Supplementary Online Materials

eAppendix

Table of Contents

1. eTable 1: Characteristics of Eligible Mother-Child Pairs

	Total Eligible Sample (n=3,188)		**Vitamin D- PDP (n=1,971)**		**Vitamin D- Mother-reported ADHD (n=1,970)**		**Vitamin D – Teacher-reported ADHD (n=1,146)**	
Maternal Characteristics	*n*	*(%)*	*n*	*(%)*	*n*	*(%)*	*n*	*(%)*
Serum 25OHD								
<50 nmol/L	711	(22.3)	580	(29.4)	582	(29.5)	333	(29.1)
50-75 nmol/L	748	(23.5)	604	(30.7)	604	(30.6)	365	(31.8)
>= 75 nmol/L	971	(30.5)	786	(39.9)	785	(39.8)	448	(39.1)
Education Completed								
Primary or less	79	(2.5)	37	(1.9)	37	(1.9)	31	(2.7)
Secondary	1167	(36.6)	699	(35.5)	697	(35.4)	423	(36.9)
Higher	1841	(57.7)	1200	(60.9)	1203	(61.0)	675	(58.9)
Maternal Age at Intake - mean(sd)	31.7	(4.4)	31.6	(4.1)	31.62367	(4.1)	31.69603	(4.2)
Drinking during pregnancy								
None	849	(26.6)	559	(28.4)	558	(28.3)	329	(28.7)
Until pregnancy was known	434	(13.6)	319	(16.2)	319	(16.2)	185	(16.1)
After pregnancy was known	1321	(41.4)	923	(46.9)	924	(46.9)	539	(47.0)
Offspring Characteristics								
Mother-reported PDP symptoms	57	(1.8)	38	(1.9)	*N/A*	*N/A*	*N/A*	*N/A*
Mother-reported ADHD symptoms	110	(3.5)	*N/A*	*N/A*	69	(3.5)	*N/A*	*N/A*
Teacher-reported ADHD symptoms	64	(2.0)	*N/A*	*N/A*	*N/A*	*N/A*	40	(3.49)
Female	1586	(49.7)	992	(50.4)	994	(50.4)	563	(49.1)

2. Novel Visualization Methods for the Instrumental Inequalities

One disadvantage of other methods of representing the instrumental inequalities, like forest plots, heatmaps, and tables, is that the ordering in which SNP combinations appear is relatively arbitrary, and it can be difficult to identify consistent patterns, such as single SNP appearing in all sets which violate the instrumental inequalities. While traditional network graphs can somewhat improve this issue,

when the number of included SNPs grows large, these graphs begin to resemble "hairballs" and become increasingly difficult to interpret[44]. To ease interpretation, we developed a new visualization method for the instrumental inequalities, roughly based on BioFabric [44]. In these visualizations, each horizontal line represents a SNP, and each vertical line connects a set of SNPs proposed as instruments (with the number of included SNPs increasing from left to right). Each node thus represents a particular set of SNPs. In real data, the color of each node represents the value of the instrumental inequalities for a particular set of SNPs proposed jointly as instruments, with white indicating values ≤ 1, meaning the instrumental inequalities held, and darker colors indicating increasing maximum values of the instrumental inequalities.

In simulation studies, this same visualization can be used to visualize the number of simulations in which the instrumental inequalities failed to hold for a given set of simulated proposed instruments. In that setting, the color of the nodes would represent the number of simulations in which the instrumental inequalities were violated for each set of variables jointly proposed as instruments, with darker colors indicating increasing numbers of simulations in which the instrumental inequalities were violated, rather than the value of the instrumental inequalities for a particular set of SNPs jointly proposed as instruments. One benefit of these visualizations is that they provide a simpler and less dense means of representing the values of the instrumental inequalities for large numbers of SNPs than tables. For very large numbers of SNPs, future research in this area might consider reducing computational burden by eliminating calculations of the inequalities for sets of SNPs containing subsets that had already violated the instrumental inequalities and marking such sets with a unique color on the resulting visualization.

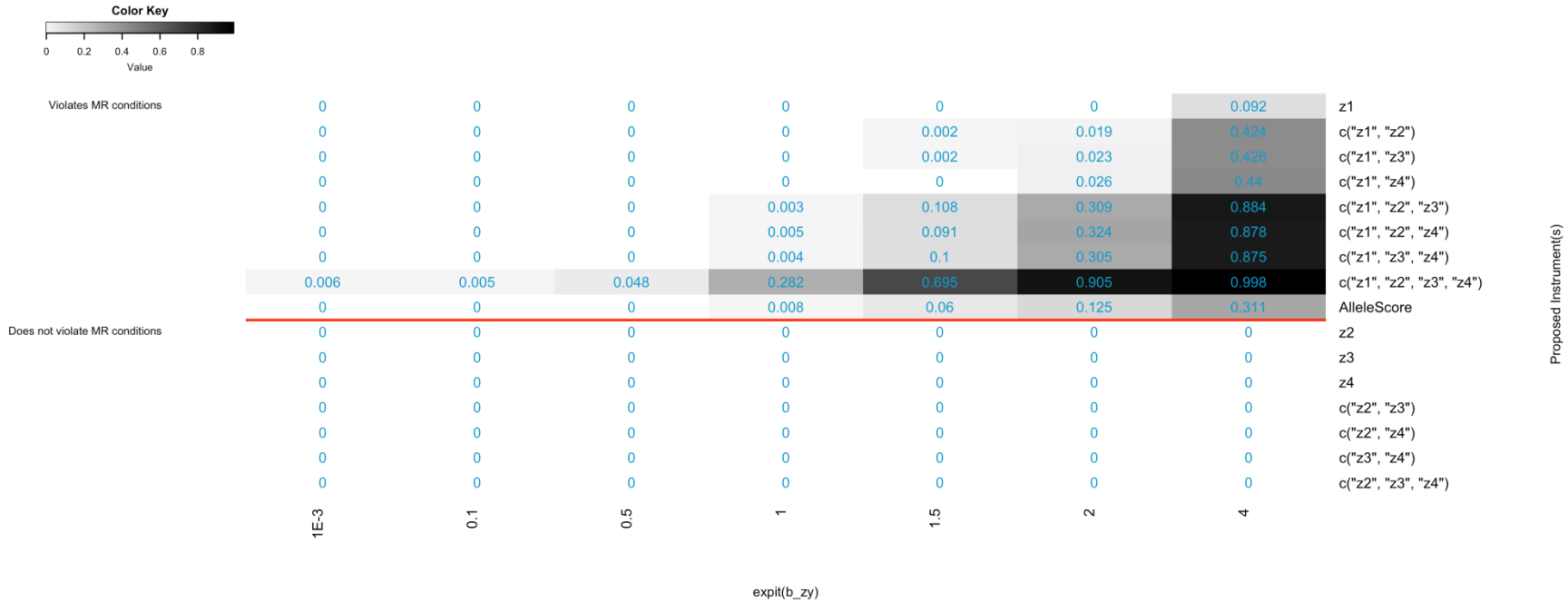
One notable advantage of this visualization technique is that it allows for easier identification of a consistent pattern of violations of the MR assumptions originating from a single SNP. As we can see in Figure 4 **D**, when all violations are of sufficient magnitude, and originate from a single SNP (Z_1), we see a single dark horizontal line (a SNP where the instrumental inequalities were violated for most or all sets of SNPs jointly proposed as instruments including that particular SNP), and inconsistent dark patterns across the other SNPs (showing violations only in sets of SNPs jointly proposed as instruments including the problem SNP). This contrasts with Figure 4 **C**, where we only see violations of the instrumental inequalities when Z_1, Z_2, Z_3, and Z_4 are all jointly proposed as instruments. In Figure 4 **C**, we do not have enough evidence to suggest that violations of the MR assumptions arise from a single SNP, only that the MR conditions cannot hold for all 4 variables jointly proposed as instruments in the sample.

