Confounder analysis in Steiger filtering

Gibran Hemani

Background

For a system in which the MR Steiger test infers that X is causal for Y, and the estimated effect is β_{xy}, where

$$X = \alpha_x + \beta_{gx}G + \beta_{ux}U + e_x$$

where SNP with allele frequency p has variance $\sigma^2_G = 2 * p * (1 - p)$, $U \sim N(0, \sigma^2_u)$ is an unmeasured confounder, and $e_x \sim N(0, \sigma^2_{e_x})$ is an error term. The variance of X will be

$$\sigma^2_x = \beta^2_{gx}\sigma^2_g + \beta^2_{ux}\sigma^2_u + \sigma^2_{e_x}$$

Write

$$Y = \alpha_y + \beta_{xy}X + \beta_{uy}U + e_y$$

where $e_y \sim N(0, \sigma^2_{e_y})$ is an error term. Going forwards intercept terms can be ignored. The variance of Y will be

$$\sigma^2_y = \beta^2_{xy}\beta^2_{gx}\sigma^2_g + \sigma^2_u(\beta_{xy}\beta_{ux} + \beta_{uy})^2 + \beta^2_{xy}\sigma^2_{e_x} + \sigma^2_{e_y}$$

The variance explained in X by G will be

$$R^2_{gx} = \frac{\beta^2_{gx}\sigma^2_g}{\beta^2_{gx}\sigma^2_g + \beta^2_{ux} * \sigma^2_u + \sigma^2_{e_x}}$$

The variance explained in Y by G will be

$$R^2_{gy} = \frac{\beta^2_{gx}\beta^2_{xy}\sigma^2_g}{\sigma^2_u(\beta^2_{xy}\beta^2_{gx}\sigma^2_g + \sigma^2_u(\beta_{xy}\beta_{ux} + \beta_{uy})^2 + \beta^2_{xy}\sigma^2_{e_x} + \sigma^2_{e_y}}$$

The variance explained in X by U will be

$$R^2_{ux} = \frac{\beta^2_{gx}\sigma^2_g}{\beta^2_{gx}\sigma^2_g + \beta^2_{ux}\sigma^2_u + \sigma^2_{e_x}}$$

The variance explained in Y by U will be

$$R^2_{uy} = \frac{\sigma^2_u(\beta_{uy} + \beta_{ux}\beta_{xy})^2}{\beta^2_{xy}\beta^2_{gx}\sigma^2_g + \sigma^2_u(\beta_{xy}\beta_{ux} + \beta_{uy})^2 + \beta^2_{xy}\sigma^2_{e_x} + \sigma^2_{e_y}}$$

Under this system, the observed R^2_{gy} will be smaller than the observed R^2_{gx} unless all variance in Y is explained by X, in which case $R^2_{gy} = R^2_{gx}$. In practice we tend to know the following values: β_{gx}, σ^2_g, σ^2_x, σ^2_y. The analysis is used to estimate β_{xy}. We can often obtain estimates of β_{OLS}. We do not know σ^2_u, β_{ux} or β_{uy}, but given estimates of β_{OLS} and β_{xy} we can obtain possible values for these confounder parameters. The observational association in this system will be

$$\beta_{OLS} = \frac{\beta^2_{gx}\beta_{xy}\sigma^2_g + \beta^2_{ux}\beta_{xy}\sigma^2_u + \beta_{ux}\beta_{uy}\sigma^2_u + \beta_{xy}\sigma^2_{e_x}}{\sigma^2_g\beta^2_{gx} + \sigma^2_u\beta^2_{ux} + \sigma^2_{e_x}}$$

Hence the association between X and Y due to confounding will be

$$\begin{aligned}\beta_C &= \beta_{OLS} - \beta_{xy} \\ &= \frac{\beta_{ux}\beta_{uy}\sigma^2_u}{\sigma^2_g\beta^2_{gx} + \sigma^2_u\beta^2_{ux} + \sigma^2_{e_x}}\end{aligned}$$

The key question is this: **If β_{gx}, σ^2_g, σ^2_x, σ^2_y, $\hat{\beta}_{OLS}$ and $\hat{\beta}_{xy}$ are fixed, are there values of R_{ux} and R_{uy} that can satisfy either X being causal for Y or Y being causal for X?** We approach this question by analytically exploring this possible confounding parameter space. The possible range of U-X confounding is

$$\beta_{ux} \in \left\{ -\sqrt{\frac{\sigma^2_x - \beta^2_{gx}\sigma^2_g}{\sigma^2_u}}, \sqrt{\frac{\sigma^2_x - \beta^2_{gx}\sigma^2_g}{\sigma^2_u}} \right\}$$

which means that for any particular value of β_{ux} within this range the values of

$$\sigma^2_{e_x} = \sigma^2_x - \beta^2_{gx}\sigma^2_g - \beta^2_{ux}\sigma^2_u$$

and

$$\beta_{uy} = \beta_C \frac{\beta^2_{gx}\sigma^2_g + \beta^2_{ux}\sigma^2_u + \sigma^2_{e_x}}{\beta_{ux}\sigma^2_u}$$

and

$$\sigma^2_{e_y} = \sigma^2_y - \beta^2_{xy}\beta^2_{gx}\sigma^2_g - \sigma^2_u(\beta_{xy}\beta_{ux} + \beta_{uy})^2 - \beta^2_{xy}\sigma^2_{e_x}$$

can be inferred directly. Overall, through this set of equations, we can obtain confounding values that could give rise to the observed fixed parameters under either the inferred causal direction or the reverse causal direction. In the case of the reverse causal direction the value of $\beta_{xy,rev} = 1/\beta_{xy}$ and $\beta_{OLS,rev} = \beta_{OLS}\sigma^2_x/\sigma^2_y$, $\beta_{gx,rev} = \beta_{gx}\beta_{xy}$. The sensitivity analysis proceeds by finding the total confounding parameter space across models for the inferred causal direction and the reverse causal direction, and then calculating the fraction of that parameter space that agrees with the inferred causal direction. A proportion close to 1 will suggest that there is relatively little chance of the inferred direction being incorrect due to unmeasured confounding. If the OLS estimate is unknown then a range of plausible values can be evaluated.

