets of SNPs which either collectively violate or satisfy the InSIDE assumption, it is possible induce two sample summary data with two distinct clusters.

Suppose that we have data from a Mendelian randomisation study, where it consists of N individuals, each subject k has L independent genetic variants (G_{kj}), an exposure (X_k) and an outcome (Y_k) of interest. We assume the confounders (U) between X and Y are unknown. We consider the case where balanced pleiotropy exists and the instruments are allowed to potentially violate the InSIDE assumption. Specifically, violation of InSIDE will be induced whenever a SNP is associated with an outcome through a variable which is a common cause of X and Y, which is denote by the parameter ψ. We assume the following linear models for U, X and Y:

$$U_k|G_k = \sum_{j=1}^{L} \psi_j G_{kj} + \epsilon_k^U \tag{8}$$

$$X_k|U_k, G_k = \sum_{j=1}^{L} \delta_j G_{kj} + U_k + \epsilon_k^X \tag{9}$$

$$Y_k|X_k, U_k, G_k = \sum_{j=1}^{L} \Delta_j G_{kj} + \beta X_k + U_k + \epsilon_k^Y \tag{10}$$

where ϵ_k^U, ϵ_k^X and ϵ_k^Y are independent error term for U, X and Y respectively. Substituting (8) into (9) and (10) gives:

$$X_k|U_k, G_k = \sum_{j=1}^{L}(\delta_j + \psi_j)G_{kj} + \epsilon_k'^X \tag{11}$$

$$Y_k|X_k, U_k, G_k = \sum_{j=1}^{L}(\Delta_j + \psi_j)G_{kj} + \beta X_k + \epsilon_k'^Y \tag{12}$$

where $\epsilon_k'^X = \epsilon_k^X + \epsilon_k^U$ and $\epsilon_k'^Y = \epsilon_k^Y + \epsilon_k^U$. Substituting (11) into (12) gives the *reduced-form* equation of Y

$$Y_k|X_k, U_k, G_k = \sum_{j=1}^{L}\Big[\Delta_j + \psi_j + \beta(\delta_j + \psi_j)\Big]G_{kj} + \epsilon_k''^Y \tag{13}$$

where $\epsilon_k''^Y = \epsilon_k'^Y + \beta\epsilon_k'^X$.

As we are focusing on the case where only summary statistics of SNP-exposure and SNP-outcome are available, therefore we need to marginalise over (U, G_{-l}) and (X, U, G_{-l}) in (11) and (13) to give $X|G_l$ and $Y|G_l$ respectively

$$X_k|G_{kl} = (\delta_l + \psi_l)G_{kl} + \epsilon_{kl}''^X \tag{14}$$

$$\text{where} \quad \epsilon_{kl}''^X = \sum_{j\neq l}^{L}(\delta_j + \psi_j)G_{kj} + \epsilon_k'^X$$

$$Y_k|G_{kl} = \Big[\Delta_l + \psi_l + \beta(\delta_l + \psi_l)\Big]G_{kl} + \epsilon_{kl}'''^Y \tag{15}$$

$$\text{where} \quad \epsilon_{kl}'''^Y = \sum_{j\neq l}^{L}\Big[\Delta_j + \psi_j + \beta(\delta_j + \psi_j)\Big]G_{kj} + \epsilon_k''^Y$$

Now consider a two-sample summary data Mendelian randomization analysis. A first study provides estimates of L SNP-exposure associations, $\hat{\gamma}_j$ with standard error σ_{Xj}. The second study provides estimates of L SNP-outcome associations, $\hat{\Gamma}_j$ with standard error σ_{Yj}. Specifically, we assume that

$$\hat{\Gamma}_j = \alpha_j + \beta\gamma_j + \sigma_{Yj}\epsilon_j \tag{16}$$

where $\alpha_j = \Delta_j + \psi_j$, $\gamma_j = \delta_j + \psi_j$ and $\epsilon \sim N(0, 1)$. Throughout we assume that σ_{Xj} and σ_{Yj} are fixed and known quantities. The individual Wald ratio causal effect estimand:

$$\beta_j = \frac{\Gamma_j}{\gamma_j} = \beta + \frac{\alpha_j}{\gamma_j} = \beta + \frac{\Delta_j + \psi_j}{\delta_j + \psi_j}$$

If either $\Delta_j \neq 0$ or $\psi_j \neq 0$ then an instrument is strictly invalid due to pleiotropy.

Suppose we have two different groups of invalid instruments: in the first group we have $\psi_j = 0$ for all SNPs and $\bar{\Delta} = 0$. In words, the pleiotropy is balanced pleiotropy and satisfies the InSIDE assumption. That is, the corresponding SNP-outcome model for this set is still equation 16, but $\alpha_j \sim N(0, \tau^2)$ and $Cov(\alpha_j, \gamma_j) = 0$. This means that we can consistently estimate β from this set. Suppose now that for the remaining group of

instruments, $\delta_j = 0$ and $\Delta_j = 0$.This means that their total pleiotropic effects is via a confounder of X and Y and the InSIDE assumption is perfectly violated. The following model then describes these data

$$\hat{\Gamma}_j = 0 + \beta^* \gamma_j + \sigma_{Yj} \epsilon_j \tag{17}$$

where $\beta^* = \beta + 1$. It is clear that this set of SNPs identify a distinct, biased version of the causal effect.