3. Details of Simulation Parameters

We conducted simulations of a relationship between 4 binary proposed instruments (Z_1, Z_2, Z_3, and Z_4), a binary exposure X, and a binary outcome Y, where the relationship between X and Y was confounded by a continuous variable U, and the proposed instrument Z_1 was an invalid instrument with a direct effect (β_2) on the outcome Y. Each simulation was constructed such that Z_{1i}~bernoulli(0.5), Z_{2i}~bernoulli(0.5), Z_{3i}~bernoulli(0.5), Z_{4i}~bernoulli(0.5), U_i~norm(0,1), X_i~bernoulli(expit(0.6+0.1*U_i+β_1*Z_{1i}+β_1*Z_{2i}+β_1*Z_{3i}+β_1*Z_{4i})), and

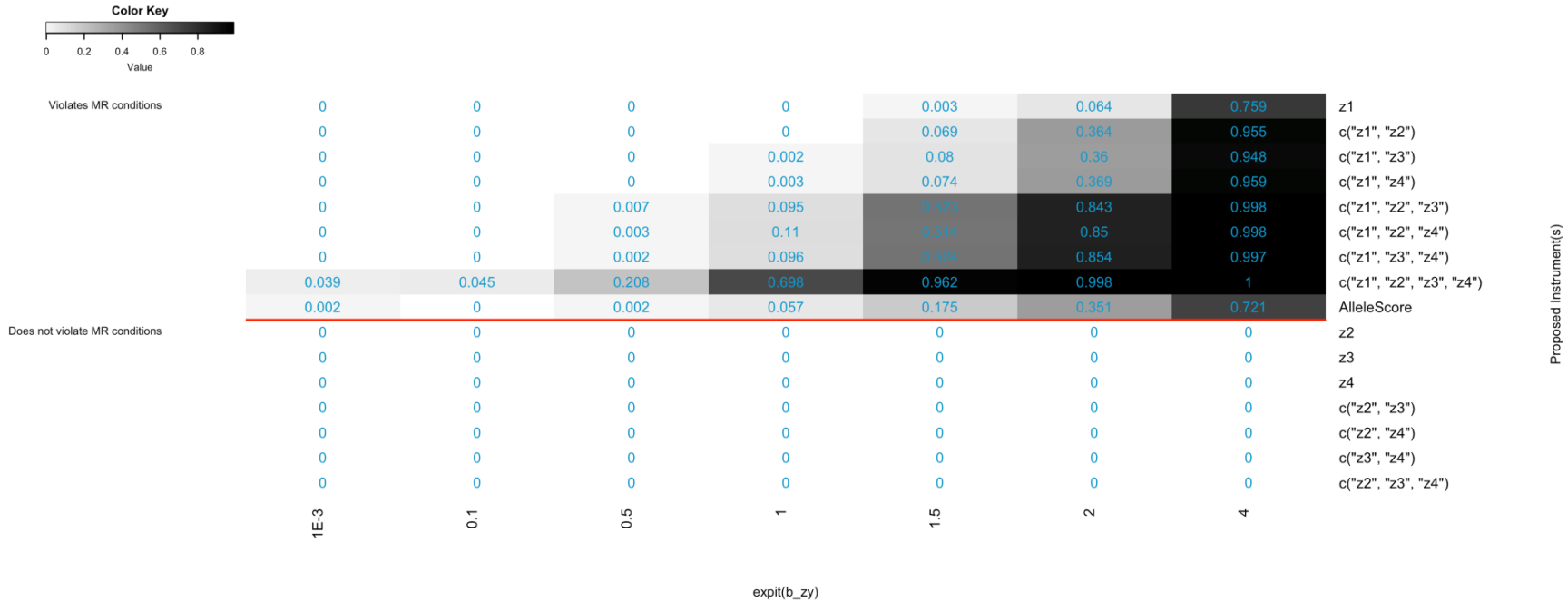
Y_i~bernoulli(expit(0.02+0.1*U_i+β_2*Z_{1i})). In order to examine the effects of changing sample size and varying magnitudes of violation of the MR assumptions on the instrumental inequalities, we varied simulations across 3 samples sizes (1,000 individuals, 10,000 individuals, 100,000 individuals), 4 possible instrument strengths (β_1= 0.01, 0.1, 0.5, and 1.0, corresponding roughly to risk differences of 0.003, 0.021, 0.071, 0.079), and 7 possible strengths of violations of the MR assumptions (β_2 = 0.01, 0.1, 0.5, 1, 1.5, 2, 4, resulting in violation strengths on the risk difference scale of 0.001, 0.025, 0.121, 0.189, 0.230, 0.315, 0.377, and 0.478). For each combination of sample size, instrument strength, and magnitude of direct path violation, we conducted 1,000 simulations.

eFigure 1. Results of instrumental inequalities for 1000 simulations of samples of 1,000 individuals with effect of each proposed instrument on exposure 0.003 (risk difference scale)



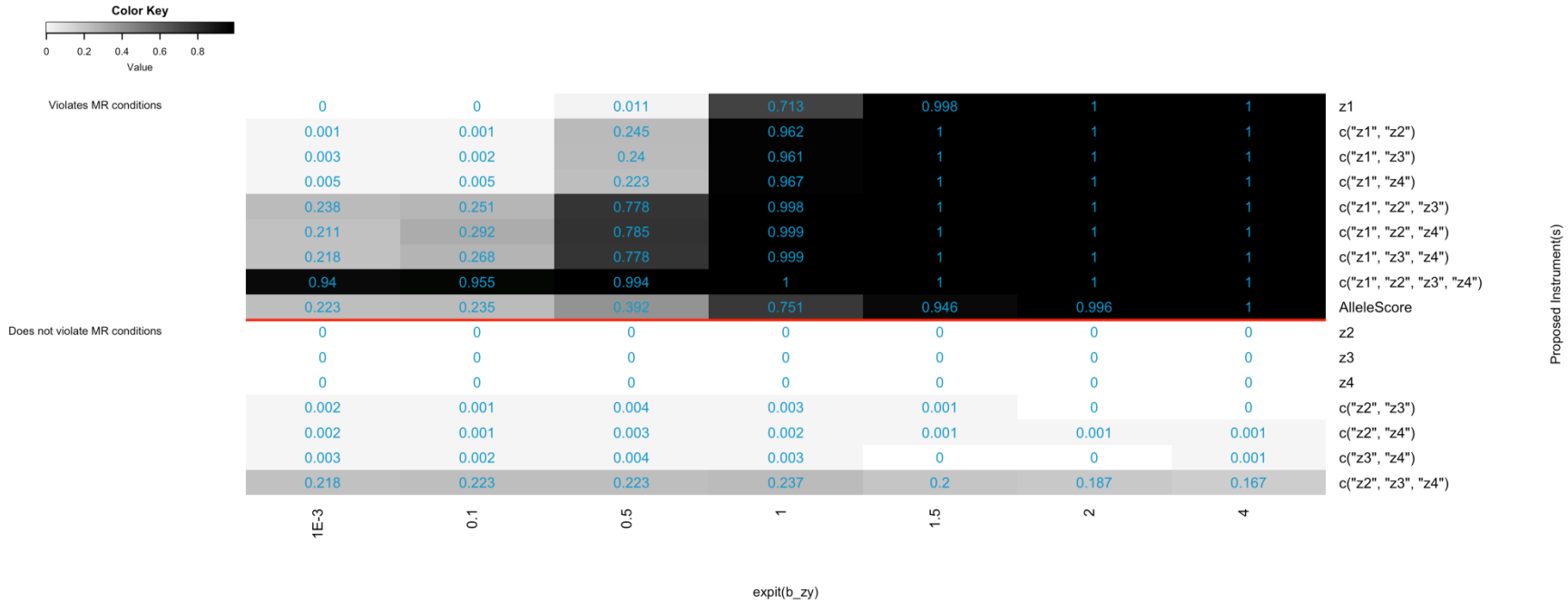
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 2. Results of instrumental inequalities for 1000 simulations of samples of 1,000 individuals with effect of each proposed instrument on exposure 0.021 (risk difference scale)



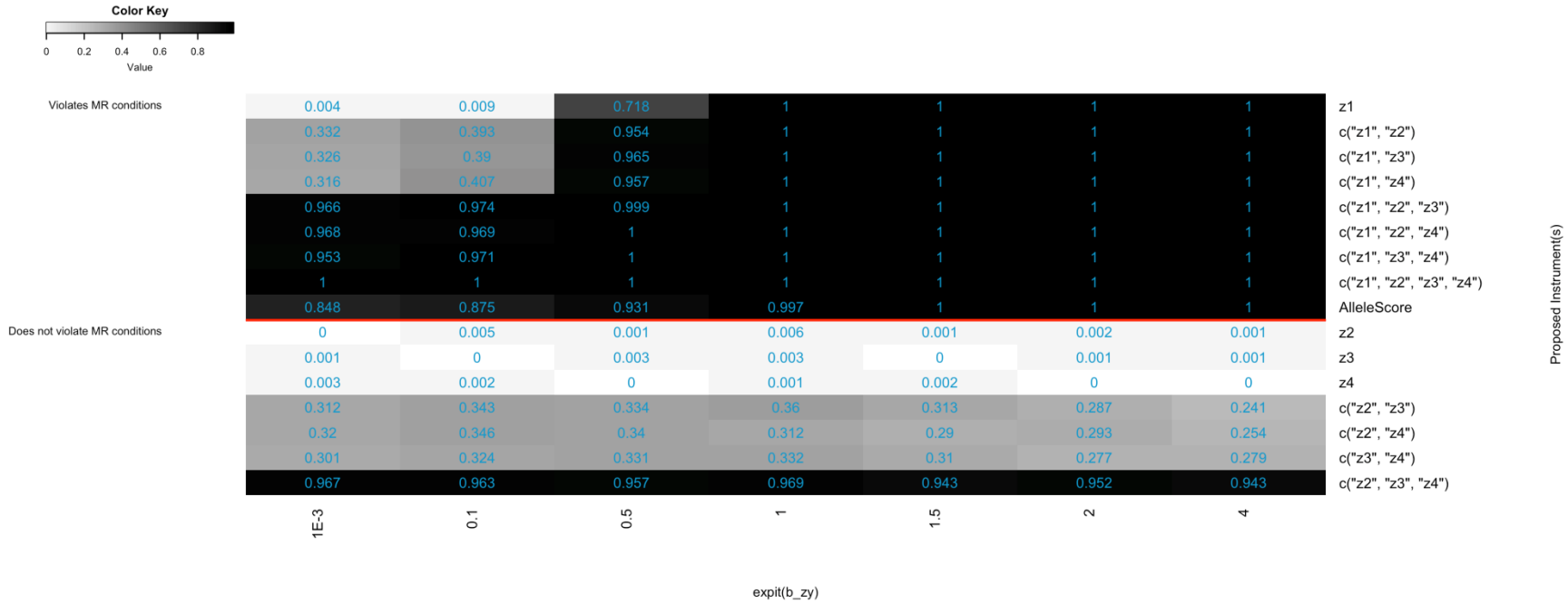
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 3. Results of instrumental inequalities for 1000 simulations of samples of 1,000 individuals with effect of each proposed instrument on exposure 0.071 (risk difference scale)