One further component to this approach is the option to weight the possible parameter space. We might consider it less plausible that large fractions of the variance in X and Y are explained by confounding variables, and so the contribution of scenarios that have confounding values that explain more of the variance can be downweighted. The weighting is obtained by

$$w = \phi_{0,s}(R^2_{ux})\phi_{0,s}(R^2_{uy})$$

where $\phi_{0,s}$ is the normal density function with mean 0 and standard deviation s, the scaling parameter. Smaller s will lead to more downweighting of larger confounding variances.

Analysis

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(latex2exp)
library(ieugwasr)
```

```
## API: public: http://gwas-api.mrcieu.ac.uk/
```

```
library(TwoSampleMR)
```

```
## TwoSampleMR version 0.5.6
## [>] New: Option to use non-European LD reference panels for clumping etc
## [>] Some studies temporarily quarantined to verify effect allele
## [>] See news(package='TwoSampleMR') and https://gwas.mrcieu.ac.uk for further details

##
## Attaching package: 'TwoSampleMR'

## The following object is masked from 'package:ieugwasr':
##
##     ld_matrix
```

This function obtains the rsq values given fixed parameters

```
get_calcs <- function(bxy, bgx, bux, buy, vg, vu, vex, vey) {
    bxyo <- ((bgx^2*bxy*vg + bux^2*bxy*vu + bux*buy*vu + bxy*vex) / (vg*bgx^2 + vu*bux^2 + vex))
    vx <- bgx^2 * vg + bux^2 * vu + vex
    vy <- bxy^2*bgx^2*vg + (bxy*bux+buy)^2*vu + bxy^2*(vex) + vey
    conf <- bux * vu * buy / (vg * bgx^2 + vu * bux^2 + vex)
    rsqxyo <- bxyo^2 * vx / vy
    rsqxyos <- rsqxyo * sign(bxyo)
    rsqxy <- bxy^2 * vx / vy
```

```
    rsqxys <- rsqxy * sign(bxy)
    rsqgx <- bgx^2*vg / (bgx^2 * vg + bux^2 * vu + vex)
    rsqgy <- bgx^2*bxy^2*vg / (bxy^2*bgx^2*vg + (bxy*bux+buy)^2*vu + bxy^2*(vex) + vey)
    rsqux <- bux^2*vu / (bgx^2 * vg + bux^2 * vu + vex)
    rsquy <- (buy + bux * bxy)^2 * vu / (bxy^2*bgx^2*vg + (bxy*bux+buy)^2*vu + bxy^2*(vex) + vey)
    rsquxs <- rsqux * sign(bux)
    rsquys <- rsquy * sign(buy)
    return(list(
        vx=vx,
        vy=vy,
        bxyo=bxyo,
        conf=conf,
        rsqgx=rsqgx,
        rsqgy=rsqgy,
        rsqux=rsqux,
        rsquy=rsquy,
        rsqxy=rsqxy,
        rsqxyo=rsqxyo,
        rsqxyos=rsqxyos,
        rsquxs=rsquxs,
        rsquys=rsquys,
        rsqxys=rsqxys
    ))
}
```

Check by comparing to simulated individual level data

```
get_calcs_id <- function(bxy, bgx, bux, buy, vg, vu, vex, vey, n=500000){
    u <- rnorm(n, sd = sqrt(vu))
    g <- rnorm(n, sd = sqrt(vg))
    ex <- rnorm(n, sd = sqrt(vex))
    ey <- rnorm(n, sd = sqrt(vey))
    x <- u * bux + g * bgx + ex
    y <- u * buy + x * bxy + ey
    res <- tibble(
        bxyo=cov(x,y)/var(x),
        rsqux=cor(u,x)^2,
        rsquy=cor(u,y)^2,
        rsqxyo=cor(x,y)^2,
        rsqxyos=rsqxyo*sign(bxyo),
        rsquxs=rsqux*sign(bux),
        rsquys=rsquy*sign(buy),
        rsqgx = cor(g,x)^2,
        rsqgy = cor(g,y)^2
    )
    return(res)
}
bind_rows(get_calcs(0.1, 1, 1, 1, 0.5, 0.1, 0.4, 0.9-0.1^2) %>% as_tibble(),
get_calcs_id(0.1, 1, 1, 1, 0.5, 0.1, 0.4, 0.9-0.1^2) %>% as_tibble())
```

```
## # A tibble: 2 x 14
##      vx    vy  bxyo  conf rsqgx   rsqgy rsqux rsquy    rsqxy rsqxyo rsqxyos
##   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl>    <dbl>  <dbl>   <dbl>
```

