**Supplementary**

**R Code**

#--------------------------------------------------------------------------------------------------------------------------------------------------------#

library(nlme)

library(MASS)

library(TwoSampleMR)

#--------------------------------------------------------------------------------------------------------------------------------------------------------#

set.seed(17)

# Uncomment the appropriate scenario here

Scenario <- "Scenario1"

#Scenario <- "Scenario2"

# Causal exposure-outcome effect

Beta\_XY <- 0

#Beta\_XY <- 0.4

# ID of this run to save file as

ID <- Scenario

# Number of simulation replicates

Nrep <- 1000

# Number of families

N <- 50000

# Number of SNPs

Nsnps <- 20

# Covariance between residuals

Rho <- -0.3

#--------------------------------------------------------------------------------------------------------------------------------------------------------#

# Create empty vectors and matrices to store:

# Simulated error for correlation between s\_beta\_snp\_m\_exp and s\_alpha\_snp\_o\_out increasing alleles

error <- vector(mode = "numeric", length = Nsnps)

# Simulated allele frequencies

p <- vector(mode = "numeric", length = Nsnps)

q <- vector(mode = "numeric", length = Nsnps)

# Simulated association between maternal SNP and the exposure

s\_beta\_snp\_m\_exp <- vector(mode = "numeric", length = Nsnps)

# Simulated association between offspring SNP and the exposure

s\_beta\_snp\_o\_exp <- vector(mode = "numeric", length = Nsnps)

# Simulated pleiotropic effect between offspring SNP and the outcome

s\_alpha\_snp\_o\_out <- vector(mode = "numeric", length = Nsnps)

# Create empty matrix containing the number of effect alleles for each SNP in mothers, fathers and offspring

m\_genotype <- matrix(nrow = N, ncol = Nsnps)

f\_genotype <- matrix(nrow = N, ncol = Nsnps)

o\_genotype <- matrix(nrow = N, ncol = Nsnps)

# Column names of each SNP in the maternal genotype matrix

m\_column\_names <- vector(mode = "character", length = Nsnps)

# Column names of each SNP in the offspring genotype matrix

o\_column\_names <- vector(mode = "character", length = Nsnps)

# Genotype column multiplied by the corresponding association between maternal SNP and the exposure

maternal\_x <- matrix(nrow = N, ncol = Nsnps)

# Genotype column multiplied by the corresponding association between offspring SNP and the exposure

offspring\_x <- matrix(nrow = N, ncol = Nsnps)

# Genotype column multiplied by the corresponding association between maternal SNP and the outcome

maternal\_y <- matrix(nrow = N, ncol = Nsnps)

# Genotype column multiplied by the corresponding association between offspring SNP and the outcome

offspring\_y <- matrix(nrow = N, ncol = Nsnps)

# Phenotypic value of X and Y

X <- vector(mode = "numeric", length = N)

Y <- vector(mode = "numeric", length = N)

# Association between offspring SNP and exposure from regressions

beta\_snp\_o\_exp <- vector(mode = "numeric", length = Nsnps)

# Association between offspring SNP and outcome from regressions

beta\_snp\_o\_out <- vector(mode = "numeric", length = Nsnps)

# Standard error of association between offspring SNP and exposure from regressions

se\_snp\_o\_exp <- vector(mode = "numeric", length = Nsnps)

# Standard error of association between offspring SNP and outcome from regressions

se\_snp\_o\_out <- vector(mode = "numeric", length = Nsnps)

# Association between maternal SNP and exposure from regressions

beta\_snp\_m\_exp <- vector(mode = "numeric", length = Nsnps)

# Association between offspring SNP and exposure multiplied by the number of increaser alleles

o\_weighted\_score <- matrix(nrow = N, ncol = Nsnps)

# Association between offspring SNP and exposure multiplied by the number of increaser alleles

m\_weighted\_score <- matrix(nrow = N, ncol = Nsnps)

# Each individual's (offspring and maternal) sum weighted score

sum\_o\_weighted\_score <- vector(mode = "numeric", length = N)

sum\_m\_weighted\_score <- vector(mode = "numeric", length = N)