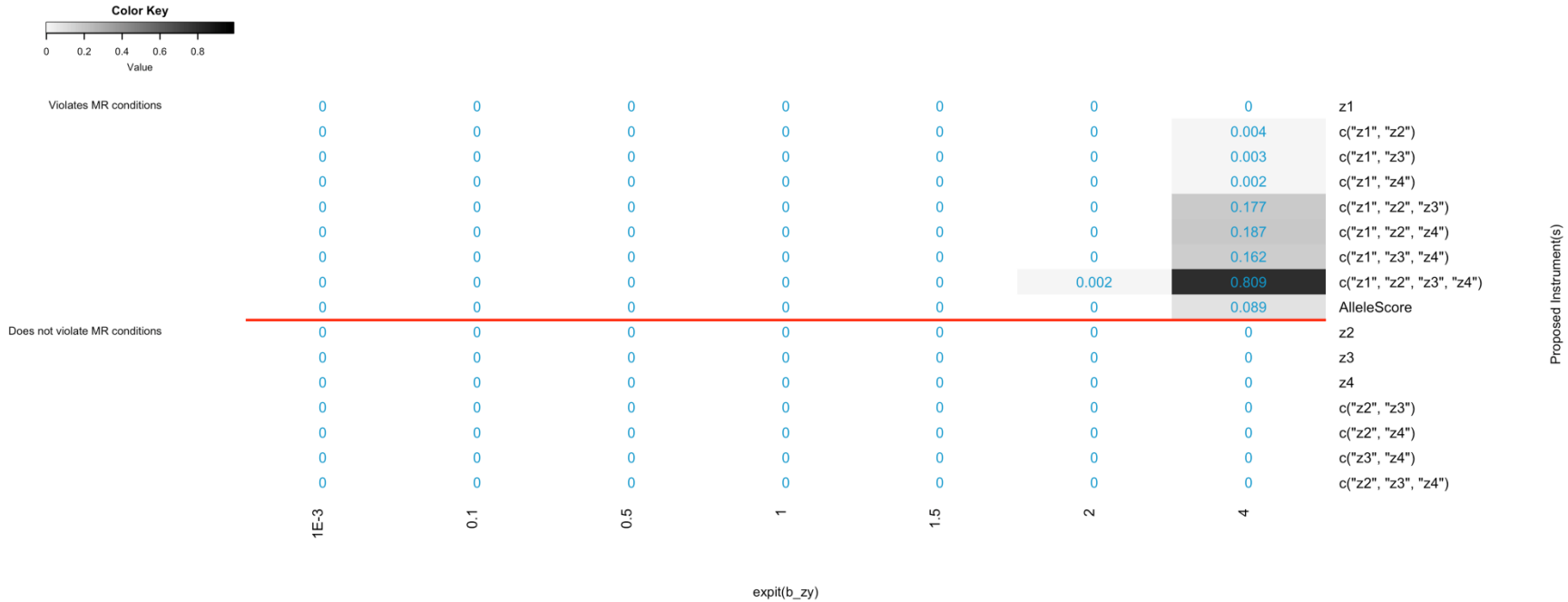
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 4. Results of instrumental inequalities for 1000 simulations of samples of 1,000 individuals with effect of each proposed instrument on exposure 0.079 (risk difference scale)



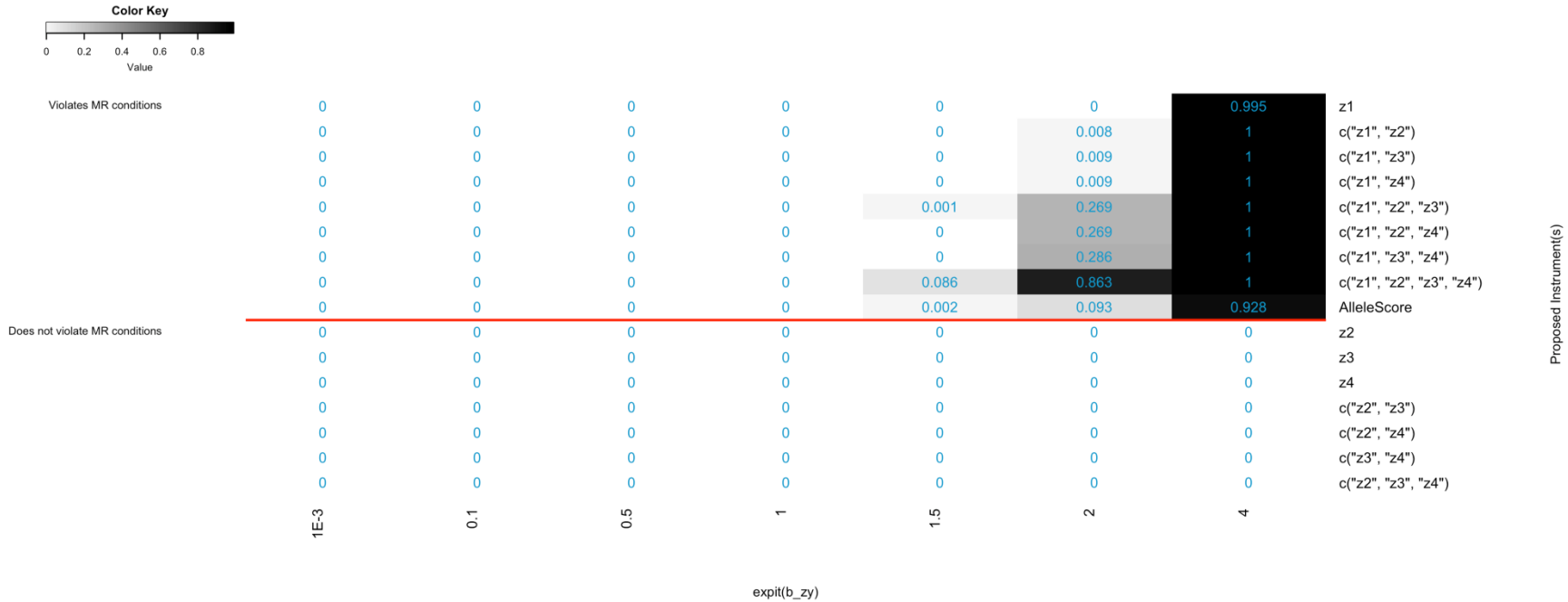
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 5. Results of instrumental inequalities for 1000 simulations of samples of 10,000 individuals with effect of each proposed instrument on exposure 0.003 (risk difference scale)