```
## 1      1  1.02 0.2      0.1 0.5    0.00490 0.1    0.119   0.00980 0.0392   0.0392
## 2     NA NA     0.199   NA    0.501 0.00509 0.100 0.118 NA        0.0390   0.0390
## # ... with 3 more variables: rsquxs <dbl>, rsquys <dbl>, rsqxys <dbl>
```

Parameter ranges used by Lutz et al

```
bind_rows(
    get_calcs(1, 1, -5, 0, 1, 1, 1, 1) %>% as_tibble(),
    get_calcs(1, 1, -5, 11, 1, 1, 1, 1) %>% as_tibble()
)
```

```
## # A tibble: 2 x 14
##       vx    vy  bxyo   conf   rsqgx   rsqgy rsqux rsquy rsqxy rsqxyo rsqxyos rsquxs
##    <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>  <dbl>
## 1     27    28  1      0     0.0370 0.0357 0.926 0.893 0.964  0.964   0.964 -0.926
## 2     27    39 -1.04 -2.04 0.0370 0.0256 0.926 0.923 0.692  0.745  -0.745 -0.926
## # ... with 2 more variables: rsquys <dbl>, rsqxys <dbl>
```

Calculate the values of rux and ruy that would satisfy the fixed parameters

```
sens <- function(bxy=0.1, bxyo=0.2, bgx=0.5, vx=1, vy=1, vu=1, vg=0.5, simsize=100) {
    # vx <- bgx^2 * vg + p$bux_vec^2 * vu + vex
    bux_lim <- sqrt((vx - bgx^2 * vg)/vu)
    bux_vec <- seq(-bux_lim, bux_lim, length.out=simsize)
    # Allow causal effect to vary by +/- 200%
    vex <- vx - bgx^2 * vg - bux_vec^2 * vu
    conf <- bxyo - bxy
    buy_vec <- conf * (bgx^2*vg + bux_vec^2*vu + vex) / (bux_vec * vu)
    vey <- vy - (bxy^2*bgx^2*vg + (bxy*bux_vec+buy_vec)^2*vu + bxy^2*vex)
    # vy <- bxy^2*bgx^2*vg + (bxy*bux_vec+buy_vec)^2*vu + bxy^2*vex + vey
    bux_vec * vu * buy_vec / (vg * bgx^2 + vu * bux_vec^2 + vex)
    res <- get_calcs(bxy, bgx, bux_vec, buy_vec, vg, vu, vex, vey) %>%
      as_tibble() %>%
      mutate(bxy=bxy, bgx=bgx, bux=bux_vec, buy=buy_vec, vg=vg, vu=vu, vex=vex, vey=vey)
    return(res)
}
```

Test sensitivity analysis

```
u_sensitivity <- function(bxy, bxyo, bgx, vx, vy, vg, vu = 1, simsize=10000, scaling=1, plot=TRUE)
{
    o <- params <- bind_rows(
        sens(bxy=bxy, bxyo=bxyo, bgx=bgx, vx=vx, vy=vy, vu=vu, vg=vg, simsize=simsize) %>%
            mutate(direction="inferred"),
        sens(bxy=1/bxy, bxyo=bxyo * vx / vy, bgx=bgx * bxy, vx=vy, vy=vx, vu=vu, vg=vg, simsize=simsize)
            mutate(direction="reverse")
    ) %>%
    filter(
        vex >= 0 &
        vey >= 0 &
        rsquy >= 0 & rsquy <= 1 &
        rsqux >= 0 & rsqux <= 1 &
```

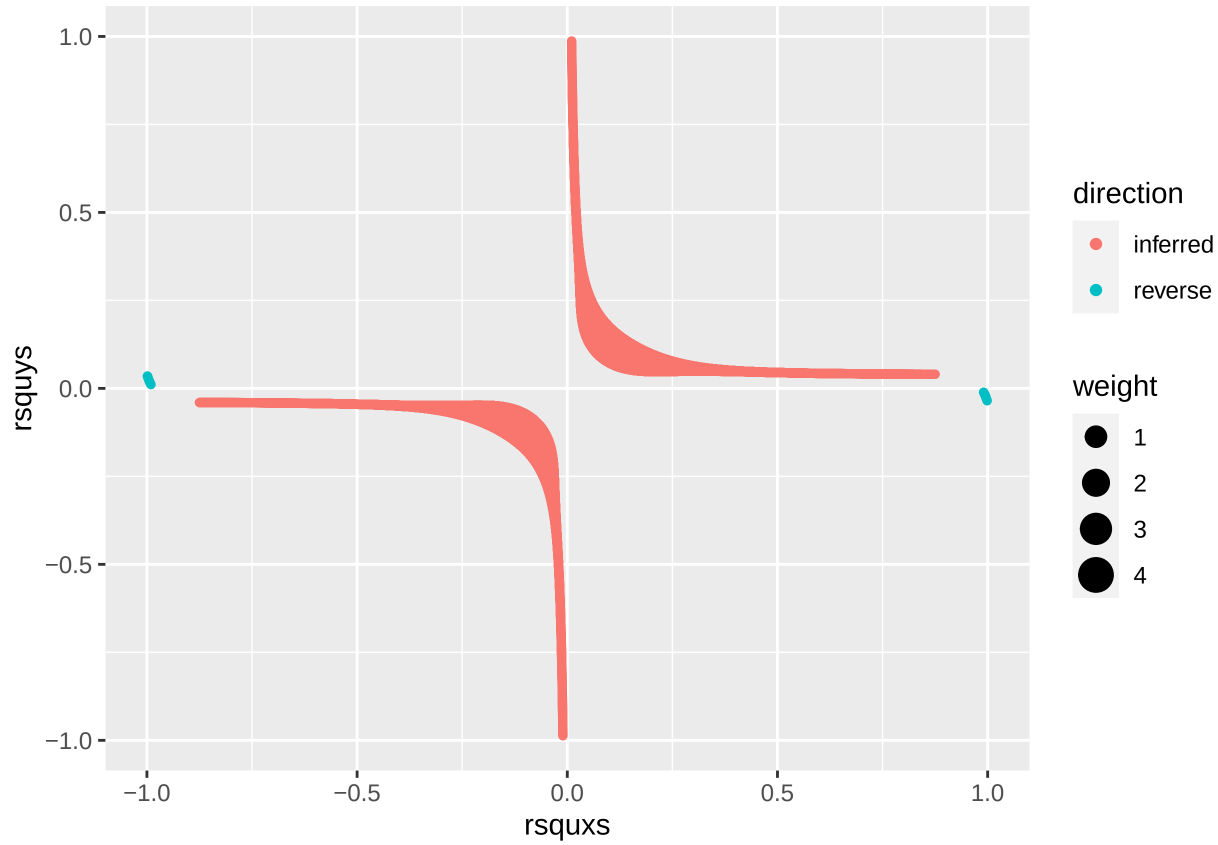
```
        rsqgx >= 0 & rsqgx <= 1 &
        rsqgy >= 0 & rsqgy <= 1
    ) %>%
        group_by(direction) %>%
        do({
            x <- .
            x1 <- x$rsqux[-1]
            x2 <- x$rsqux[-length(x$rsqux)]
            y1 <- x$rsquy[-1]
            y2 <- x$rsquy[-length(x$rsquy)]
            d <- sqrt((x1-x2)^2 + (y1-y2)^2)
            d[d > quantile(d, na.rm=T, probs=0.99)*4] <- NA
            x$d <- c(NA, d)
            x$weight <- dnorm(x$rsqux, sd=scaling) * dnorm(x$rsquy, sd=scaling)
            x
        })