# Wald ratio instrumental variable estimate for each SNP

wald\_beta <- vector(mode = "numeric", length = Nsnps)

# Standard error of the wald ratio instrumental variable estimate for each SNP

wald\_se <- vector(mode = "numeric", length = Nsnps)

# Association between maternal SNP and expopsure from regressions

beta\_snp\_m\_exp <- vector(mode = "numeric", length = Nsnps)

# Association between maternal SNP and outcome from regressions

beta\_snp\_m\_out <- vector(mode = "numeric", length = Nsnps)

# Store results of the analyses

obs\_results <- matrix(nrow = Nrep, ncol = 3)

WAS\_results <- matrix(nrow = Nrep, ncol = 3)

IVW\_results <- matrix(nrow = Nrep, ncol = 3)

WM\_results <- matrix(nrow = Nrep, ncol = 3)

Egger\_results <- matrix(nrow = Nrep, ncol = 3)

C\_UW\_results <- matrix(nrow = Nrep, ncol = 12)

C\_W\_results <- matrix(nrow = Nrep, ncol = 12)

results <- matrix(nrow = 6, Nrep, ncol = 12)

#--------------------------------------------------------------------------------------------------------------------------------------------------------#

for (i in 1:Nrep){

# Simulate a uniform distribution of allele frequencies

p <- runif(n = Nsnps, min = 0.1, max = 0.9)

q <- 1 - p

#---------------------------------------------------------------------------------------------------------------------------------------------#

if(Scenario == "Scenario1") {

s\_beta\_snp\_o\_exp <- c(rep(0, times = Nsnps))

s\_beta\_snp\_m\_exp <- runif(n = Nsnps, min = -0.05, max = -0.01)

error <- runif(n = Nsnps, min = -0.03, max = 0.03)

s\_alpha\_snp\_o\_out <- -1.1 \* s\_beta\_snp\_m\_exp + error

} else if (Scenario == "Scenario2") {

s\_beta\_snp\_o\_exp <- runif(n = Nsnps, min = -0.05, max = -0.01)

s\_beta\_snp\_m\_exp <- runif(n = Nsnps, min = 0.01, max = 0.05)

error <- runif(n = Nsnps, min = -0.03, max = 0.03)

s\_alpha\_snp\_o\_out <- 1.1 \* s\_beta\_snp\_m\_exp + error

}

#---------------------------------------------------------------------------------------------------------------------------------------------#

# Rearrange variance of X to give var\_E1

# var(X) = varE1 + sum(s\_beta\_snp\_m\_exp^2\*2pq) + sum(s\_beta\_snp\_o\_exp^2\*2pq) +

sum(2 \* s\_beta\_snp\_m\_exp \* 2pq\* 0.5 \* s\_beta\_snp\_o\_exp)

var\_E1 <- 1 - sum(s\_beta\_snp\_m\_exp^2 \* 2 \* p \* q) - sum(2 \* p \* q \* s\_beta\_snp\_o\_exp^2)

- sum(2 \* s\_beta\_snp\_m\_exp \* 2 \* p \* q \* 0.5 \* s\_beta\_snp\_o\_exp)

# Rearranged variance of Y to give var\_E2.

# var\_Y = var\_E2 + Beta\_XY^2\*var\_X + 2\*Beta\_XY\*Rho + sum(alpha\_SNP\_O\_Y^2\*2pq) +

sum(2\*alpha\_SNP\_O\_Y\*2pq\*beta\_SNP\_O\_X\*Beta\_XY) +

sum(2\*alpha\_SNP\_O\_Y\*0.5\*2pq\*beta\_SNP\_M\_X\*Beta\_XY)

# var\_x = var\_y = 1 so these terms were dropped

var\_E2 <- 1 - Beta\_XY^2 - 2 \* Beta\_XY \* Rho - sum(s\_alpha\_snp\_o\_out^2 \* 2 \* p \* q) -

sum(2 \* s\_alpha\_snp\_o\_out \* 2 \* p \* q \* s\_beta\_snp\_o\_exp \* Beta\_XY) -

sum(2 \* s\_alpha\_snp\_o\_out \* 0.5 \* 2 \* p \* q \* s\_beta\_snp\_m\_exp \* Beta\_XY)

