Integrative multi-context Mendelian randomization method for identifying risk genes across human tissues

Yihao Lu, Ke Xu, Fan Yang and Lin S. Chen

Introduction

This vignette provides an introduction to the `mintMR` package. R package `mintMR` implements the mintMR method for integrative multi-context Mendelian randomization.

Install the development version of `mintMR` by use of the 'devtools' package. Note that `mintMR` depends on the `CCA`, `Rcpp`, and `RcppArmadillo` package.

To install this package, run the following command in R

```
library(devtools)
install_github("ylustat/mintMR")
```

Load the package using the following command:

```r
library("mintMR")
```

This vignette depends on R packages `ggplot2` and `tidyverse`, you can load these two packages using the following command:

```r
suppressMessages(library("tidyverse"));
library("ggplot2");
```

Fit mintMR for correlated SNPs using simulated data

In this section, we fit mintMR using simulated data (provided in the package).

```r
data(example_data)
L <- 50
K1 <- K2 <- 5
```

The simulated data has 2 exposures, each with five contexts. It has 50 gene-CpG pairs. True effect status (0 indicates no effect, 1 indicates true non-zero effect). The mintMR requires the information for what contexts included for each exposure (the 'group' variable).

```r
group <- list(exposure1 = 1:K1, exposure2 = 1:K2+K1)
```

In addition, mintMR takes IV-to-exposure effect (a list of $I_g \times \sum K_l$ matrices with length L), IV-to-outcome effect and corresponding standard errors. The LD matrices (`corr_mat`) and sample overlap (`Lambda`) are optional.

Default prior parameters used in the algorithm is

```r
opts <- get_opts(L)
names(opts)
```

```
##  [1] "a_gamma" "b_gamma" "a_alpha" "b_alpha" "a_beta"  "b_beta"  "a"
##  [8] "b"       "maxIter" "thin"    "burnin"
```

It can be changed by specifying the parameters you want to use. For example:

```
opts <- get_opts(L, maxIter = 1000)
```

MintMR can be run without specifying LD or sample overlap using following command.

```
set.seed(2024)
res <- mintMR(gammah = dat$gamma_hat,
              Gammah = dat$Gamma_hat,
              se1 = dat$se_g,
              se2 = dat$se_G,
              corr_mat = dat$LD,
              group = group)
```

When the LD information is available and formatted as a list of matrices, it can be accounted for:

```
res_LD <- mintMR(gammah = dat$gamma_hat,
                 Gammah = dat$Gamma_hat,
                 se1 = dat$se_g,
                 se2 = dat$se_G,
                 corr_mat = dat$LD,
                 group = group)
```

When sample overlap information $\widehat{C}$ is available as a $(1+\sum_l K_l) \times (1+\sum_l K_l)$ matrix, it can be used in the function as

```
# Example correlation matrix for sample overlap. In real data, it can be pre-estimated.
C <- 0.9 * diag(1,(1+K1+K2)) +
  0.1 * matrix(1,nrow=1+K1+K2,ncol=1+K1+K2)

Lambda <- solve(C)

res_LD_sample_overlap <- mintMR(gammah = dat$gamma_hat,
                                Gammah = dat$Gamma_hat,
                                se1 = dat$se_g,
                                se2 = dat$se_G,
                                corr_mat = dat$LD,
                                Lambda = Lambda,
                                group = group)
```

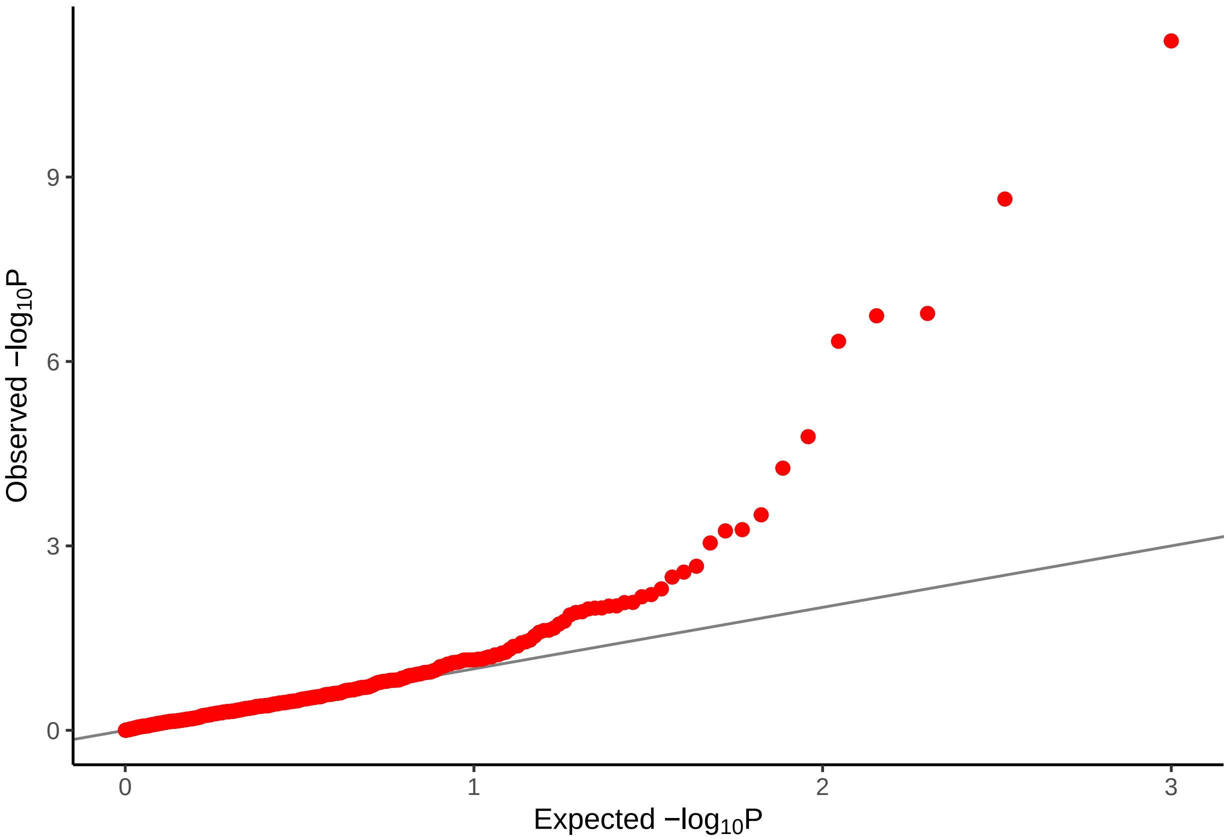
The output includes the effect estimates and p-values.

```
names(res)
```

```
## [1] "Estimate" "Pvalue"
```