    w <- o$d * o$weight
    w1 <- w[o$direction=="inferred"]
    prop <- sum(w1, na.rm=T) / sum(w, na.rm=T)
    ret <- list(result=o, prop=prop)
    if(plot) {
        ret$pl <- ggplot(o, aes(x=rsquxs, y=rsquys)) +
        geom_point(aes(colour=direction, size=weight))
    }
    return(ret)
}
```

Example 1 - bxy and bxyo are similar, and confounders that explain more of the variance are strongly downweighted

```
r <- u_sensitivity(bxy=0.1, bxyo=0.2, bgx=0.5, vx=1, vy=1, vg=0.5, scaling=0.1)
r
```

```
## $result
## # A tibble: 8,958 x 25
## # Groups:   direction [2]
##       vx    vy  bxyo  conf rsqgx   rsqgy rsqux  rsquy rsqxy rsqxyo rsqxyos
##    <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>  <dbl>   <dbl>
##  1     1     1   0.2   0.1 0.125 0.00125 0.875 0.0402  0.01   0.04    0.04
##  2     1     1   0.2   0.1 0.125 0.00125 0.875 0.0402  0.01   0.04    0.04
##  3     1     1   0.2   0.1 0.125 0.00125 0.874 0.0402  0.01   0.04    0.04
##  4     1     1   0.2   0.1 0.125 0.00125 0.874 0.0402  0.01   0.04    0.04
##  5     1     1   0.2   0.1 0.125 0.00125 0.874 0.0402  0.01   0.04    0.04
##  6     1     1   0.2   0.1 0.125 0.00125 0.873 0.0402  0.01   0.04    0.04
##  7     1     1   0.2   0.1 0.125 0.00125 0.873 0.0402  0.01   0.04    0.04
##  8     1     1   0.2   0.1 0.125 0.00125 0.873 0.0402  0.01   0.04    0.04
##  9     1     1   0.2   0.1 0.125 0.00125 0.872 0.0402  0.01   0.04    0.04
## 10     1     1   0.2   0.1 0.125 0.00125 0.872 0.0402  0.01   0.04    0.04
## # ... with 8,948 more rows, and 14 more variables: rsquxs <dbl>, rsquys <dbl>,
## #   rsqxys <dbl>, bxy <dbl>, bgx <dbl>, bux <dbl>, buy <dbl>, vg <dbl>,
## #   vu <dbl>, vex <dbl>, vey <dbl>, direction <chr>, d <dbl>, weight <dbl>
##
```