#---------------------------------------------------------------------------------------------------------------------------------------------#

for (k in 1:Nsnps) {

# Create a vector of O\_SNP1, O\_SNP2, .., O\_SNPn

o\_name <- paste0("O\_SNP",(k))

m\_name <- paste0("M\_SNP",(k))

o\_column\_names[k] <- o\_name

m\_column\_names[k] <- m\_name

}

colnames(o\_genotype) <- c(o\_column\_names)

colnames(m\_genotype) <- c(m\_column\_names)

#---------------------------------------------------------------------------------------------------------------------------------------------#

# Simulate the number of effect alleles for each SNP (0, 1 or 2 given the allele frequency probabilities)

for(j in 1:Nsnps) {

pp <- p[j]^2

pq2 <- 2 \* p[j] \* q[j]

qq <- q[j]^2

m\_genotype[ ,j] <- sample(x = c(2, 1, 0), size = N, replace = TRUE, prob = c(pp, pq2, qq))

f\_genotype[ ,j] <- sample(x = c(2, 1, 0), size = N, replace = TRUE, prob = c(pp, pq2, qq))

}

#---------------------------------------------------------------------------------------------------------------------------------------------#

# Offspring genotype given mother/father genotypes

o\_genotype[m\_genotype == 0 & f\_genotype == 0] <- 0

r <- runif(length(o\_genotype[m\_genotype == 0 & f\_genotype == 1]), min = 0, max = 1)

o\_genotype[m\_genotype == 0 & f\_genotype == 1] <- ifelse (r <= 0.5, 0, 1)

o\_genotype[m\_genotype == 0 & f\_genotype == 2] <- 1

r <- runif(length(o\_genotype[m\_genotype == 1 & f\_genotype == 0]), min = 0, max = 1)

o\_genotype[m\_genotype == 1 & f\_genotype == 0] <- ifelse(r <= 0.5, 0, 1)

r <- runif(length(o\_genotype[m\_genotype == 1 & f\_genotype == 1]), min = 0, max = 1)

o\_genotype[m\_genotype == 1 & f\_genotype == 1] <- ifelse(r <= 0.25, 0, ifelse(r > 0.25 & r <= 0.75, 1, 2))

r <- runif(length(o\_genotype[m\_genotype == 1 & f\_genotype == 2]), min = 0, max = 1)

o\_genotype[m\_genotype == 1 & f\_genotype == 2] <- ifelse(r <= 0.5, 1, 2)

o\_genotype[m\_genotype == 2 & f\_genotype == 0] <- 1

r <- runif(length(o\_genotype[m\_genotype == 2 & f\_genotype == 1]), min = 0, max = 1)

o\_genotype[m\_genotype == 2 & f\_genotype == 1] <- ifelse(r <= 0.5, 1, 2)

o\_genotype[m\_genotype == 2 & f\_genotype == 2] <- 2

#---------------------------------------------------------------------------------------------------------------------------------------------#

# Simulate X (birthweight) and Y for each offspring

if(Scenario == "Scenario1") {

# Set the mean for X and Y (systolic blood pressure) at 3.5Kg and 120 mm Hg

mu <- c(3.5, 120)

# Sigma is the E1 E2 covariance matrix

sigma <- matrix(c(var\_E1, Rho, Rho, var\_E2), 2, 2)

# Draw values of E1 and E2 from the multivariate normal distribution

E <- mvrnorm(n = N, mu, sigma, tol = 0.1, empirical = FALSE, EISPACK = FALSE)

} else if (Scenario == "Scenario2") {

# Set the mean for X and Y (fasting glucose levels) at 3.5Kg and 5 mmol/L

mu <- c(3.5, 5)

# Sigma is the E1 E2 covariance matrix

sigma <- matrix(c(var\_E1, Rho, Rho, var\_E2), 2, 2)