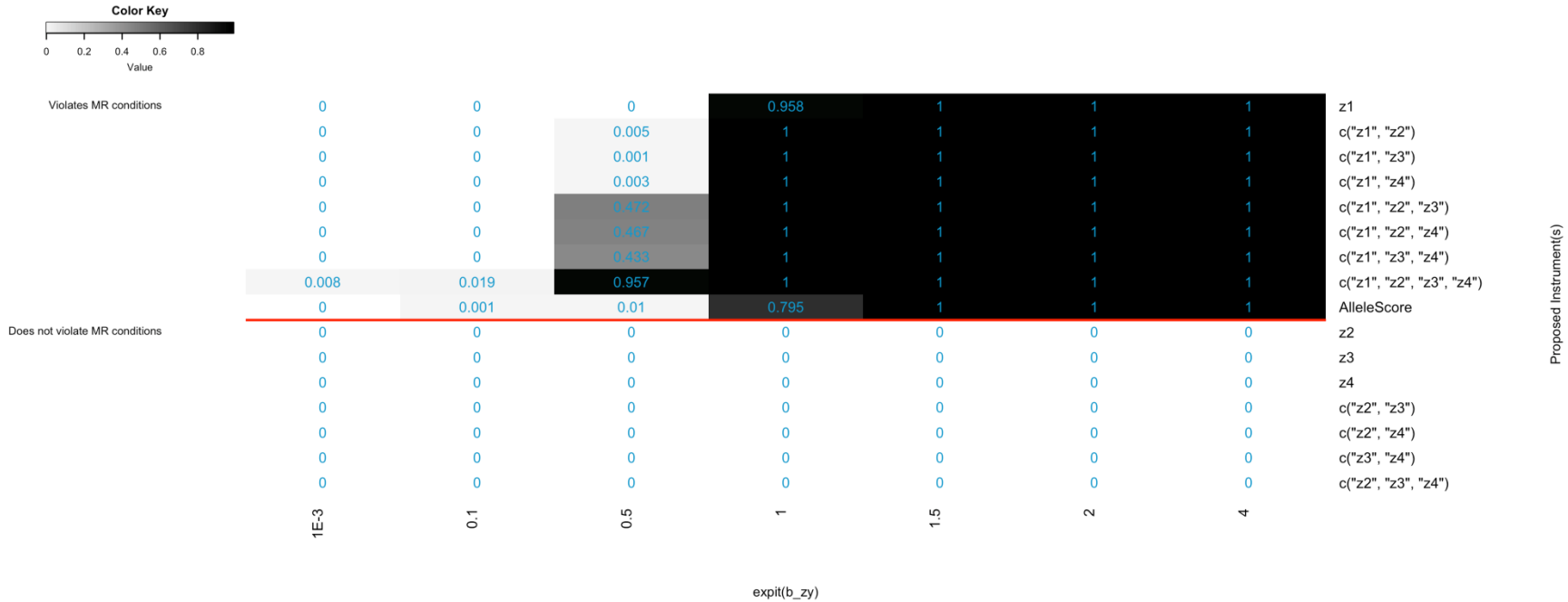
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 6. Results of instrumental inequalities for 1000 simulations of samples of 10,000 individuals with effect of each proposed instrument on exposure 0.021 (risk difference scale)



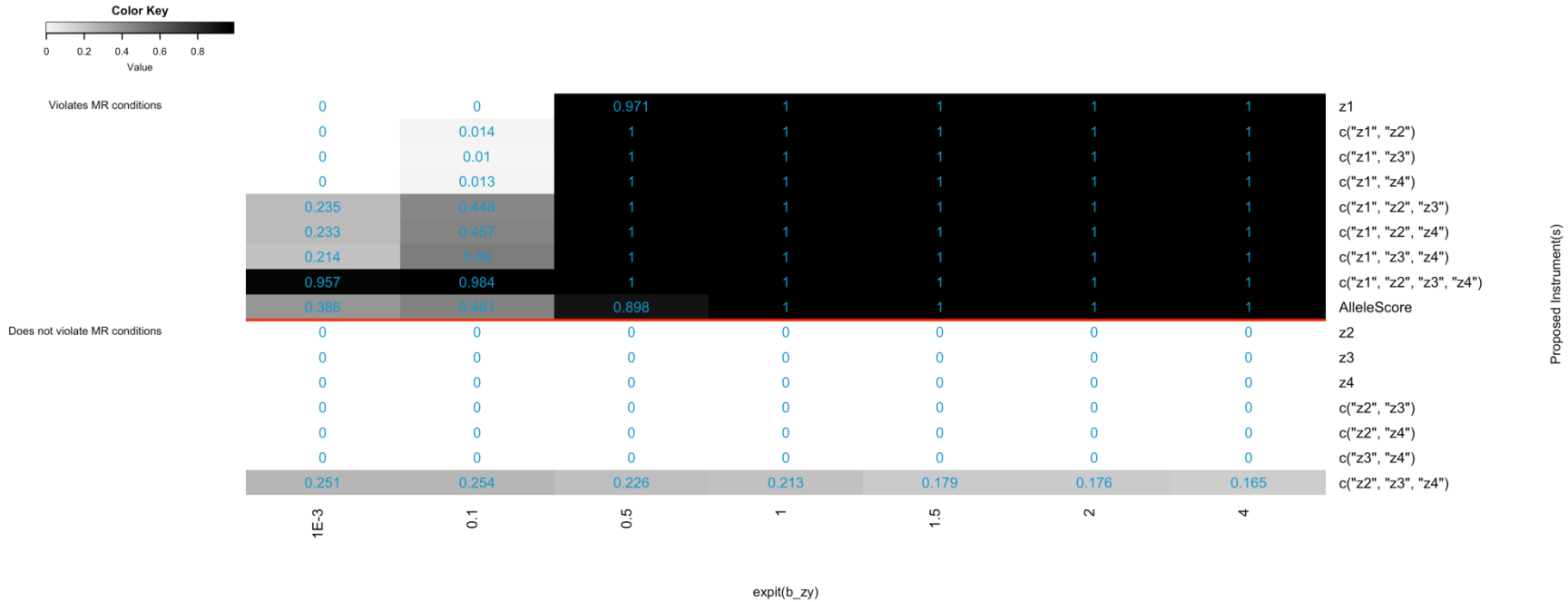
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 7. Results of instrumental inequalities for 1000 simulations of samples of 10,000 individuals with effect of each proposed instrument on exposure 0.071 (risk difference scale)



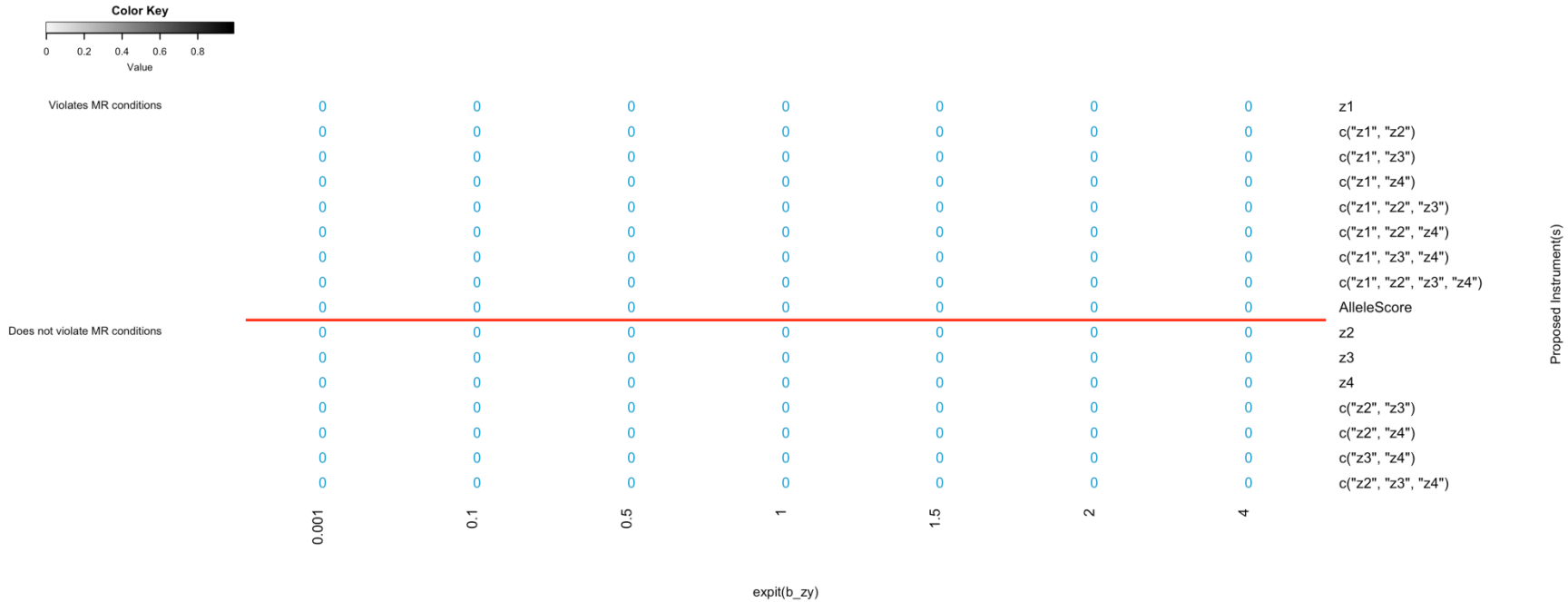
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 8. Results of instrumental inequalities for 1000 simulations of samples of 10,000 individuals with effect of each proposed instrument on exposure 0.079 (risk difference scale)