In the general case where the total set of SNPs can be partitioned into a set satisfying InSIDE (S_1) and a set violating InSIDE (S_2), we could allow Δ_j, δ_j and ψ_j to all be non-zero in S_2. Although InSIDE would not be perfectly violated in S_2 we would still see two clusters in the data:

$$j \in S_1 : \alpha_j + \beta \gamma_j + \sigma_{Yj} \epsilon_j, \quad \alpha_j \sim N(0, \tau_1^2) \tag{18}$$

$$j \in S_2 : \alpha_j + \beta^* \gamma_j + \sigma_{Yj} \epsilon_j, \quad \alpha_j \sim N(0, \tau_2^2) \tag{19}$$

Modified method

In order to allow for InSIDE violation, we will modify the likelihood and the Metropolis-Hastings algorithm seen in the previous section.

The adopted likelihood is a function of four model parameters with two indicator functions, $\theta = (\beta_1, \tau_1^2, \beta_2, \tau_2^2, I_1, I_2)$ and a combination link function, I_{12j}:

$$\begin{aligned}
l(\theta|\hat{\gamma}, \hat{\Gamma}) =& \text{Max}_{\boldsymbol{\gamma}} l(\beta_1, \tau_1^2, \beta_2, \tau_2^2 | \hat{\gamma}, \hat{\Gamma}) \\
=& \log f(\hat{\gamma}, \hat{\Gamma} | \beta_1, \tau_1^2, \beta_2, \tau_2^2) \\
=& -\frac{\sum_{j=1}^{L} I_{12j}}{2} log(2\pi) \\
& - \frac{1}{2} \sum_{j=1}^{L} I_{12j} \left\{ log(\sigma_{Yj}^2 + \tau_1^2) + \left(\frac{(\hat{\Gamma}_j - \beta_1 \hat{\gamma}_j)^2}{\beta_1^2 \sigma_{Xj}^2 + \sigma_{Yj}^2 + \tau_1^2} \right) \right\} \\
& - \frac{\sum_{j=1}^{L} (1 - I_{12j})}{2} log(2\pi) \\
& - \frac{1}{2} \sum_{j=1}^{L} (1 - I_{12j}) \left\{ log(\sigma_{Yj}^2 + \tau_2^2) + \left(\frac{(\hat{\Gamma}_j - \beta_2 \hat{\gamma}_j)^2}{\beta_2^2 \sigma_{Xj}^2 + \sigma_{Yj}^2 + \tau_2^2} \right) \right\}
\end{aligned} \tag{20}$$

where

$$I_{12j} = \begin{cases} 1 & \text{if either } I_1 \text{or } I_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

The above procedure can therefore include SNPs in the analysis by assigning them to: S_1 only if $I_{1j} = 1, I_{2j} = 0$; S_2 only if $I_{1j} = 0, I_{2j} = 1$; neither S_1 or S_2 if $(I_{1j} = 0, I_{2j} = 0)$ or $(I_{1j} = 1, I_{2j} = 1)$, if they are plausible candidates for neither or both respectively. For the latter case, we consider $(I_{1j} = 0, I_{2j} = 0)$ to be equivalent to $(I_{1j} = 1, I_{2j} = 1)$, therefore we restrict the Metropolis-Hastings jump to only $(I_{1j} = 0, I_{2j} = 0)$, we will discuss this further in the next section.

The Metropolis-Hastings (MH) algorithm remains the same as in Section , except instead of the parameter space of $\theta = (\beta, \tau^2, I)$ we have $\theta = (\beta_1, \tau_1^2, \beta_2, \tau_2^2, I_1, I_2)$. They will be updated the same way as when the algorithm was estimating for a single causal effect.

- **Update** I_1
  1. Generate a random number between 1 and L, define it as I^*_{1q}, which is the qth element of I^*_1
  2. Set $I^*_{1d} = I_{i1d}$ for all $d \neq q$, if $I_{i2q} \neq 1$, otherwise repeat step 1.
  3. Set $I^*_{1q} = (I_{i1q} - 1)^2$.
  4. If $\sum_{j=1}^{L} I_{1j} \geq 5$ is true, proceed to next step, otherwise start from step again where another random number is generated.
  5. Accept $I_{1i+1} = I^*_1$ with probability:

$$prob = \min\left\{1, \frac{P(\beta_{1i+1}, \tau^2_{1i+1}, \beta_{2i+1}, \tau^2_{2i+1}, I^*_1, I_{2i})}{P(\beta_{1i+1}, \tau^2_{1i+1}, \beta_{2i+1}, \tau^2_{2i+1}, I_{1i}, I_{2i})}\right\}$$

  otherwise set $I_{1i+1} = I_{1i}$.

- **Update** I_2
  1. Generate a random number between 1 and L, define it as I^*_{2q}, which is the qth element of I^*_2
  2. Set $I^*_{2d} = I_{i2d}$ for all $d \neq q$, if $I_{i1q} \neq 1$, otherwise repeat step 1.
  3. Set $I^*_{2q} = (I_{i2q} - 1)^2$.
  4. If $\sum_{j=1}^{L} I_{2j} \geq 5$ is true, proceed to next step, otherwise start from step 1 again where another random number is generated.
  5. Accept $I_{2i+1} = I^*_2$ with probability:

$$prob = \min\left\{1, \frac{P(\beta_{1i+1}, \tau^2_{1i+1}, \beta_{2i+1}, \tau^2_{2i+1}, I_{1i+1}, I^*_2)}{P(\beta_{1i+1}, \tau^2_{1i+1}, \beta_{2i+1}, \tau^2_{2i+1}, I_{1i+1}, I_{2i})}\right\}$$

  otherwise set $I_{2i+1} = I_{2i}$.