# Draw values of E1 and E2 from the multivariate normal distribution

E <- mvrnorm(n = N, mu, sigma, tol = 0.1, empirical = FALSE, EISPACK = FALSE)

}

#---------------------------------------------------------------------------------------------------------------------------------------------#

for(j in 1:Nsnps) {

# Multiply each genotype by the corresponding beta\_SNP\_EXP element

maternal\_x[ ,j] <- m\_genotype[ ,j] \* s\_beta\_snp\_m\_exp[j]

offspring\_x[ ,j] <- o\_genotype[ ,j] \* s\_beta\_snp\_o\_exp[j]

offspring\_y[ ,j] <- o\_genotype[ ,j] \* s\_alpha\_snp\_o\_out[j]

}

# Calculate the phenotypic value of X and Y for each individual

X <- E[ ,1] + rowSums(maternal\_x) + rowSums(offspring\_x)

Y <- Beta\_XY \* X + E[ ,2] + rowSums(offspring\_y)

# Observational study (extract beta, SE and p-value)

MRdata <- as.data.frame(cbind(X, Y))

obs\_beta <- summary(lm(Y ~ X, data = MRdata))$coef[2,1]

obs\_se <- summary(lm(Y ~ X, data = MRdata))$coef[2,2]

obs\_p <- summary(lm(Y ~ X, data = MRdata))$coef[2,4]

obs\_results[i, ] <- c(obs\_beta, obs\_se, obs\_p)

########################################################################################

# Conditional analysis using unweighted scores #

########################################################################################

# Construct unweighted scores

m\_score <- rowSums(m\_genotype)

o\_score <- rowSums(o\_genotype)

# Set X, Y, maternal score, offspring score as columns of MRdata

MRdata <- as.data.frame(cbind(X, Y, m\_score, o\_score))

# Extract unbiased estimates of the maternal and offspring effects on the exposure

and outcome (extract beta, SE and p-value)

c\_uw\_beta\_m\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[2,1]

c\_uw\_se\_m\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[2,2]

c\_uw\_p\_m\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[2,4]

c\_uw\_beta\_o\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[3,1]

c\_uw\_se\_o\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[3,2]

c\_uw\_p\_o\_y <- summary(lm(Y ~ m\_score + o\_score, data = MRdata))$coef[3,4]

c\_uw\_beta\_m\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[2,1]

c\_uw\_se\_m\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[2,2]

c\_uw\_p\_m\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[2,4]

c\_uw\_beta\_o\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[3,1]

c\_uw\_se\_o\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[3,2]

c\_uw\_p\_o\_x <- summary(lm(X ~ m\_score + o\_score, data = MRdata))$coef[3,4]

C\_UW\_results[i, ] <- c(c\_uw\_beta\_m\_y, c\_uw\_se\_m\_y, c\_uw\_p\_m\_y,

c\_uw\_beta\_o\_y, c\_uw\_se\_o\_y, c\_uw\_p\_o\_y,

c\_uw\_beta\_m\_x, c\_uw\_se\_m\_x, c\_uw\_p\_m\_x,

c\_uw\_beta\_o\_x, c\_uw\_se\_o\_x, c\_uw\_p\_o\_x)

########################################################################################

# Conditional analysis using weighted maternal scores #

# conditioning on offspring SNPs individually #

########################################################################################

# Weight the maternal scores

for (m in 1:Nsnps) { m\_weighted\_score[ ,m] <- (s\_beta\_snp\_m\_exp[m] \* m\_genotype[ ,m]) }

sum\_m\_weighted\_score <- rowSums(m\_weighted\_score)

# Weight the offspring scores

for (m in 1:Nsnps) { o\_weighted\_score[ ,m] <- (s\_beta\_snp\_o\_exp[m] \* o\_genotype[ ,m]) }

sum\_o\_weighted\_score <- rowSums(o\_weighted\_score)