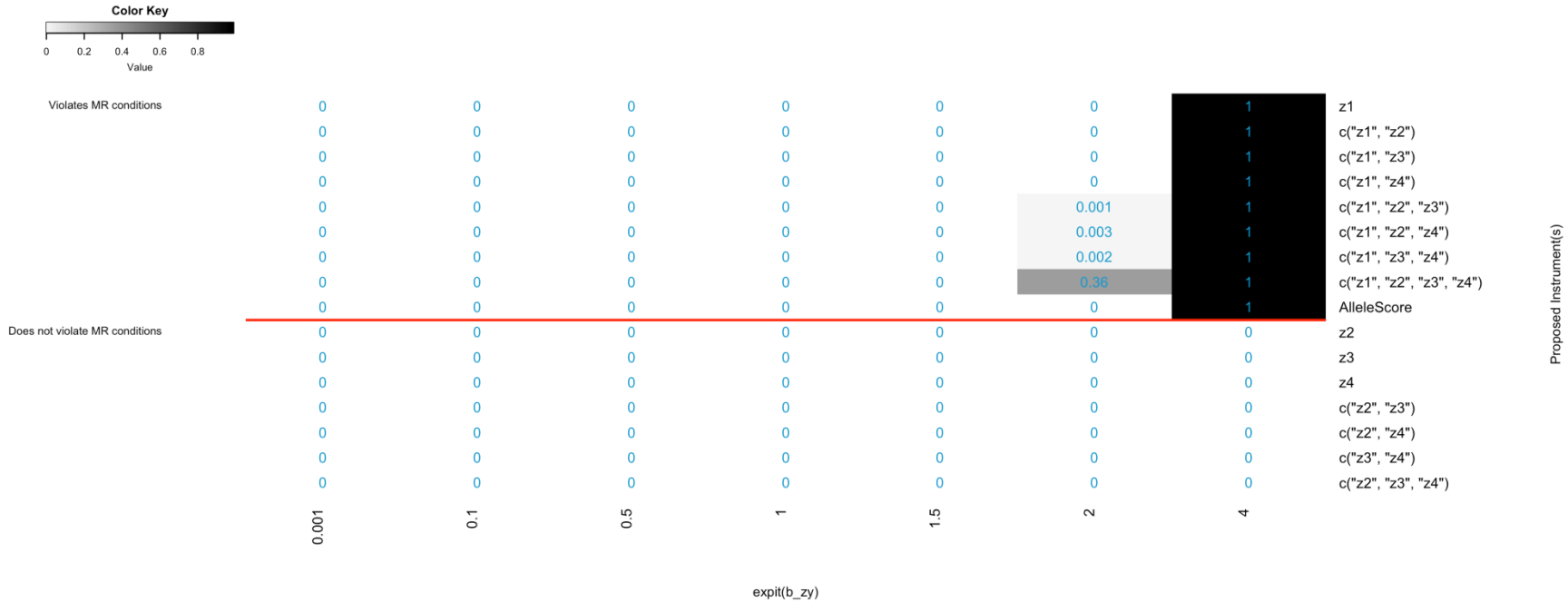
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 9. Results of instrumental inequalities for 1000 simulations of samples of 100,000 individuals with effect of each proposed instrument on exposure 0.003 (risk difference scale)



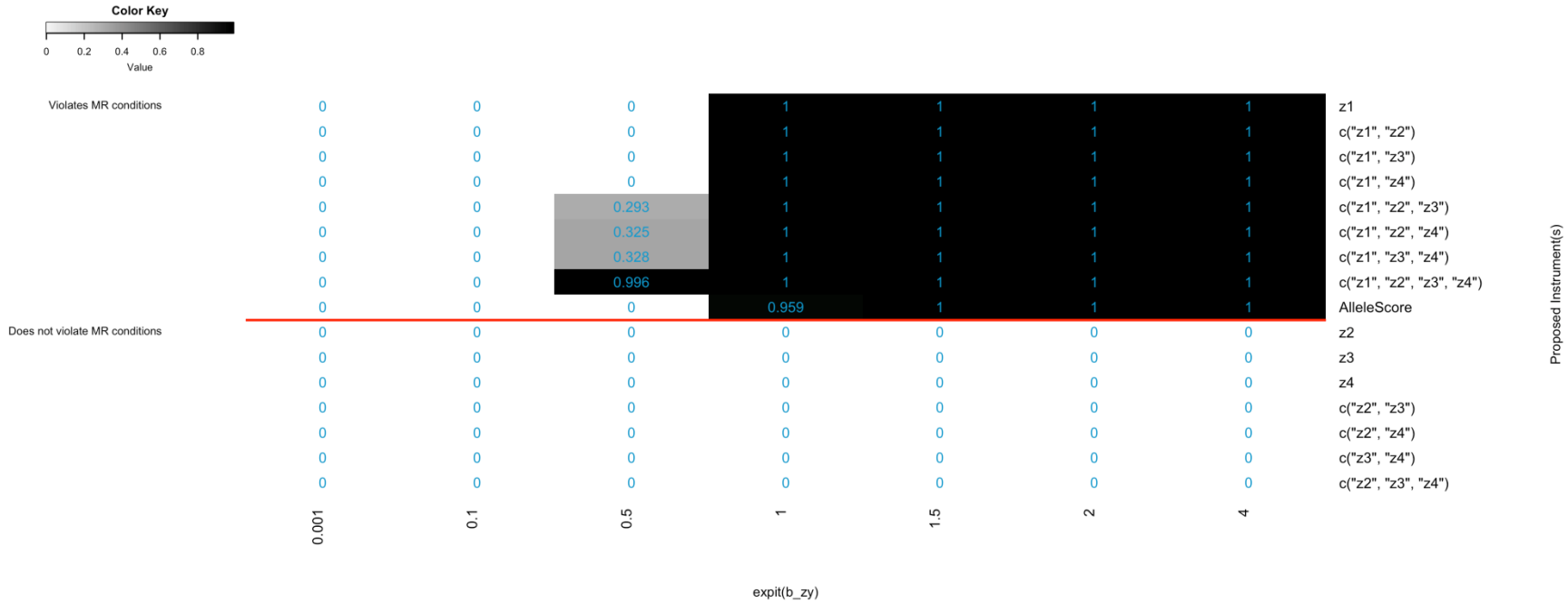
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 10. Results of instrumental inequalities for 1000 simulations of samples of 100,000 individuals with effect of each proposed instrument on exposure 0.021 (risk difference scale)



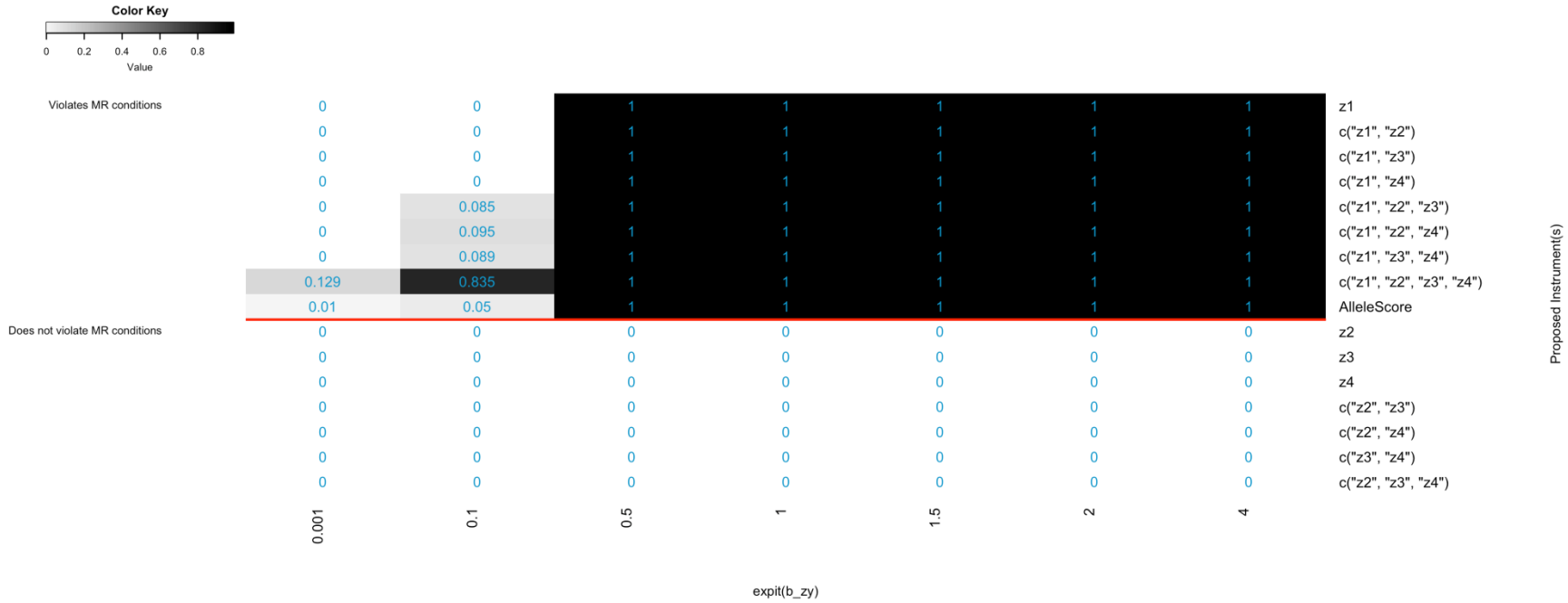
Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 11. Results of instrumental inequalities for 1000 simulations of samples of 100,000 individuals with effect of each proposed instrument on exposure 0.071 (risk difference scale)



Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

eFigure 12. Results of instrumental inequalities for 1000 simulations of samples of 100,000 individuals with effect of each proposed instrument on exposure 0.079 (risk difference scale)



Heatmap showing proportion of 1000 simulations for which the instrumental inequalities failed to hold across possible combinations of the four variables (Y axis) and across increasing size of effect of Z1 on the outcome Y (X axis). Proposed instruments containing Z1, which violate the MR conditions, are shown above the red line. Proposed instruments for which the MR conditions hold are shown below the red line.