Step 2 in **Update** I_1 and **Update** I_2 restricts the new jump to be conditional on I_2 and I_1 respectively, this will stop the case of $(I_{1j} = 1, I_{2j} = 1)$. The reason for this restriction is that model space including both $(I_{1j} = 1, I_{2j} = 1)$ and $(I_{1j} = 0, I_{2j} = 0)$ is equivalent to giving model that consists of outlying instruments higher probability then models where instruments have to be designated to either I_1 or I_2.

Software

Code for analysis

Data

```
# loading all the BMA functions see section 'Functions for
# BMA'
source(paste(functdir, "Functions_BMA_SumSat_v9.R", sep = ""))
library(mr.raps)
library("MRChallenge2019")

data_amd <- Challenge_dat[, c("rsid", "beta_XL.HDL.C", "p_XL.HDL.C",
    "beta_amd", "se_amd")]
data_amd_complete <- na.omit(data_amd)

data_amd_complete[, "se_XL.HDL.C"] <- (-1) * (data_amd_complete$beta_XL.HDL.C/qnorm(data_amd_complete$p

plot(data_amd_complete$beta_XL.HDL.C, data_amd_complete$beta_amd,
```

```
    ylim = c(-0.3, 0.2), xlim = c(-0.3, 0.2), pch = 16)
arrows(data_amd_complete$beta_XL.HDL.C, data_amd_complete$beta_amd -
    (data_amd_complete$se_amd * 1.96), data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd + (data_amd_complete$se_amd *
        1.96), length = 0, angle = 90, code = 3, lty = 2)
arrows(data_amd_complete$beta_XL.HDL.C - (data_amd_complete$se_XL.HDL.C *
    1.96), data_amd_complete$beta_amd, data_amd_complete$beta_XL.HDL.C +
    (data_amd_complete$se_XL.HDL.C * 1.96), data_amd_complete$beta_amd,
    length = 0, angle = 90, code = 3, lty = 2)
```

Analysis

```
# BMA: DL estimate
# ------------------------------------------------------------------------
nIter <- 5e+05  #number of iterations
burnin <- 1e+05  #number of burn-ins
L <- nrow(data_amd_complete)

# Prior choice for beta, tau and inclusion of instruments
Ins_prior <- rep(0.5, L)
Prior_DL <- list(hyper_Beta_mean = 0, hyper_Beta_sd = 1, Ins_prob = Ins_prior)

# Tuning parameter for beta
H_DL <- list(Beta = 0.5)

# Generate initial values
gen_inits_DL <- list(Beta = rnorm(1, 0, 10), Ins_L = randomS.initial.LI(rep(0,
    L), L, Ins_prior))

# M-H algorithm
res_DL <- BMA_MRanalysis("DL_approx", data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd, data_amd_complete$se_XL.HDL.C,
    data_amd_complete$se_amd, L, nIter, Prior_DL, H_DL, gen_inits_DL)

# BMA: Full Bayesian
# ------------------------------------------------------------------------
nIter <- 5e+05  #number of iterations
burnin <- 1e+05  #number of burn-ins
L <- nrow(data_amd_complete)

# Prior choice for beta, tau and inclusion of instruments
Ins_prior <- rep(0.5, L)
Prior_gamma <- list(hyper_Beta_mean = 0, hyper_Beta_sd = 1, hyper_Prec_shape = 2,
    hyper_Prec_rate = 5e-05, Ins_prob = Ins_prior)

# Tuning parameter for beta
H_gamma <- list(Beta = 0.5, Prec_LL = 0, Prec_UL = 1e+06, Prec_gap = 150000)

# Generate initial values
gen_inits_gamma <- list(Beta = rnorm(1, 0, 10), UBPrec = 1e+06,
    LBPrec = 0, Ins_L = randomS.initial.LI(rep(0, L), L, Ins_prior))

# M-H algorithm
```

```
res_gamma <- BMA_MRanalysis("Full_Bayes", data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd, data_amd_complete$se_XL.HDL.C,
    data_amd_complete$se_amd, L, nIter, Prior_gamma, H_gamma,
    gen_inits_gamma)

# BMA with invalid InSIDE assumption: DL estimate
# ------------------------------------------------------------------------
nIter <- 5e+05  #number of iterations
burnin <- 1e+05  #number of burn-ins
L <- nrow(data_amd_complete)

# Prior choice for beta, tau and inclusion of instruments
Ins1_prior <- rep(0.5, L)
Ins2_prior <- rep(0.5, L)
Prior_DL <- list(hyper_Beta1_mean = 0, hyper_Beta1_sd = 1, Ins1_prob = Ins1_prior,
    Ins2_prob = Ins2_prior, hyper_Beta2_mean = 0, hyper_Beta2_sd = 1)

# Tuning parameter for beta
H_DL <- list(Beta1_h = 0.8, Beta2_h = 0.8)

# Generate initial values
gen_inits_DL <- list(Beta1 = rnorm(1, 0, 10), Beta2 = rnorm(1,
    0, 10), Ins_L1 = randomS.initial.LI(rep(0, L), L, Ins1_prior),
    Ins_L2 = randomS.initial.LI(rep(0, L), L, Ins2_prior))

# M-H algorithm
res_DL <- BMA_MRanalysis("DL_approx", 2, 1, data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd, data_amd_complete$se_XL.HDL.C,
    data_amd_complete$se_amd, L, nIter, Prior_DL, H_DL, gen_inits_DL)

# BMA with invalid InSIDE assumption: Full Bayesian
# ------------------------------------------------------------------------
nIter <- 5e+05  #number of iterations
burnin <- 1e+05  #number of burn-ins
L <- nrow(data_amd_complete)

# Prior choice for beta, tau and inclusion of instruments
Ins1_prior <- rep(0.5, L)
Ins2_prior <- rep(0.5, L)
Prior_gamma <- list(hyper_Beta1_mean = 0, hyper_Beta1_sd = 1,
    hyper_Prec1_shape = 2, hyper_Prec1_rate = 5e-05, Ins1_prob = Ins1_prior,
    hyper_Beta2_mean = 0, hyper_Beta2_sd = 1, hyper_Prec2_shape = 2,
    hyper_Prec2_rate = 5e-05, Ins2_prob = Ins2_prior)

# Tuning parameter for beta
H_gamma <- list(Beta1_h = 0.8, Beta2_h = 0.8, Prec1_LL = 0, Prec1_UL = 1e+06,
    Prec1_gap = 150000, Prec2_LL = 0, Prec2_UL = 1e+06, Prec2_gap = 150000)

# Generate initial values
gen_inits_gamma <- list(Beta1 = rnorm(1, 0, 10), UBPrec1 = 1e+06,
    LBPrec1 = 0, Ins_L1 = randomS.initial.LI(rep(0, L), L, Ins1_prior),
    Beta2 = rnorm(1, 0, 10), UBPrec2 = 1e+06, LBPrec2 = 0, Ins_L2 = randomS.initial.LI(rep(0,
        L), L, Ins2_prior))
```

```
# M-H algorithm
res_gamma <- BMA_MRanalysis("Full_Bayes", 2, 1, data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd, data_amd_complete$se_XL.HDL.C,
    data_amd_complete$se_amd, L, nIter, Prior_gamma, H_gamma,
    gen_inits_gamma)

# MR-RAPS
# ------------------------------------------------------------------------
res_mr_raps <- mr.raps.all(data_amd_complete$beta_XL.HDL.C, data_amd_complete$beta_amd,
    data_amd_complete$se_XL.HDL.C, data_amd_complete$se_amd)

# IVW
# ------------------------------------------------------------------------
Analysis = weightedIVW_NoPlot(data_amd_complete$beta_XL.HDL.C,
    data_amd_complete$beta_amd, data_amd_complete$se_XL.HDL.C,
    data_amd_complete$se_amd, tol = 1e-05)
```