# Extract unbiased estimates of the maternal effects on the outcome (beta, SE and p)

MRdata <- as.data.frame(cbind(Y, sum\_m\_weighted\_score, o\_genotype))

c\_w\_beta\_m\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,1]

c\_w\_se\_m\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,2]

c\_w\_p\_m\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,4]

MRdata <- as.data.frame(cbind(Y, sum\_o\_weighted\_score, m\_genotype))

c\_w\_beta\_o\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,1]

c\_w\_se\_o\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,2]

c\_w\_p\_o\_y <- summary(lm(formula(MRdata), data = MRdata))$coef[2,4]

MRdata <- as.data.frame(cbind(X, sum\_m\_weighted\_score, o\_genotype))

c\_w\_beta\_m\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,1]

c\_w\_se\_m\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,2]

c\_w\_p\_m\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,4]

MRdata <- as.data.frame(cbind(X, sum\_o\_weighted\_score, m\_genotype))

c\_w\_beta\_o\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,1]

c\_w\_se\_o\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,2]

c\_w\_p\_o\_x <- summary(lm(formula(MRdata), data = MRdata))$coef[2,4]

C\_W\_results[i, ] <- c(c\_w\_beta\_m\_y, c\_w\_se\_m\_y, c\_w\_p\_m\_y,

c\_w\_beta\_o\_y, c\_w\_se\_o\_y, c\_w\_p\_o\_y,

c\_w\_beta\_m\_x, c\_w\_se\_m\_x, c\_w\_p\_m\_x,

c\_w\_beta\_o\_x, c\_w\_se\_o\_x, c\_w\_p\_o\_x)

########################################################################################

# Perform regressions to estimate betas and standard errors to #

# be used in the un-conditional MR-Analyses #

########################################################################################

# Set X, Y, SNPo etc. as columns of MRdata

MRdata <- as.data.frame(cbind(X, Y, o\_genotype))

# Regress each offspring genotype on exposure to get beta\_snp\_o\_exp for each SNP

for (m in 1:Nsnps) {

# Create the formula for each SNP-EXPOSURE regression.

eg. lm(X ~ SNP1, data = MRdata)

model1 <- lm(formula(paste0("X ~ O\_SNP",m)), MRdata)

# Create the formula for each SNP-OUTCOME regression.

eg. lm(Y ~ SNP, data = MRdata)

model2 <- lm(formula(paste0("Y ~ O\_SNP",m)), MRdata)

# Extract the beta (beta\_snp\_exp)

beta1 <- summary(model1)$coef[2,1]

# Extract the SE (se\_snp\_exp)

se1 <- summary(model1)$coef[2,2]

# Extract the beta (beta\_snp\_out)

beta2 <- summary(model2)$coef[2,1]

# Extract the SE (se\_snp\_out)

se2 <- summary(model2)$coef[2,2]

beta\_snp\_o\_exp[m] <- c(beta1)

se\_snp\_o\_exp[m] <- c(se1)

beta\_snp\_o\_out[m] <- c(beta2)

se\_snp\_o\_out[m] <- c(se2)

}

########################################################################################

# Inverse Variance Weighted Mendelian Randomization #

########################################################################################

ivw <- mr\_ivw(b\_exp = beta\_snp\_o\_exp, b\_out = beta\_snp\_o\_out,

se\_exp = se\_snp\_o\_exp, se\_out = se\_snp\_o\_out)

IVW\_effect <- ivw$b

IVW\_SE <- ivw$se

IVW\_pvalue <- ivw$pval

IVW\_results[i, ] <- c(IVW\_effect, IVW\_SE, IVW\_pvalue)

########################################################################################

# Weighted Median Mendelian Randomization #

########################################################################################

wm <- mr\_weighted\_median(b\_exp = beta\_snp\_o\_exp, b\_out = beta\_snp\_o\_out,

se\_exp = se\_snp\_o\_exp, se\_out = se\_snp\_o\_out)

WM\_effect <- wm$b

WM\_SE <- wm$se

WM\_pvalue <- wm$pval

WM\_results[i, ] <- c(WM\_effect, WM\_SE, WM\_pvalue)