5. Possible sources of structural violations of the MR conditions within the data example

Pleiotropy, in which genetic loci affect multiple traits, violating the 2nd assumption, is one of the most commonly noted sources of potential bias in MR (Figure 5A)[7,8,11]. Although we restricted our sample to mothers of European ancestry, it is possible that this strategy did not adequately control for population stratification, or that our sample contained substantial cryptic relatedness, both of which could result in assumption violations (Figure 5B). Previous research has also found that the required assumptions can be violated for MR analyses proposing maternal genetic factors as instruments for the effect of pregnancy exposures on offspring outcomes if the offspring's own genotype has a causal effect on the outcome, the mother's exposure status continues to affect the offspring after birth, or if the association between maternal genotype and vitamin D status changed over the course of pregnancy (Figures 5C, 5D, 5E)[39, 40, 41]. In addition, if Vitamin D exposure impacted fertility or ability to carry a pregnancy to term, the MR assumptions could be violated by selection bias resulting from conditioning on live birth (Figure 5F). As previously mentioned, categorization of a truly continuous exposure, which is necessary for the use of the instrumental inequalities, can also violate the assumptions of an MR analysis (Figure 5G) [39]. If maternal genotype is related to missingness of exposure or outcome data, the MR assumptions could be violated by our use of complete case analysis (Figure 5H). These possible sources of bias are not mutually exclusive, and all could be present in our data at some level.

6. R code for instrumental inequalities with multicategorical instruments, exposures, and outcomes

```
## Functions to apply and visualize the instrumental inequalities
with multiple proposed instruments
## Created by: Elizabeth Diemer, Jeremy Labrecque
## Date last Edited: 10/2/2018

##loading required packages
# install.packages("tidyverse")
# install.packages("xlsx")
library(foreign)
library(tidyverse)
library(xlsx)

## Creating internal function to get maximum value of the
instrumental inequalities for single given joint instrument
## NOTE: ALL VARIABLES CAN BE MULTICATEGORICAL BUT CANNOT BE
CONTINUOUS
## ARGUMENTS
## data: names of dataset (data.frame)
## instrument: name of instrument variable (character)
## x: name of exposure variable (character)
## y: name of outcome variable (character)

run_instrumental_inequalities_singlejointiv <- function(data, y, x,
instrument){

  # Set variable names
  data$Y <- data[[y]]
  data$X <- data[[x]]
  data$IV <- data[[instrument]]
  n_uniq_Y <- length(unique(data$Y))

  # Creating matrix with all possible combinations of proposed IV
and exposure
  com <- lapply(1:n_uniq_Y, function(k) {
    unique(data$IV)
  })
  com[[length(com)+1]] <- unique(data$X)
  com <- expand.grid(com, stringsAsFactors = FALSE)
  names(com) <- c(paste0("IV", 1:n_uniq_Y),"X")

  # Filling in com matrix with proportions from the data - values of
instrumental inequalities for different combinations
  prp <- lapply(1:n_uniq_Y, function(i) {
    sapply(1:nrow(com), function(j) {
      sum(data$IV==com[j, i] & data$X==com$X[j] &
data$Y==unique(data$Y)[i])/sum(data$IV==com[j, i])
```

```
    })
  }) %>% do.call(cbind, .) %>% as.data.frame
  names(prp) <- paste0("Y", unique(data$Y), "_IV", 1:n_uniq_Y)
  prp$sum_prop <- rowSums(prp)

  # Combine com matrix with proportions and values of instrumental
inequalities
  com <- cbind(com, prp)
  return(com)
}



## Function applying the instrumental inequalities for a given
exposure-outcome
## pair across all combinations of multiple proposed instruments.
##
## ARGUMENTS:
## datasetname: the dataset to be used (data.frame)
## IV: a character vector containing the names of the variables
proposed as instruments (character vector)
## exposure: the name of the exposure of interest (character)
## outcome: the name of the outcome of interest (character)
##
## The function outputs a list of 4 results - the first is a summary
table of the
## findings, and the second, third, and fourth are information
necessary for the
## creation of the bfi graph visualizations of the results.
## The function will remove any rows with missing data.

instrumental_inequalities_multiv <- function(datasetname, IV,
exposure, outcome){

  k <- length(IV)

  # create list of sets of instruments
  mylist <- lapply(seq_along(IV), function(i) combn(IV, i, FUN =
list))

  mylist <- flatten(mylist)

  # check GRS and add to list
  datasetname$AlleleScore <- apply(datasetname[IV], 1, sum)
  mylist <- c(mylist, "AlleleScore")

  # create summary table
  summarydat <- matrix(nrow=length(mylist), ncol=8)
  colnames(summarydat) <- c('Nonzero Cell Count', 'Smallest Cell',
                            'Number Cells greater than 10','Bonet
trichotomous inequality holds?',
```

```
                              'Balke-Pearl IV Inequalities Hold?', 'BP
Inequalities Max Value',
                              'BP Violating Strata of Instrument', 'BP
Violating Exposure Level')

   rownames(summarydat) <- c(mylist)


        # create variable and run results for each possible
combination of proposed IVs
  for (i in 1:length(mylist)){
        dat <- datasetname %>% select(IV, everything())
        dat$a <- dat[[exposure]]
        dat$y <- dat[[outcome]]
        dat <- dat%>% select(IV, AlleleScore, a, y)
        dat <- dat %>% drop_na()
        n_uniq_Y <- length(unique(dat$Y))

        #create new joint variable
        IVT = mylist[i]
        dat <- unite_(dat,"jointIV", flatten(IVT), remove = FALSE)

        #running instrumental inequalities function
        combo <-
run_instrumental_inequalities_singlejointiv(data=dat,
                                                                    y="y",
x="a",

instrument="jointIV")
        ineq <- aggregate(sum_prop ~ IV1, data=combo,
FUN=max)$sum_prop %>% max

        #print IV inequalities held or no
        summarydat[i, 5] <- if (ineq<=1) {print("yes")} else
{print("no")}
        summarydat[i, 6] <- ineq

        #creates dataset of violating strata
        combo1 <- combo %>% filter(sum_prop > 1)

        #generate list of jointIVs to be printed in violating strata
        summarydat[i, 7] <- ifelse(ineq <= 1, print("none"),
                                   paste(list(unique(flatten(flatten(
                                     lapply(1:n_uniq_Y, function(j)
{combo1[j]}))))))))

        #number cells, smallest cells, count cells under 10