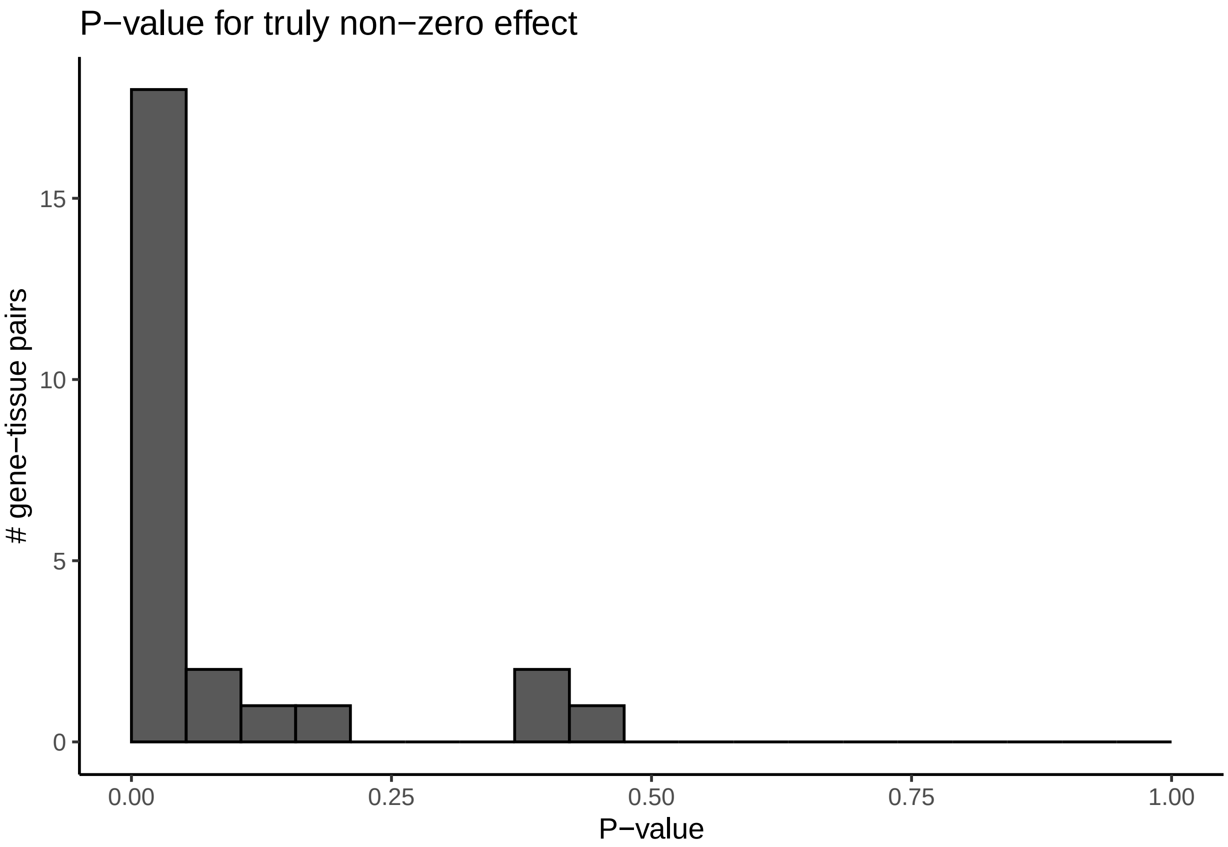
The p-value distribution for all gene-tissue pairs analyzed can be visualized as below.

```
p <- data.frame(mintMR = as.vector(res$Pvalue))
p %>%
  arrange(mintMR) %>%
  mutate(unif = ppoints(n())) %>%
  mutate(value = -log10(mintMR), unif = -log10(unif)) %>%
  ggplot(aes(y = value, x = unif))+
  geom_abline(intercept = 0, slope = 1, alpha = 0.5)+
  geom_point(size = 2, col = "red", alpha=1)+
  theme_classic()+
  labs(y = expression(paste("Observed -log"[10], plain(P))),
       x = expression(paste("Expected -log"[10], plain(P))))
```