########################################################################################

# MR-Egger Regression #

########################################################################################

egger <- mr\_egger\_regression(b\_exp = beta\_snp\_o\_exp, b\_out = beta\_snp\_o\_out,

se\_exp = se\_snp\_o\_exp, se\_out = se\_snp\_o\_out)

Egger\_effect <- egger$b

Egger\_SE <- egger$se

Egger\_pvalue <- egger$pval

Egger\_results[i, ] <- c(Egger\_effect, Egger\_SE, Egger\_pvalue)

########################################################################################

# Weighted Allele Score Mendelian Randomization #

######################################################################################### Weight the scores

for (m in 1:Nsnps) { o\_weighted\_score[ ,m] <- (beta\_snp\_o\_exp[m] \* o\_genotype[ ,m]) }

sum\_o\_weighted\_score <- rowSums(o\_weighted\_score)

MRdata <- as.data.frame(cbind(sum\_o\_weighted\_score, X, Y))

# Regress the scaled score on exposure and outcome

# Extract the offspring snp-exposure

beta\_o\_x <- summary(lm(X ~ sum\_o\_weighted\_score, data = MRdata))$coef[2,1]

# Extract the beta coefficient (beta\_snp\_out)

beta\_o\_y <- summary(lm(Y ~ sum\_o\_weighted\_score, data = MRdata))$coef[2,1]

# Extract the se(beta\_snp\_out)

se\_o\_y <- summary(lm(Y ~ sum\_o\_weighted\_score, data = MRdata))$coef[2,2]

# Divide the offspring snp-outcome effect by the offspring snp-exposure effect

WAS\_effect <- beta\_o\_y / beta\_o\_x

WAS\_se <- se\_o\_y / beta\_o\_x

# Calcualte the IV estimate p-value

WAS\_pvalue <- pnorm(abs(WAS\_effect/WAS\_se), lower.tail=F)

WAS\_results[i, ] <- c(WAS\_effect, WAS\_se, WAS\_pvalue)

}

###############################################################################################

# RESULTS #

###############################################################################################

# Save input

input <- cbind (ID, Nrep, N, Nsnps, Beta\_XY, Rho)

# Set up the function for statistical analyses

analyse <- function(dataset, beta, se, p){

mean\_beta <<- mean(dataset[,beta])

mean\_se <<- mean(dataset[,se])

mean\_p <<- mean(dataset[,p])

lower\_ci <<- mean\_beta - 1.96 \* mean\_se

upper\_ci <<- mean\_beta + 1.96 \* mean\_se

T1\_error <<- (sum(dataset[,p] < 0.05))/Nrep

T1\_error\_se <<- sqrt((T1\_error \* (1 - T1\_error)) / Nrep)

T1\_error\_lower <<- T1\_error - 1.96 \* T1\_error\_se

T1\_error\_upper <<- T1\_error + 1.96 \* T1\_error\_se

mcse\_se <<- sqrt((sum((dataset[,beta] - mean\_beta)^2))/((Nrep\*(Nrep - 1))))

mcse\_lower\_ci <<- mean\_beta - 1.96 \* mcse\_se

mcse\_upper\_ci <<- mean\_beta + 1.96 \* mcse\_se

# Return estimates of interest

c("Mean\_Beta" = mean\_beta,

"Mean\_SE" = mean\_se,

"Mean\_P" = mean\_p,

"Lower\_CI" = lower\_ci,

"Upper\_CI" = upper\_ci,

"T1\_Error" = T1\_error,

"T1\_Error\_SE" = T1\_error\_se,

"T1\_Error\_Lower\_CI" = T1\_error\_lower,

"T1\_Error\_Upper\_CI" = T1\_error\_upper,

"MCSE" = mcse\_se,

"MCSE\_Lower\_CI" = mcse\_lower\_ci,

"MCSE\_Upper\_CI" = mcse\_upper\_ci)

}

#---------------------------------------------------------------------------------------------------------------------------------------------#

obs\_results <- data.frame(obs\_results)