Summary of results

```
library(metafor)

# Valid InSIDE assumption
# ------------------------------------------------------------------------
summary_res <- matrix(0, 5, 4)
summary_res[1, ] <- c(Analysis$RESULTS["1st order", "Est"], Analysis$RESULTS["1st order",
    "Est"], Analysis$RESULTS["1st order", "CI_L"], Analysis$RESULTS["1st order",
    "CI_U"])

summary_res[2, ] <- beta1_DL
summary_res[3, ] <- beta1_gamma

mr_raps_beta <- res_mr_raps[which(res_mr_raps$over.dispersion ==
    "TRUE" & res_mr_raps$loss.function == "l2"), "beta.hat"]
mr_raps_se <- res_mr_raps[which(res_mr_raps$over.dispersion ==
    "TRUE" & res_mr_raps$loss.function == "l2"), "beta.se"]
summary_res[4, ] <- c(mr_raps_beta, mr_raps_beta, mr_raps_beta -
    (1.96 * mr_raps_se), mr_raps_beta + (1.96 * mr_raps_se))

mr_raps_huber_beta <- res_mr_raps[which(res_mr_raps$over.dispersion ==
    "TRUE" & res_mr_raps$loss.function == "huber"), "beta.hat"]
mr_raps_huber_se <- res_mr_raps[which(res_mr_raps$over.dispersion ==
    "TRUE" & res_mr_raps$loss.function == "huber"), "beta.se"]
summary_res[5, ] <- c(mr_raps_huber_beta, mr_raps_huber_beta,
    mr_raps_huber_beta - (1.96 * mr_raps_huber_se), mr_raps_huber_beta +
        (1.96 * mr_raps_huber_se))

rownames(summary_res) <- c("IVW", "DL estimate", "Full Bayesian",
    "APS", "RAPS")
colnames(summary_res) <- c("Mean", "Median", "LI", "UI")

forest(summary_res[, "Mean"], ci.lb = summary_res[, "LI"], ci.ub = summary_res[,
    "UI"], slab = rownames(summary_res))
```

```r
# Invalid InSIDE assumption
# ------------------------------------------------------------------------
beta1_DL <- c(mean(res_DL$S[burnin:nIter, 1]), median(res_DL$S[burnin:nIter,
    1]), quantile(res_DL$S[burnin:nIter, 1], prob = c(0.025,
    0.975)))
beta2_DL <- c(mean(res_DL$S[burnin:nIter, 2]), median(res_DL$S[burnin:nIter,
    2]), quantile(res_DL$S[burnin:nIter, 2], prob = c(0.025,
    0.975)))

beta1_gamma <- c(mean(res_gamma$S[burnin:nIter, 1]), median(res_gamma$S[burnin:nIter,
    1]), quantile(res_gamma$S[burnin:nIter, 1], prob = c(0.025,
    0.975)))
beta2_gamma <- c(mean(res_gamma$S[burnin:nIter, 2]), median(res_gamma$S[burnin:nIter,
    2]), quantile(res_gamma$S[burnin:nIter, 2], prob = c(0.025,
    0.975)))

summary_res <- matrix(0, 4, 4)
summary_res[1, ] <- beta1_DL
summary_res[2, ] <- beta1_gamma
summary_res[3, ] <- beta2_DL
summary_res[4, ] <- beta2_gamma

rownames(summary_res) <- c("DL estimate Beta1", "Full Bayesian Beta1",
    "DL estimate Beta2", "Full Bayesian Beta2")
colnames(summary_res) <- c("Mean", "Median", "LI", "UI")

forest(summary_res[, "Mean"], ci.lb = summary_res[, "LI"], ci.ub = summary_res[,
    "UI"], slab = c(expression(paste("DL estimate ", beta[1])),
    expression(paste("Full Bayesian ", beta[1])), expression(paste("DL estimate ",
        beta[2])), expression(paste("Full Bayesian ", beta[2]))))

# Instrument Posterior Probability for DL estimate
p1_sub <- ifelse(res_DL$S[burnin:nIter, 5:(L + 4)] == 1 & res_DL$S[burnin:nIter,
    (L + 5):ncol(res_DL$S)] == 0, 1, 0)
p1 <- apply(p1_sub, 2, mean)
p1_data <- cbind(data_amd_complete$rsid, rep("p1 1:0", length(p1)),
    p1)

p2_sub <- ifelse(res_DL$S[burnin:nIter, 5:(L + 4)] == 0 & res_DL$S[burnin:nIter,
    (L + 5):ncol(res_DL$S)] == 1, 1, 0)
p2 <- apply(p2_sub, 2, mean)
p2_data <- cbind(data_amd_complete$rsid, rep("p2 0:1", length(p2)),
    p2)

p4_sub <- ifelse(res_DL$S[burnin:nIter, 5:(L + 4)] == 0 & res_DL$S[burnin:nIter,
    (L + 5):ncol(res_DL$S)] == 0, 1, 0)
p4 <- apply(p4_sub, 2, mean)
p4_data <- cbind(data_amd_complete$rsid, rep("p3 0:0", length(p4)),