The p-value distribution for truly non-zero effects:

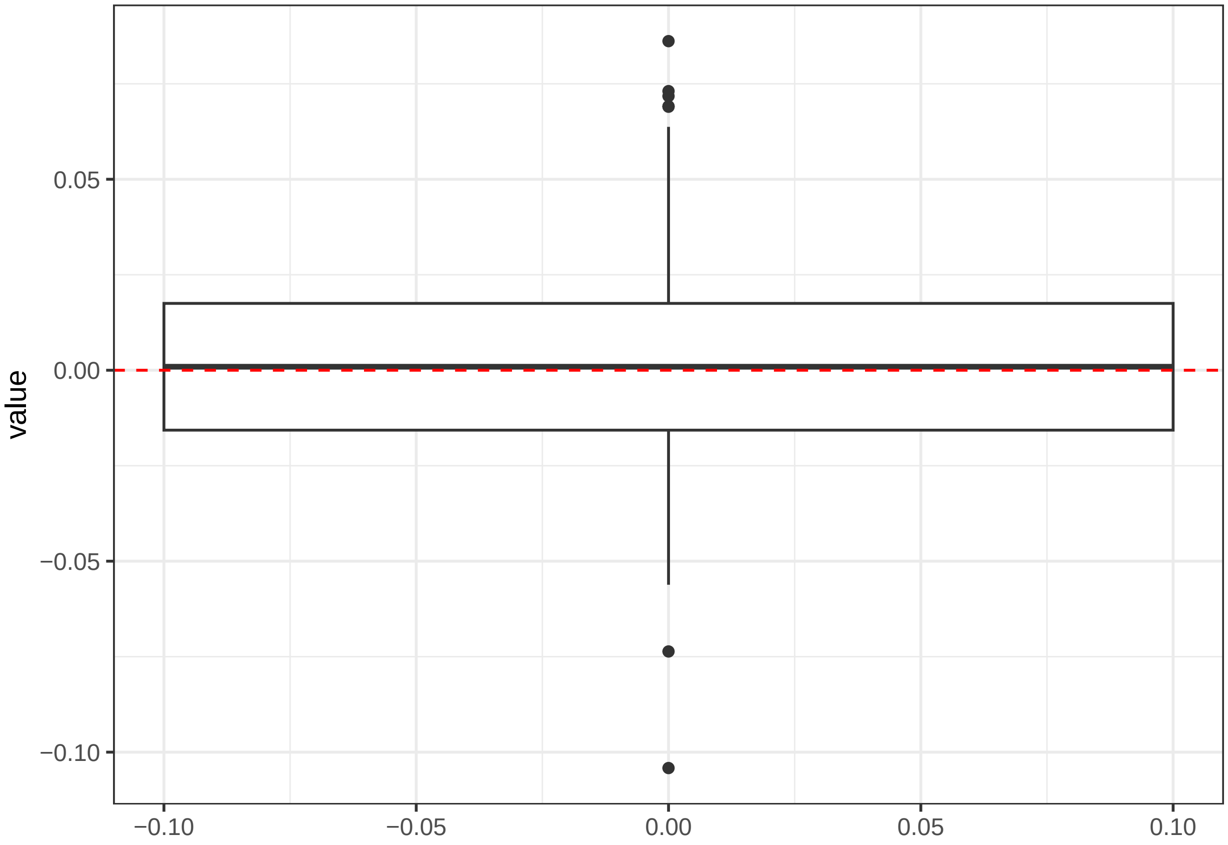
```r
p <- data.frame(mintMR = res$Pvalue[dat$latent_status == 1])
p %>%
  ggplot(aes(x = mintMR))+
  geom_histogram(col="black", boundary = 0, bins=20)+
  theme_classic()+
  scale_x_continuous(limits = c(NA,1))+
  labs(x = "P-value", y = "# gene-tissue pairs",
       title = "P-value for truly non-zero effect")
```



The effect estimate for null effects:

```r
estimate <- res$Estimate
estimate[dat$latent_status == 1] <- NA

estimate %>%
  reshape2::melt() %>%
  drop_na() %>%
  ggplot(aes(y = value))+
  theme_bw()+
  geom_boxplot(width=0.2)+
  geom_hline(yintercept = 0,col="red",linetype="dashed")
```


value
0.05
0.00
−0.05
−0.10
−0.10
−0.05
0.00
0.05
0.10