        ftable <- rle(sort(dat$jointIV))
        summarydat[i, 1] <- length(ftable$lengths)
        summarydat[i, 2] <- min(ftable$lengths)
        summarydat[i, 3] <- sum(ftable$lengths >= 10)
```

```
        #exposure level violated
        summarydat[i, 8] <- if (ineq <= 1) {print("none")}
                            else
{print(paste(list(unique(combo1$X))))}

        #Bonet trichotomous instrument inequality
        #only eligible if binary exposure and outcome, trichotomous
instrument
        triineq <- ifelse(length(unique(dat$a)) > 2, "NA",
                        ifelse(length(unique(dat$y)) > 2, "NA",
                               ifelse(length(unique(dat$jointIV)) >
3, "NA",
                                      sum(with(dat,a==min(a) &
y==max(y) & jointIV==max(as.numeric(jointIV))-1)) / sum(with(dat,
jointIV==max(as.numeric(jointIV))-1))+
                                        sum(with(dat, a==min(a) &
y==min(y) & jointIV==max(as.numeric(jointIV)))) / sum(with(dat,
jointIV==max(as.numeric(jointIV))))+
                                        sum(with(dat, a==min(a) &
y==max(y) & jointIV==min(as.numeric(jointIV)))) / sum(with(dat,
jointIV==min(as.numeric(jointIV))))+
                                        sum(with(dat, a==max(a) &
y==max(y) & jointIV==max(as.numeric(jointIV))-1)) / sum(with(dat,
jointIV==max(as.numeric(jointIV))-1))+
                                        sum(with(dat, a==max(a) &
y==min(y) & jointIV==min(as.numeric(jointIV)))) / sum(with(dat,
jointIV==min(as.numeric(jointIV)))) )))
        summarydat[i, 4] <- ifelse(triineq<=2, print("yes"),
print(triineq))
  }

  resultslist <- list(summarydat, mylist, k, unname(summarydat[,
6]))
  return(resultslist)
}



##Function to create instrumental inequality plots
##
##ARGUMENTS:
##instru: list containing names of all possible combinations of
variables -
##resultslist[[2]] from instrumental_inequalities_multiv function
(list)
##k: number of variable jointly proposed as instruments -
resultslist[[3]]
##from instrumental_inequalities_multiv function (integer)
##ineqs: vector of maximum value of the instrumental inequalities
for
##each combination of variables - resultslist[[4]] from
instrumental_inequalities_multiv function (vector)
```

```
##title: optional title of plot (character)
##
##Required arguments are supplied by
instrumental_inequalities_multiv function as second, third, and
##fourth objects on output list of results. The required inputs from
the instrumental_inequalities_multiv
##function should be double bracketed.

plot_instrumental_inequalities <- function(instru,k,ineqs, title){

  ##determining title
  if(missing(title)){title<-NA}

  ##generating nodes dataset
  nodes <- data.frame(id=1:length(flatten(instru)))
  nodes$label1 <- lapply(1:length(flatten(instru)),
function(i){flatten(instru)[[i]]})

  ##generate y position - aligning along y spots
  nodes$y <- 1
  for (j in 1:k){nodes<-nodes %>%
mutate(y=ifelse(label1==instru[[j]], j, y))}
  nodes$y <- as.numeric(nodes$y)
  nodes$y <- nodes$y*10

  ##label of group - creates their x axis coordinates
  nodes$x <- flatten(lapply(1:length(instru),
function(i){rep(length(instru[1:i]),

length(flatten(instru[i])))}))
  nodes$x <- unlist(nodes$x)
  nodes$x <- nodes$x*10

  ##edges generation
  edges <- data.frame(id=1:length(unique(nodes$y)))
  edges$fromy <- unique(nodes$y)
  edges$fromx <- tapply(nodes$x, nodes$y, min)
  edges$toy <- edges$fromy
  edges$tox <- tapply(nodes$x, nodes$y, max)
  vertedges <- data.frame(id=1:length(unique(nodes$x)))
  vertedges$fromx <- unique(nodes$x)
  vertedges$tox <- vertedges$fromx
  vertedges$fromy <- tapply(nodes$y, nodes$x, min)
  vertedges$toy <- tapply(nodes$y, nodes$x, max)
  edges <- rbind(edges, vertedges)

  ## size and color of nodes
  printsumnum <- function(i){print(ineqs[i])}
  nodes <- nodes %>% rowwise %>%
mutate(colorfactor=printsumnum(x/10))
  nodes$ii <- ifelse(nodes$colorfactor<=1, NA,
cut(as.numeric(nodes$colorfactor),
```

```
                                                    breaks=seq(1, 2,
len=100),

include.lowest=TRUE))
  nodes$color <- ifelse(nodes$colorfactor<=1,
colorRampPalette("grey99")(1),
                        colorRampPalette(c("gray60",
"gray22"))(99)[nodes$ii])

  ##node labels
  nodes$label1 <- ifelse(nodes$id<=k, nodes$label1,
ifelse(nodes$id==length(flatten(instru)), nodes$label1, ""))
  ##generating plot itself
  layout(matrix(1:2,nrow=1),widths=c(0.8,0.2))
  par(mar=c(5.1,4.1,4.1,1.0))
  plot(c(-2,max(nodes$x)+5),c(0,max(nodes$y)+5),type = 'n', axes =
F,xlab = '', ylab = '', main = title)
  segments(edges$fromx, edges$fromy, x1= edges$tox, y1= edges$toy)
  points(nodes$x, nodes$y, pch=21, cex=3.5, bg=nodes$color)
  text(nodes$x[1:length(flatten(instru))-1],
nodes$y[1:length(flatten(instru))-1],
       nodes$label1[1:length(flatten(instru))-1], pos=1, offset=1)
  text(nodes$x[length(flatten(instru))]+2,
nodes$y[length(flatten(instru))],
       nodes$label1[length(flatten(instru))], pos=3, offset=1.5)
  legend_image <- as.raster(matrix(colorRampPalette(c('gray22',
'gray60'))(99), ncol=1))
  par(mar=c(5.1,1.0,4.1,2.1))
  plot(c(0,2),c(0,2),type = 'n', axes = F,xlab = '', ylab = '', main
= 'Legend')
  text(x=1.5, y = c(.5, seq(1,2,by=.25)), labels = c(0, 1,
seq(1.25,2,by=.25)))
  rasterImage(legend_image, 0, 1, 1,2)
  rect(0.025,.5,1,1, col='grey99', border='black')
}


##RUNNING THE FUNCTIONS
##running the instrumental inequalities across all combinations
exposurelist <- c("dichot_vitaminD", "trichot_vitaminD")
outcomelist <- c("pdp_symptoms", "adhd_symptoms_mom",
"adhd_symptoms_teacher")
expout <- list(exposurelist, outcomelist)
expout <- expand.grid(expout, stringsAsFactors = FALSE)
names(expout) <- c("exposure", "outcome")
comboresults <- sapply(1:nrow(expout), function(i){
  instrumental_inequalities_multiv(datasetname=total,
               IV=c("rs2282679_mother", "rs12785878_mother",
"rs6013897_mother", "rs10741657_mother"),
               exposure=expout$exposure[i],
outcome=expout$outcome[i])
})
```

```
##generating instrumental inequality plots and saving as png files
in working directory
sapply(1:nrow(expout), function(i){
eval(parse(text=sprintf("png('%s_%s_%s_instrumentalinequalities.png'
, width=800, height=700)
plot_instrumental_inequalities(instru = comboresults[[i*4-2]],
k=comboresults[[i*4-1]],
ineqs=comboresults[[i*4]], title='Instrumental Inequalities for MR
model of effect of %s on %s')
dev.off()", Sys.Date(), expout$exposure[i], expout$outcome[i],
expout$exposure[i], expout$outcome[i])))
})
```