```
## $prop
## [1] 1
##
## $pl
```

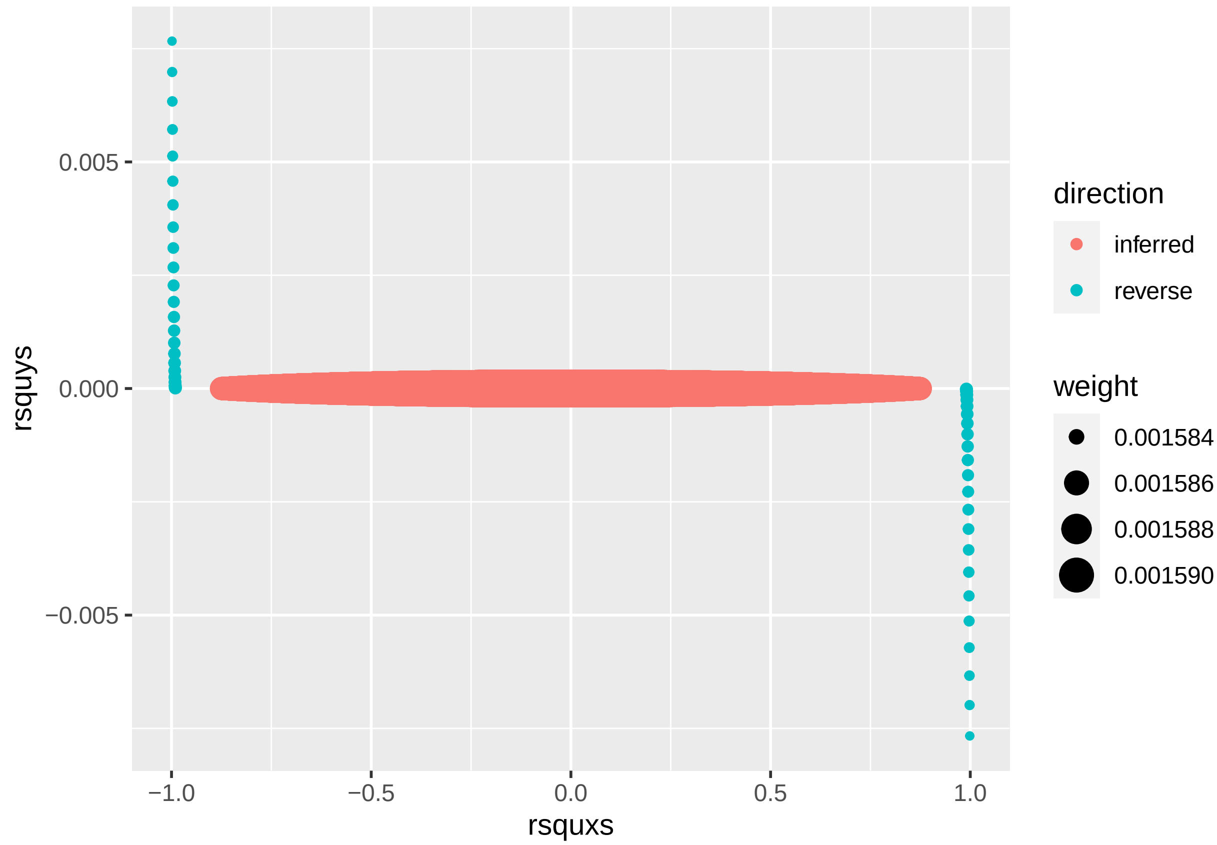


Same example but with more neutral weighting

```
r <- u_sensitivity(bxy=0.1, bxyo=0.1, bgx=0.5, vx=1, vy=1, vg=0.5, scaling=10)
r
```

```
## $result
## # A tibble: 10,044 x 25
## # Groups:   direction [2]
##        vx    vy  bxyo  conf rsqgx   rsqgy rsqux   rsquy rsqxy rsqxyo rsqxyos
##     <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>   <dbl> <dbl>  <dbl>   <dbl>
##  1      1     1   0.1     0 0.125 0.00125 0.875 0.00875  0.01   0.01    0.01
##  2      1     1   0.1     0 0.125 0.00125 0.875 0.00875  0.01   0.01    0.01
##  3      1     1   0.1     0 0.125 0.00125 0.874 0.00874  0.01   0.01    0.01
##  4      1     1   0.1     0 0.125 0.00125 0.874 0.00874  0.01   0.01    0.01
##  5      1     1   0.1     0 0.125 0.00125 0.874 0.00874  0.01   0.01    0.01
##  6      1     1   0.1     0 0.125 0.00125 0.873 0.00873  0.01   0.01    0.01
##  7      1     1   0.1     0 0.125 0.00125 0.873 0.00873  0.01   0.01    0.01
##  8      1     1   0.1     0 0.125 0.00125 0.873 0.00873  0.01   0.01    0.01
##  9      1     1   0.1     0 0.125 0.00125 0.872 0.00872  0.01   0.01    0.01
## 10      1     1   0.1     0 0.125 0.00125 0.872 0.00872  0.01   0.01    0.01
## # ... with 10,034 more rows, and 14 more variables: rsquxs <dbl>, rsquys <dbl>,
```