    p4)

prob_ins_all1 <- as.data.frame(rbind(p1_data[1:50, ], p2_data[1:50,
    ], p4_data[1:50, ]))
```

```
colnames(prob_ins_all1) = c("rsid", "prob", "prob_value")
prob_ins_all1$prob_value <- as.numeric(as.character(prob_ins_all1$prob_value))

prob_ins_all2 <- as.data.frame(rbind(p1_data[51:100, ], p2_data[51:100,
    ], p4_data[51:100, ]))
colnames(prob_ins_all2) = c("rsid", "prob", "prob_value")
prob_ins_all2$prob_value <- as.numeric(as.character(prob_ins_all2$prob_value))

prob_ins_all3 <- as.data.frame(rbind(p1_data[101:nrow(p1_data),
    ], p2_data[101:nrow(p2_data), ], p4_data[101:nrow(p4_data),
    ]))
colnames(prob_ins_all3) = c("rsid", "prob", "prob_value")
prob_ins_all3$prob_value <- as.numeric(as.character(prob_ins_all3$prob_value))

SNP_group1 <- ggplot(data = prob_ins_all1, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
        beta[1]:beta[2])))

SNP_group2 <- ggplot(data = prob_ins_all2, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
        beta[1]:beta[2])))

SNP_group3 <- ggplot(data = prob_ins_all3, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
        beta[1]:beta[2])))

plot_grid(SNP_group1, SNP_group2, SNP_group3, nrow = 3)

# Instrument Posterior Probability for full Bayes
p1_sub <- ifelse(res_gamma$S[burnin:nIter, 5:(L + 4)] == 1 &
    res_gamma$S[burnin:nIter, (L + 5):ncol(res_gamma$S)] == 0,
    1, 0)
p1 <- apply(p1_sub, 2, mean)
p1_data <- cbind(data_amd_complete$rsid, rep("P1 1:0", length(p1)),
    p1)

p2_sub <- ifelse(res_gamma$S[burnin:nIter, 5:(L + 4)] == 0 &
    res_gamma$S[burnin:nIter, (L + 5):ncol(res_gamma$S)] == 1,
```

```
    1, 0)
p2 <- apply(p2_sub, 2, mean)
p2_data <- cbind(data_amd_complete$rsid, rep("P2 0:1", length(p2)),
    p2)

p4_sub <- ifelse(res_gamma$S[burnin:nIter, 5:(L + 4)] == 0 &
    res_gamma$S[burnin:nIter, (L + 5):ncol(res_gamma$S)] == 0,
    1, 0)
p4 <- apply(p4_sub, 2, mean)
p4_data <- cbind(data_amd_complete$rsid, rep("P3 0:0", length(p4)),
    p4)

prob_ins_all1 <- as.data.frame(rbind(p1_data[1:50, ], p2_data[1:50,
    ], p4_data[1:50, ]))
colnames(prob_ins_all1) = c("rsid", "prob", "prob_value")
prob_ins_all1$prob_value <- as.numeric(as.character(prob_ins_all1$prob_value))

prob_ins_all2 <- as.data.frame(rbind(p1_data[51:100, ], p2_data[51:100,
    ], p4_data[51:100, ]))
colnames(prob_ins_all2) = c("rsid", "prob", "prob_value")
prob_ins_all2$prob_value <- as.numeric(as.character(prob_ins_all2$prob_value))

prob_ins_all3 <- as.data.frame(rbind(p1_data[101:nrow(p1_data),
    ], p2_data[101:nrow(p2_data), ], p4_data[101:nrow(p4_data),
    ]))
colnames(prob_ins_all3) = c("rsid", "prob", "prob_value")
prob_ins_all3$prob_value <- as.numeric(as.character(prob_ins_all3$prob_value))

SNP_group1 <- ggplot(data = prob_ins_all1, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
        beta[1]:beta[2])))

SNP_group2 <- ggplot(data = prob_ins_all2, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
        beta[1]:beta[2])))

SNP_group3 <- ggplot(data = prob_ins_all3, aes(x = rsid, y = prob_value,
    fill = prob)) + geom_bar(stat = "identity") + ylab("Posterior Probability") +
    xlab("SNP id") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, size = 12), plot.title = element_text(size = 16),
    axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14),
    legend.title = element_text(size = 12), legend.text = element_text(size = 12)) +
    scale_fill_discrete(name = expression(paste("Incl. prob. ",
```

```
        beta[1]:beta[2])))

plot_grid(SNP_group1, SNP_group2, SNP_group3, nrow = 3)
```