WAS\_results <- data.frame(WAS\_results)

IVW\_results <- data.frame(IVW\_results)

WM\_results <- as.data.frame(WM\_results)

Egger\_results <- as.data.frame(Egger\_results)

C\_W\_results <- data.frame(C\_W\_results)

C\_UW\_results <- data.frame(C\_UW\_results)

cnames <- c("Effect", "SE", "P")

colnames(obs\_results) <- cnames

colnames(WAS\_results) <- cnames

colnames(IVW\_results) <- cnames

colnames(WM\_results) <- cnames

colnames(Egger\_results) <- cnames

colnames(C\_UW\_results) <- c("c\_uw\_beta\_m\_y", "c\_uw\_se\_m\_y", "c\_uw\_p\_m\_y",

"c\_uw\_beta\_o\_y", "c\_uw\_se\_o\_y", "c\_uw\_p\_o\_y",

"c\_uw\_beta\_m\_x", "c\_uw\_se\_m\_x", "c\_uw\_p\_m\_x",

"c\_uw\_beta\_o\_x", "c\_uw\_se\_o\_x", "c\_uw\_p\_o\_x")

colnames(C\_W\_results) <- c("c\_w\_beta\_m\_y", "c\_w\_se\_m\_y", "c\_w\_p\_m\_y",

"c\_w\_beta\_o\_y", "c\_w\_se\_o\_y", "c\_w\_p\_o\_y",

"c\_w\_beta\_m\_x", "c\_w\_se\_m\_x", "c\_w\_p\_m\_x",

"c\_w\_beta\_o\_x", "c\_w\_se\_o\_x", "c\_w\_p\_o\_x")

obs <- analyse(dataset = obs\_results, beta = "Effect", se = "SE", p = "P")

WAS <- analyse(dataset = WAS\_results, beta = "Effect", se = "SE", p = "P")

IVW <- analyse(dataset = IVW\_results, beta = "Effect", se = "SE", p = "P")

WM <- analyse(dataset = WM\_results, beta = "Effect", se = "SE", p = "P")

Egger <- analyse(dataset = Egger\_results, beta = "Effect", se = "SE", p = "P")

C\_UW\_M\_Y <- analyse(dataset = C\_UW\_results, beta = "c\_uw\_beta\_m\_y", se = "c\_uw\_se\_m\_y", p = "c\_uw\_p\_m\_y")

results <- rbind(WAS, IVW, WM, Egger,C\_UW\_M\_Y, C\_W\_M\_Y)

# Name saved results according to the scenario and whether Beta\_XY is causal

if(Beta\_XY == 0) {

write.table(x = obs\_results, file=paste(ID,"\_obs\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = WAS\_results, file=paste(ID,"\_WAS\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = IVW\_results, file=paste(ID,"\_IVW\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = WM\_results, file=paste(ID,"\_WM\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = Egger\_results, file=paste(ID,"\_Egger\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = C\_UW\_results, file=paste(ID,"\_C\_UW\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = C\_W\_results, file=paste(ID,"\_C\_W\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = results, file=paste(ID,"\_results.txt", sep = ""),

quote = F, row.names = F, col.names = T)

write.table(x = input, file=paste(ID,"\_input.txt", sep = ""),

quote = F, row.names = F, col.names = T)

} else {

write.table(x = obs\_results, file=paste(ID,"\_causal\_obs\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = WAS\_results, file=paste(ID,"\_causal\_WAS\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = IVW\_results, file=paste(ID,"\_causal\_IVW\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = WM\_results, file=paste(ID,"\_causal\_WM\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = Egger\_results, file=paste(ID,"\_causal\_Egger\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = C\_UW\_results, file=paste(ID,"\_causal\_C\_UW\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = C\_W\_results, file=paste(ID,"\_causal\_C\_W\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = results, file=paste(ID,"\_causal\_results.txt",

sep = ""), quote = F, row.names = F, col.names = T)

write.table(x = input, file=paste(ID,"\_causal\_input.txt", sep = ""),

quote = F, row.names = F, col.names = T)

}