```
## #   rsqxys <dbl>, bxy <dbl>, bgx <dbl>, bux <dbl>, buy <dbl>, vg <dbl>,
## #   vu <dbl>, vex <dbl>, vey <dbl>, direction <chr>, d <dbl>, weight <dbl>
##
## $prop
## [1] 0.9867898
##
## $pl
```

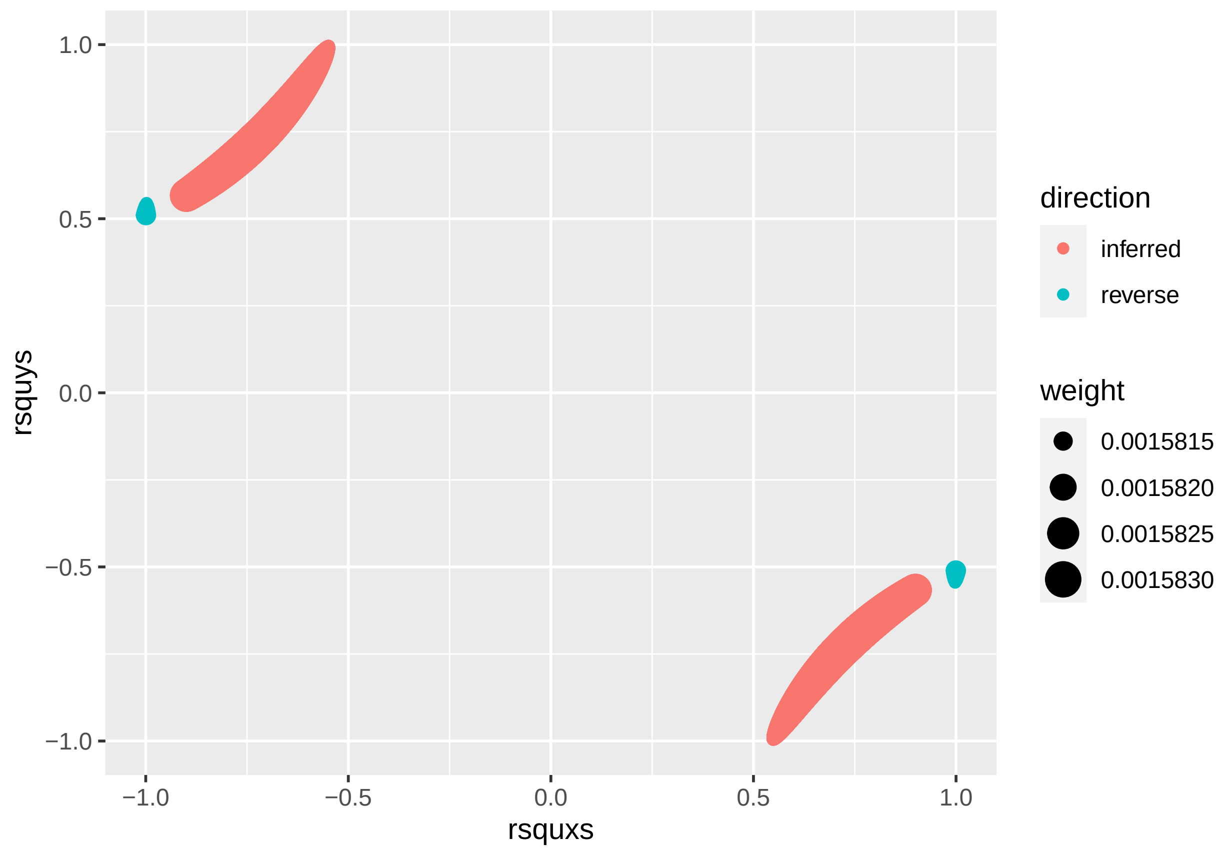


Example 2 - bxy and bxyo are very different

```
r <- u_sensitivity(bxy=0.1, bxyo=-1, vg=1, bgx=1, vx=10, vy=20, scaling=10)
r
```