Functions for BMA

Please note the following function is for when InSIDE is valid, for the function of invalid InSIDE is still under-construction.

```
### randomS.initial.LI()###

# Generate random initial model space that doesn't produce
# empty and 1 variable model space

randomS.initial.LI <- function(Ind_L, L, ins_prior) {
    repeat {
        # do something
        Ind_L <- rbinom(L, 1, ins_prior)
        # exit if the condition is met
        if (sum(Ind_L) >= 5)
            break
    }
    return(Ind_L)
}

### BMA_MRanalysis()###

# User specified options
#'tau_estimate' Use DL estimate or Full Bayesian to analyse data
#'BetaXG'       Effect size for X-G association
#'BetaXG'       Effect size for Y-G association
#'seBetaXG'     Standard error for X-G association
#'seBetaYG'     Standard error for X-G association
#'N_Ins'        Number of genetic instruments
#'N_Iter'       Number of iterations
#'Prior'        Hyper parameter for the prior distribution
#'tuning_para'  Tuning parameter to ensure sufficient acceptance rate
#'gen_inits'    Initial values to start the iterations

BMA_MRanalysis <- function(tau_estimate, BetaXG, BetaYG, seBetaXG,
    seBetaYG, N_Ins, N_Iter, Prior, tuning_para, gen_inits) {

    if (tau_estimate == "DL_approx") {

        # Generate initial values
        Beta <- gen_inits$Beta
        Ins_L <- gen_inits$Ins_L

        # Saved space
        S <- matrix(0, N_Iter, N_Ins + 2)
        accept <- matrix(1, N_Iter, 2)

        for (i in 1:N_Iter) {
```

```
            oldBeta <- Beta
            oldIns_L <- Ins_L

            # 1st step: Compare likelihood for old and new Beta
            Beta <- Beta + rnorm(1, 0, tuning_para$Beta)
            oldlogpost <- logpost_DL(BetaXG, BetaYG, seBetaXG,
                seBetaYG, oldBeta, oldIns_L, Prior$hyper_Beta_mean,
                Prior$hyper_Beta_sd)
            newlogpost <- logpost_DL(BetaXG, BetaYG, seBetaXG,
                seBetaYG, Beta, oldIns_L, Prior$hyper_Beta_mean,
                Prior$hyper_Beta_sd)
            oldlp <- oldlogpost$logpost
            newlp <- newlogpost$logpost
            if (log(runif(1, 0, 1)) > (newlp - oldlp)) {
                Beta <- oldBeta  # fails so return to oldb
                accept[i, 1] <- 0
            }

            # 3nd step: Compare likelihood for old and new model space
            Ins_L <- randomS.LI(Ins_L, N_Ins, Prior$Ins_prob)
            oldlogpost <- logpost_DL(BetaXG, BetaYG, seBetaXG,
                seBetaYG, Beta, oldIns_L, Prior$hyper_Beta_mean,
                Prior$hyper_Beta_sd)
            newlogpost <- logpost_DL(BetaXG, BetaYG, seBetaXG,
                seBetaYG, Beta, Ins_L, Prior$hyper_Beta_mean,
                Prior$hyper_Beta_sd)

            tausq_hat <- newlogpost$tausq

            oldlp <- oldlogpost$logpost
            newlp <- newlogpost$logpost
            if (log(runif(1, 0, 1)) > (newlp - oldlp)) {
                Ins_L <- oldIns_L  # fails so return to oldIL
                tausq_hat <- oldlogpost$tausq
                accept[i, 2] <- 0
            }
            S[i, ] <- c(Beta, tausq_hat, Ins_L)
        }
        returnlist <- list(S = S, accept_rate = apply(accept,
            2, mean))
        return(returnlist)
    }

    if (tau_estimate == "Full_Bayes") {

        # Generate initial values
        Beta <- gen_inits$Beta
        UBPrec <- gen_inits$UBPrec
        LBPrec <- gen_inits$LBPrec
        Prec <- runif(1, LBPrec, UBPrec)
        Ins_L <- gen_inits$Ins_L

        # Saved space
```

```
S <- matrix(0, N_Iter, N_Ins + 2)
accept <- matrix(1, N_Iter, 3)

for (i in 1:N_Iter) {
    oldBeta <- Beta
    oldPrec <- Prec
    oldUBPrec <- UBPrec
    oldLBPrec <- LBPrec
    oldIns_L <- Ins_L

    # 1st step: Compare likelihood for old and new Beta
    Beta <- Beta + rnorm(1, 0, tuning_para$Beta)
    oldlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
        seBetaYG, oldBeta, oldPrec, oldIns_L, Prior$hyper_Beta_mean,
        Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
        Prior$hyper_Prec_rate)
    newlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
        seBetaYG, Beta, oldPrec, oldIns_L, Prior$hyper_Beta_mean,
        Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
        Prior$hyper_Prec_rate)
    oldlp <- oldlogpost + log(oldUBPrec - oldLBPrec)
    newlp <- newlogpost + log(oldUBPrec - oldLBPrec)
    if (log(runif(1, 0, 1)) > (newlp - oldlp)) {
        Beta <- oldBeta  # fails so return to oldb
        accept[i, 1] <- 0
    }

    # 2nd step: Compare likelihood for old and new tausq
    UBPrec <- min(tuning_para$Prec_UL, Prec + tuning_para$Prec_gap)
    LBPrec <- max(tuning_para$Prec_LL, Prec - tuning_para$Prec_gap)
    Prec <- runif(1, LBPrec, UBPrec)
    oldlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
        seBetaYG, Beta, oldPrec, oldIns_L, Prior$hyper_Beta_mean,
        Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
        Prior$hyper_Prec_rate)
    newlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
        seBetaYG, Beta, Prec, oldIns_L, Prior$hyper_Beta_mean,
        Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
        Prior$hyper_Prec_rate)
    oldlp <- oldlogpost + log(UBPrec - LBPrec)
    newlp <- newlogpost + log(oldUBPrec - oldLBPrec)
    if (log(runif(1, 0, 1)) > (newlp - oldlp)) {
        Prec <- oldPrec  # fails so return to oldPrecsq
        UBPrec <- oldUBPrec
        LBPrec <- oldLBPrec
        accept[i, 2] <- 0
    }

    # 3nd step: Compare likelihood for old and new model space
    Ins_L <- randomS.LI(Ins_L, N_Ins, Prior$Ins_prob)
    oldlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
        seBetaYG, Beta, Prec, oldIns_L, Prior$hyper_Beta_mean,
        Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
```

```
                Prior$hyper_Prec_rate)
            newlogpost <- logpost_gammaTau(BetaXG, BetaYG, seBetaXG,
                seBetaYG, Beta, Prec, Ins_L, Prior$hyper_Beta_mean,
                Prior$hyper_Beta_sd, Prior$hyper_Prec_shape,
                Prior$hyper_Prec_rate)
            oldlp <- oldlogpost + log(UBPrec - LBPrec)
            newlp <- newlogpost + log(UBPrec - LBPrec)
            if (log(runif(1, 0, 1)) > (newlp - oldlp)) {
                Ins_L <- oldIns_L  # fails so return to oldIL
                accept[i, 3] <- 0
            }
            S[i, ] <- c(Beta, 1/Prec, Ins_L)
        }
        returnlist <- list(S = S, accept_rate = apply(accept,
            2, mean))
        return(returnlist)
    }
}

### logpost_DL() ###

# Calculate the log posterior with DL esimate and hyper
# parameter options This function is used by BMA_MRanalysis()

logpost_DL <- function(BetaXG, BetaYG, seBetaXG, seBetaYG, Beta,
    indicator, hyper_Beta_mean, hyper_Beta_sd) {

    BetaXG_sub <- BetaXG[which(indicator == 1)]
    BetaYG_sub <- BetaYG[which(indicator == 1)]
    seBetaXG_sub <- seBetaXG[which(indicator == 1)]
    seBetaYG_sub <- seBetaYG[which(indicator == 1)]

    BIV = BetaYG_sub/BetaXG_sub
    se.BIV = sqrt((seBetaYG_sub^2)/BetaXG_sub^2)
    tausq <- DL(BIV, se.BIV)$tausq.hat

    logpost = -0.5 * sum(indicator) * log(2 * pi) + (-0.5 * sum(log(seBetaYG_sub^2 +
        tausq) + ((BetaYG_sub - (Beta * BetaXG_sub))^2/(Beta^2 *
        seBetaXG_sub^2 + seBetaYG_sub^2 + tausq)))) + dnorm(Beta,
        hyper_Beta_mean, hyper_Beta_sd, log = T)

    return(list(tausq = tausq, logpost = logpost))
}

### logpost_gammaTau() ###

# Calculate the log posterior with tau as gamma distribution,
# and includes hyper parameter options This function is used
# by BMA_MRanalysis()

logpost_gammaTau <- function(BetaXG, BetaYG, seBetaXG, seBetaYG,
    Beta, Prec, indicator, hyper_Beta_mean, hyper_Beta_sd, hyper_Prec_shape,
    hyper_Prec_rate) {
```

```
    BetaXG_sub <- BetaXG[which(indicator == 1)]
    BetaYG_sub <- BetaYG[which(indicator == 1)]
    seBetaXG_sub <- seBetaXG[which(indicator == 1)]
    seBetaYG_sub <- seBetaYG[which(indicator == 1)]

    Tausq <- 1/Prec

    logpost = -0.5 * sum(indicator) * log(2 * pi) + (-0.5 * sum(log(seBetaYG_sub^2 +
        Tausq) + ((BetaYG_sub - (Beta * BetaXG_sub))^2/(Beta^2 *
        seBetaXG_sub^2 + seBetaYG_sub^2 + Tausq)))) + dnorm(Beta,
        hyper_Beta_mean, hyper_Beta_sd, log = T) + dgamma(Prec,
        shape = hyper_Prec_shape, rate = hyper_Prec_rate, log = T)

    return(logpost = logpost)
}

### randomS.LI() ###

# Generate random model space that doesn't produce empty and
# 1 variable model space This function is used by
# BMA_MRanalysis()

randomS.LI <- function(Ind_L, L, ins_prior) {
    repeat {
        # do something
        a <- sample(1:L, 1, prob = ins_prior)
        Ind_L[a] <- (Ind_L[a] - 1)^2
        # exit if the condition is met
        if (sum(Ind_L) >= 5)
            break
    }
    return(Ind_L)
}
```