```
## $result
## # A tibble: 2,212 x 25
## # Groups:   direction [2]
##        vx    vy  bxyo   conf rsqgx   rsqgy rsqux rsquy rsqxy rsqxyo rsqxyos rsquxs
##     <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>  <dbl>
##  1     10    20    -1   -1.1   0.1 0.0005 0.9   0.567 0.005    0.5    -0.5 -0.9
##  2     10    20    -1   -1.1   0.1 0.0005 0.900 0.567 0.005    0.5    -0.5 -0.900
##  3     10    20    -1   -1.1   0.1 0.0005 0.899 0.567 0.005    0.5    -0.5 -0.899
##  4     10    20    -1   -1.1   0.1 0.0005 0.899 0.568 0.005    0.5    -0.5 -0.899
##  5     10    20    -1   -1.1   0.1 0.0005 0.899 0.568 0.005    0.5    -0.5 -0.899
##  6     10    20    -1   -1.1   0.1 0.0005 0.898 0.568 0.005    0.5    -0.5 -0.898
##  7     10    20    -1   -1.1   0.1 0.0005 0.898 0.568 0.005    0.5    -0.5 -0.898
##  8     10    20    -1   -1.1   0.1 0.0005 0.897 0.569 0.005    0.5    -0.5 -0.897
```

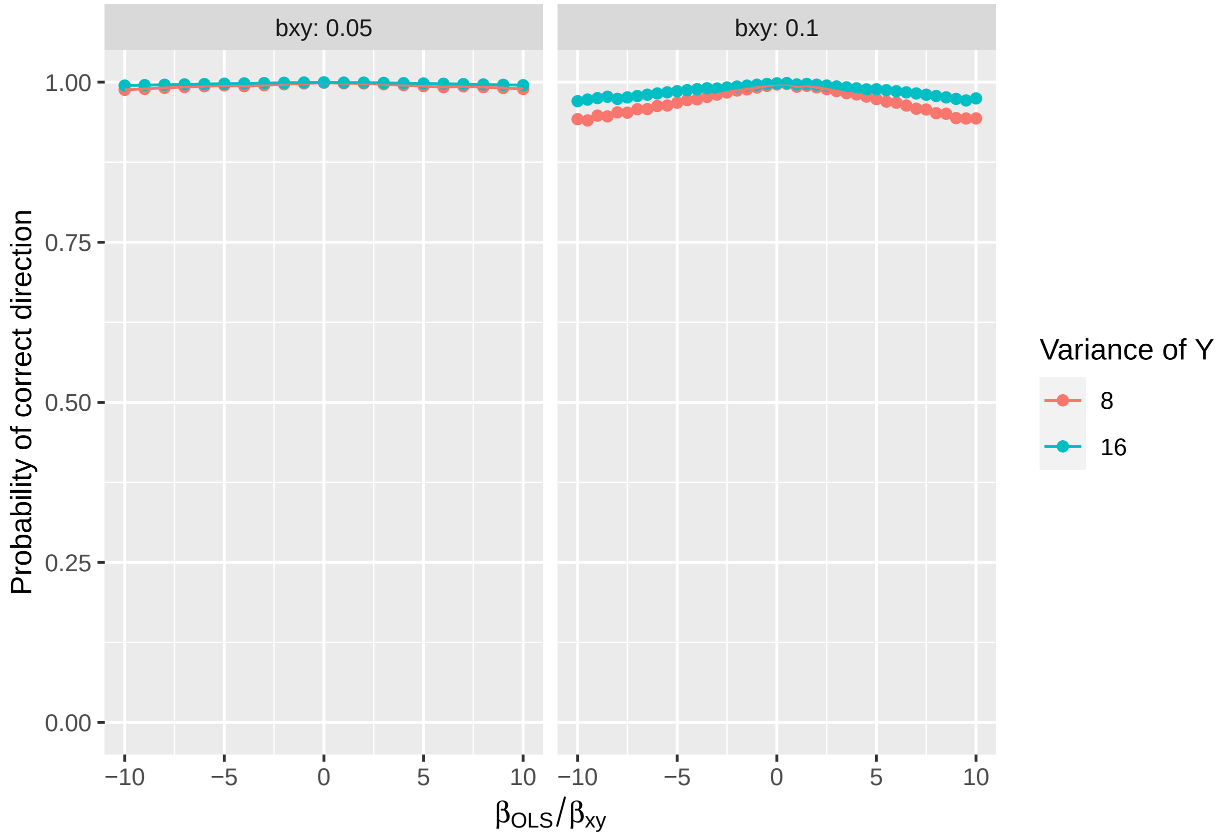
```
##  9     10     20     -1   -1.1    0.1 0.0005 0.897 0.569 0.005     0.5     -0.5 -0.897
## 10     10     20     -1   -1.1    0.1 0.0005 0.897 0.569 0.005     0.5     -0.5 -0.897
## # ... with 2,202 more rows, and 13 more variables: rsquys <dbl>, rsqxys <dbl>,
## #   bxy <dbl>, bgx <dbl>, bux <dbl>, buy <dbl>, vg <dbl>, vu <dbl>, vex <dbl>,
## #   vey <dbl>, direction <chr>, d <dbl>, weight <dbl>
##
## $prop
## [1] 0.9434405
##
## $pl
```



Explore general performance

```
param <- expand.grid(
    bxy = c(0.1, 0.05),
    bxyo = seq(-1, 1, by=0.05),
    vg=0.5,
    bgx = 0.01,
    vx = 4,
    vy = c(8,16),
    plot=FALSE,
    scaling=10
)
param$prop <- sapply(1:nrow(param), function(i) do.call(u_sensitivity, param[i,])$prop)
```

```
ggplot(param %>% filter(abs(bxyo / bxy) <=10), aes(bxyo / bxy, prop)) +
    geom_point(aes(colour=as.factor(vy))) +
    geom_line(aes(colour=as.factor(vy))) +
    ylim(c(0,1)) +
    facet_grid(. ~ bxy, labeller=label_both) +
    labs(y="Probability of correct direction", colour="Variance of Y", x=TeX(r'($\beta_{OLS}/\beta_{xy}$
```



For a particular analysis, the observational association needs to be substantially larger than the causal effect in order for there to be some chance of unmeasured confounding inferring the wrong causal direction.

Empirical analysis

BMI on SBP

```
library(TwoSampleMR)
a <- extract_instruments("ieu-a-2")
b <- extract_outcome_data(a$SNP, "ukb-b-19953") %>% convert_outcome_to_exposure()
```