References

Albert, Jim. 2009. *Bayesian Computation with R*. Springer Science & Business Media.

Bowden, Jack, George Davey Smith, and Stephen Burgess. 2015. "Mendelian Randomization with Invalid Instruments: Effect Estimation and Bias Detection Through Egger Regression." *Int J Epidemiol* 44 (2). IEA: 512–25.

Burgess, Stephen, and George Davey Smith. 2017. "Mendelian Randomization Implicates High-Density Lipoprotein Cholesterol–associated Mechanisms in Etiology of Age-Related Macular Degeneration." *Ophthalmology* 124 (8). Elsevier: 1165–74.

Colijn, JM, AI den Hollander, Ayse Demirkan, Audrey Cougnard-Grégoire, Timo Verzijden, Eveline Kersten, MA Meester, et al. 2018. "Increased High Density Lipoprotein-Levels Associated with Age-Related Macular Degeneration. Evidence from the Eye-Risk and E3 Consortia." *Ophthalmology*.

Fritsche, Lars G, Wilmar Igl, Jessica N Cooke Bailey, Felix Grassmann, Sebanti Sengupta, Jennifer L Bragg-Gresham, Kathryn P Burdon, et al. 2016. "A Large Genome-Wide Association Study of Age-Related Macular Degeneration Highlights Contributions of Rare and Common Variants." *Nat Genet* 48 (2). Nature

Publishing Group: 134.

Hartwig, Fernando Pires, George Davey Smith, and Jack Bowden. 2017. "Robust Inference in Summary Data Mendelian Randomization via the Zero Modal Pleiotropy Assumption." *Int J Epidemiol* 46 (6). Oxford University Press: 1985–98.

Kang, Hyunseung, Anru Zhang, T Tony Cai, and Dylan S Small. 2016. "Instrumental Variables Estimation with Some Invalid Instruments and Its Application to Mendelian Randomization." *J Am Stat Assoc* 111 (513). Taylor & Francis: 132–44.

Kettunen, Johannes, Ayşe Demirkan, Peter Würtz, Harmen HM Draisma, Toomas Haller, Rajesh Rawal, Anika Vaarhorst, et al. 2016. "Genome-Wide Study for Circulating Metabolites Identifies 62 Loci and Reveals Novel Systemic Effects of Lpa." *Nat. Commun* 7. Nature Publishing Group: 11122.

Kolesár, Michal, Raj Chetty, John Friedman, Edward Glaeser, and Guido W Imbens. 2015. "Identification and Inference with Many Invalid Instruments." *Journal of Business & Economic Statistics* 33 (4). Taylor & Francis: 474–84.

Leeuwen, Elisabeth M van, Eszter Emri, Benedicte MJ Merle, Johanna M Colijn, Eveline Kersten, Audrey Cougnard-Gregoire, Sascha Dammeier, et al. 2018. "A New Perspective on Lipid Research in Age-Related Macular Degeneration." *Progress in Retinal and Eye Research* 67. Elsevier: 56–86.

Zhao, Qingyuan, Jingshu Wang, Jack Bowden, and Dylan S Small. 2018. "Statistical Inference in Two-Sample Summary-Data Mendelian Randomization Using Robust Adjusted Profile Score." *arXiv Preprint arXiv:1801.09652*.

Zuber, Verena, Johanna Maria Colijn, Caroline Klaver, and Stephen Burgess. 2019. "Selecting Causal Risk Factors from High-Throughput Experiments Using Multivariable Mendelian Randomization." *bioRxiv*. Cold Spring Harbor Laboratory. doi:10.1101/396333.