```
## Extracting data for 79 SNP(s) from 1 GWAS(s)
```

```
c <- extract_outcome_data(a$SNP, "ukb-b-20175")
```

```
## Extracting data for 79 SNP(s) from 1 GWAS(s)
```

```r
d <- harmonise_data(b,c)
```

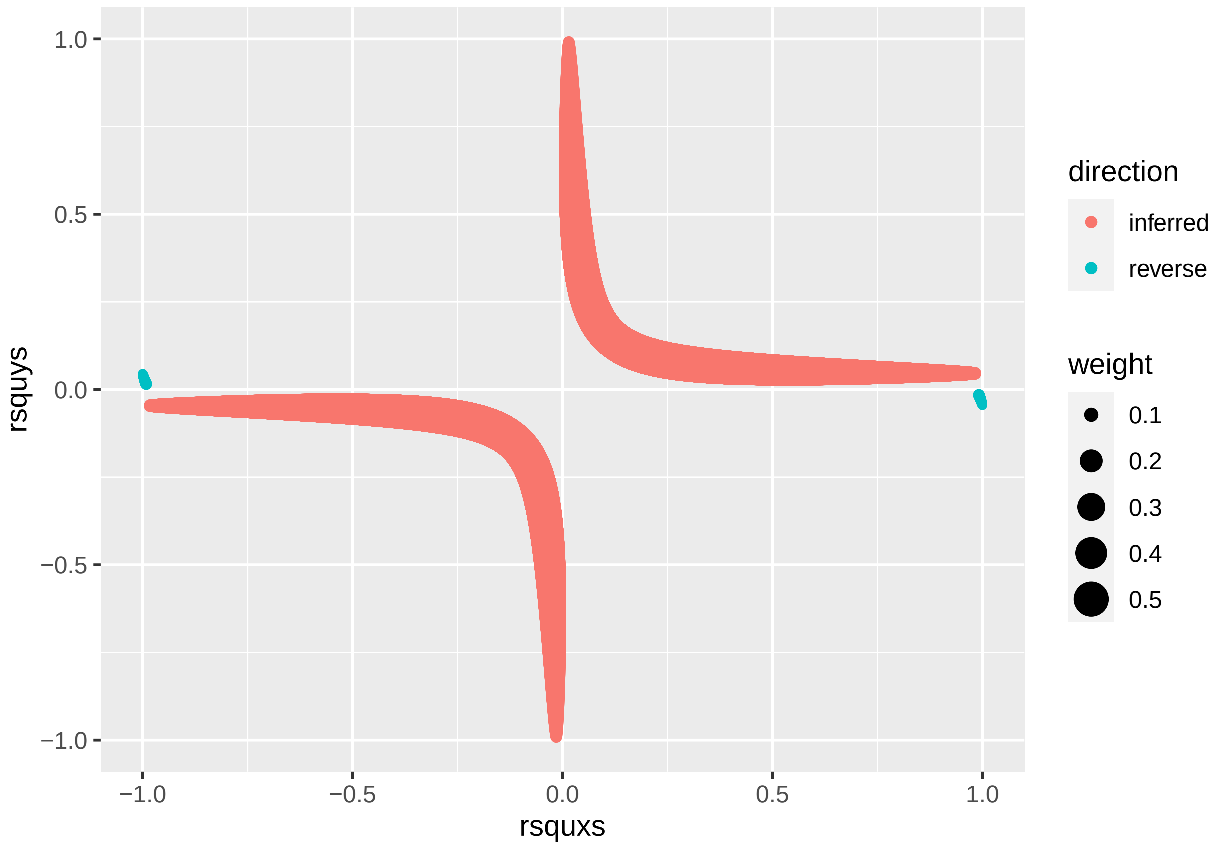
```
## Harmonising Body mass index (BMI) || id:ukb-b-19953 (ukb-b-19953) and Systolic blood pressure, automa
```

```r
d <- add_metadata(d)
d <- add_rsq(d)

bgx <- sqrt(sum(d$rsq.exposure))
bxy <- mr(d, method="mr_ivw")$b
```

```
## Analysing 'ukb-b-19953' on 'ukb-b-20175'
```

```r
# From https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6324286/
ols <- (0.8 + 1.7)/2 / 20.4 * 3.5
u_sensitivity(bxy=bxy, bxyo=ols, bgx=sqrt(sum(d$rsq.exposure)), vx=1, vy=1, vg=1, vu=1, scaling=0.5)
```

```
## $result
## # A tibble: 8,812 x 25
## # Groups:   direction [2]
##       vx    vy  bxyo  conf  rsqgx    rsqgy rsqux  rsquy   rsqxy rsqxyo rsqxyos
##    <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl> <dbl>  <dbl>   <dbl>  <dbl>   <dbl>
##  1     1     1 0.214 0.120 0.0179 0.000159 0.982 0.0461 0.00888 0.0460  0.0460
##  2     1     1 0.214 0.120 0.0179 0.000159 0.982 0.0461 0.00888 0.0460  0.0460
##  3     1     1 0.214 0.120 0.0179 0.000159 0.981 0.0461 0.00888 0.0460  0.0460
##  4     1     1 0.214 0.120 0.0179 0.000159 0.981 0.0461 0.00888 0.0460  0.0460
##  5     1     1 0.214 0.120 0.0179 0.000159 0.981 0.0461 0.00888 0.0460  0.0460
##  6     1     1 0.214 0.120 0.0179 0.000159 0.980 0.0461 0.00888 0.0460  0.0460
##  7     1     1 0.214 0.120 0.0179 0.000159 0.980 0.0461 0.00888 0.0460  0.0460
##  8     1     1 0.214 0.120 0.0179 0.000159 0.979 0.0461 0.00888 0.0460  0.0460
##  9     1     1 0.214 0.120 0.0179 0.000159 0.979 0.0461 0.00888 0.0460  0.0460
## 10     1     1 0.214 0.120 0.0179 0.000159 0.979 0.0461 0.00888 0.0460  0.0460
## # ... with 8,802 more rows, and 14 more variables: rsquxs <dbl>, rsquys <dbl>,
## #   rsqxys <dbl>, bxy <dbl>, bgx <dbl>, bux <dbl>, buy <dbl>, vg <dbl>,
## #   vu <dbl>, vex <dbl>, vey <dbl>, direction <chr>, d <dbl>, weight <dbl>
##
## $prop
## [1] 0.9955696
##
## $pl
```



1.0
0.5
0.0
−0.5
−1.0
rsquys
−1.0
−0.5
0.0
0.5
1.0
rsquxs
direction
inferred
reverse
weight
0.1
0.2
0.3
0.4
0.5