7. R code for simulations

```
library(tidyverse)
library(gplots)

##data generating function
##parameters: n = number individuals in sample, nz = number of
proposed instruments,
##z_prob = vector of probabilities of zs (for generation), b_prev_a =
baseline
##prevalence of exposure (beta 0 for a), b_ua = beta for effect of u
on a,
##b_za = vector of betas for effect of zs on exposure, b_zy = vector
of betas for
##effects of z on y, b_uy = effect of u on y, b_prev_y = baseline prev
of outcome
##(beta 0 for y).


datagen <- function(n=1000, nz=4, z_prob=rep(0.5,4), b_prev_a=0.1,
b_ua=0.1,
                    b_za=rep(0.1,4), b_zy=NULL, b_uy=0.1,
b_prev_y=0.1, b_ay =0){
  ##library so that can use pipe
  library(magrittr)

  ##setting up null effect of zs on y if effects of z on y are not
specified
  if (is.null(b_zy)) b_zy <- rep(0, nz)

  ##creating inverse logit function
  inv.logit <- function(x) {
    return(exp(x)/(1+exp(x)))
  }

  # Create nz instruments
  for (i in 1:nz) {assign(paste0("z", i), rbinom(n, size = 1, prob =
z_prob[i]))}

  # U, A and Y
  ##u is standard normal
  u <- rnorm(n = n, mean = 0, sd = 1)

  ##creating formula for a - p(a) = b_prev_a + b_ua*u + z1 +z2+ ..+z-
nz
  a_formula <- paste0("b_prev_a + b_ua*u + ", paste0(b_za, "*",
paste0("z",1:nz),
                                                     collapse= " + "))

  ##evaluating function to run a - a is random binomial of size 1,
with
```

```
  ##probability inv.logit(formula above)
  a <- rbinom(n, size = 1, prob = inv.logit(eval(parse(text =
a_formula))))

  ##generating p(y) formula as
  ##baseline prevalence (beta 0) + u*beta_u + a*beta_ay+z*beta_z
(these zero out if not sp)
  y_formula <- paste0("b_prev_y + b_uy*u + b_ay*a + ",
                      paste0(b_zy, "*", paste0("z",1:nz), collapse= "
+ "))

  ##evaluating formula to get y vector
  y <- rbinom(n, size = 1, prob = inv.logit(eval(parse(text =
y_formula))))

  #binding zs together - use mget because it evaluates pasted names as
objects,
  ##not just names. do.call creates function call from function and
list containing
  ##arguments
  test <- do.call(cbind, mget(paste0("z", 1:nz)))

  ##cbinding the rest of this
  test <- cbind(test, u, a, y) %>% data.frame

  return(list(data=test,
              params=as.list(match.call())))
}


##Alteration of inequalities function for simulations - produces
shortened output
## (maximum value of inequalities for given combination only)
##
##ARGUMENTS:
## datasetname: the dataset to be used (data.frame)
## IV: a character vector containing the names of the variables
proposed as
##instruments (character vector)
## exposure: the name of the exposure of interest (character)
## outcome: the name of the outcome of interest (character)
##
##requires internal function
run_instrumental_inequalities_singlejointiv (see eAppendix 6)

instrumental_inequalities_multiv_short <- function(datasetname, IV,
exposure, outcome){

  k <- length(IV)

  # create list of sets of instruments
```

```
  mylist <- lapply(seq_along(IV), function(i) combn(IV, i, FUN =
list))

  mylist <- flatten(mylist)

  # check GRS and add to list
  datasetname$AlleleScore <- apply(datasetname[IV], 1, sum)
  mylist <- c(mylist, "AlleleScore")

  # create summary table
  summarydat <- matrix(nrow=length(mylist), ncol=1)
  #colnames(summarydat) <- c('BP_IVineq_violation', 'BP_IVineq_mean',
'BP_IVineq_sd')
  colnames(summarydat) <- c('BP_max')
  rownames(summarydat) <- c(mylist)


  # create variable and run results for each possible combination of
proposed IVs
  for (i in 1:length(mylist)){
    dat <- datasetname %>% select(IV, everything())
    dat$a <- dat[[exposure]]
    dat$y <- dat[[outcome]]
    dat <- dat%>% select(IV, AlleleScore, a, y)
    dat <- dat %>% drop_na()
    n_uniq_Y <- length(unique(dat$Y))

    #create new joint variable
    IVT = mylist[i]
    dat <- unite_(dat,"jointIV", flatten(IVT), remove = FALSE)

    #running instrumental inequalities function
    combo <- run_instrumental_inequalities_singlejointiv(data=dat,
                                                         y="y", x="a",

instrument="jointIV")
    ineq <- aggregate(sum_prop ~ IV1, data=combo, FUN=max)$sum_prop
%>% max


    summarydat[i,1] <- if (ineq<=1) {0} else {1}
  }


  return(list(summarydat))
}

##function to check risk difference in simulation
checkstrength <-function(dataset,var){sum(with(dataset, var==1 &
z1==1))/sum(with(dataset, z1==1))-
    sum(with(dataset, var==1 & z1==0))/sum(with(dataset,z1==0))}
```

```
##combines function into single function so more easily applied in
lapply
myfunc <- function(i){
  ds <- datagen(n=param_grid$n[i],
                nz=4,
                z_prob = rep(0.5,4),
                b_prev_a = .6,
                b_ua = 0.1,
                b_za = rep(param_grid$b_za[i],4),
                b_zy = c(param_grid$b_zy[i], rep(0,3)),
                b_uy = 0.1,
                b_prev_y = 0.02)

  res <- instrumental_inequalities_multiv_short(datasetname = ds$data,
                                                IV = paste0("z",1:4),
                                                exposure = "a",
                                                outcome = "y")

  meanstrength <- checkstrength(ds[[1]], var = ds[[1]]$a)

  meanviol <- checkstrength(ds[[1]], var = ds[[1]]$y)

  return(c(res, meanstrength, meanviol))}

##running simulations
##set random seed
set.seed(587643)

##generates grid of combinations of supplied vectors
##currently number of participants, IV strength, violation strength
param_grid <- expand.grid(n = c(1000, 10000, 100000), b_za = c(.01,
.1, .5, 1),
                          b_zy=c(0.001,0.1,0.5,0.8, 1, 1.5, 2, 4))

##number replications
n_reps = 1000


##actually applying functions
##generates list of lists
##each list contains 3 outputs: [[1]] = column of proportion IV
inequalities violated
##over n_reps for given n, b_za, and b_zy, [[2]] = mean IV strength
(RD scale),
## [[3]] = mean violation strength (RD scale)

sim_res <-lapply(seq_len(nrow(param_grid)), function(i){
  sim <- replicate(n_reps,myfunc(i))
  simsum <- rowSums(as.data.frame(sim[1,]))/n_reps
  meanstrength <- sum(unlist(sim[2,]))/n_reps
```

```
  meanviol <- sum(unlist(sim[3,]))/n_reps
  return(list(simsum, meanstrength, meanviol))})



##generating list of matrices of proportion finding violation of
inequalities
##proposed instrument vs. strength of violation
##each one is different sample size and instrument strength (based on
param_grid)
heatvis <- lapply(seq(1,12),function(i){
  assign(paste0("sim_res", i), sim_res[c(seq(i ,nrow(param_grid),12))]
%>%
    map(1) %>% invoke(cbind,.))})

## naming columns of heatmap with strength of violation
for (i in seq(1,12)){
  ##colnames using b_zy values
  #colnames(heatvis[[i]]) <- param_grid$b_zy[c(seq(i,nrow(param_grid),
12))]
  ##colnames using appx violation values
  colnames(heatvis[[i]]) <- sim_res[c(seq(i,nrow(param_grid), 12))]
%>%
    map(3) %>% as.numeric(.) %>% round(., 3)
  ##could also set one set of appx values and use those
}


##Generating actual heatmaps
sapply(1:length(heatvis), function(i){
  datatable<- as.data.frame(heatvis[[i]])

  ##sorting by violating or not violating
  datatable <- datatable %>% rownames_to_column()
  datatable$sort<-ifelse(str_detect(datatable$rowname,
                                    "z1|AlleleScore")==TRUE, 5.2,3.1)
  datatable$sort2 <- ifelse(datatable$sort==5.2,
                            "contains violating SNP",
                            "does not contain violating SNP")
  datatable <- datatable %>% arrange(desc(sort))
  numsims<- as.matrix(datatable[,2:8])
  row.names(numsims) <- datatable$rowname

  ##generate color palette
  my_palette<-c("white", colorRampPalette(c("gray96", "black"))(n=99))
  sidecol <-c()

  ##color breaks manually so transition is skewed
  col_breaks <- c(0, seq(.00001,1,by=.01))

  #actual heatmap generation
```

```
  ##remember to change x axis label depending on whether using rd
scale or b_zy
  eval(parse(text=sprintf("png('%s_%s_ineqssimheatmap.png',
width=13*300, height=5*300, res=300, pointsize=8)
                          par(mar=c(0,0,0,0))
                          heatmap.2(numsims,
                          cellnote=numsims,
                          main = ' ',
                          xlab = 'expit(b_zy)',
                          ylab = 'Proposed Instrument(s)',
                          notecol='deepskyblue3',
                          notecex=1.3,
                          density.info='none',
                          trace='none',
                          col=my_palette,
                          breaks = col_breaks,
                          dendrogram='none',
                          cexRow= 1.3,
                          cexCol = 1.3,
                          keysize=1,
                          lwid=c(.75,4),
                          lhei=c(.75,4),
                          margins=c(8,18),
                          Colv=FALSE,
                          Rowv=FALSE,

rowsep=length(datatable$sort[which(datatable$sort==5.2)]),
                          sepcolor='red',
                          add.expr = text(x=c(0,0),
                          y=c(7,16),
                          label= c('Does not violate MR conditions',
'Violates MR conditions'), xpd= NA, pos=2)

                          )

                          ", Sys.Date(), i )))
  dev.off()})



plot_instrumental_inequalities_sims <- function(instru,k,ineqs,s,
title){

  ##determining title
  if(missing(title)){title<-NA}

  ##generating nodes dataset
  nodes <- data.frame(id=1:length(flatten(instru)))
  nodes$label1 <- lapply(1:length(flatten(instru)),
                         function(i){flatten(instru)[[i]]})

  ##generate y position - aligning along y spots
```

```
  nodes$y <- 1
  for (j in 1:k){nodes<-nodes %>% mutate(y=ifelse(label1==instru[[j]],
j, y))}
  nodes$y <- as.numeric(nodes$y)
  nodes$y <- nodes$y*10
  ##label of group - creates their x axis coordinates
  nodes$x <- flatten(lapply(1:length(instru),
                            function(i){rep(length(instru[1:i]),

length(flatten(instru[i])))}))
  nodes$x <- unlist(nodes$x)
  nodes$x <- nodes$x*10

  ##edges generation
  edges <- data.frame(id=1:length(unique(nodes$y)))
  edges$fromy <- unique(nodes$y)
  edges$fromx <- tapply(nodes$x, nodes$y, min)
  edges$toy <- edges$fromy
  edges$tox <- tapply(nodes$x, nodes$y, max)
  vertedges <- data.frame(id=1:length(unique(nodes$x)))
  vertedges$fromx <- unique(nodes$x)
  vertedges$tox <- vertedges$fromx
  vertedges$fromy <- tapply(nodes$y, nodes$x, min)
  vertedges$toy <- tapply(nodes$y, nodes$x, max)
  edges <- rbind(edges, vertedges)

  ## size and color of nodes
  printsumnum <- function(i){print(ineqs[i]/s)}
  nodes <- nodes %>% rowwise %>% mutate(colorfactor=printsumnum(x/10))
  nodes$ii <- cut(as.numeric(nodes$colorfactor), breaks=seq(0, 1,
len=1000),
                  include.lowest=TRUE)
  nodes$color <-ifelse(nodes$colorfactor<.01,
colorRampPalette("white")(1),
                       colorRampPalette(c("grey75",
"black"))(999)[nodes$ii])

  ##node labels
  nodes$label1 <- ifelse(nodes$id<=k, nodes$label1,
                         ifelse(nodes$id==length(flatten(instru)),
                                nodes$label1, ""))

  ##generating plot itself
  layout(matrix(1:2,nrow=1),widths=c(0.8,0.2))
  par(mar=c(5.1,4.1,4.1,1.0))
  plot(c(-2,max(nodes$x)+5),c(0,max(nodes$y)+5),type = 'n',
       axes = F,xlab = '', ylab = '', main = title)
  segments(edges$fromx, edges$fromy, x1= edges$tox, y1= edges$toy)
  points(nodes$x, nodes$y, pch=21, cex=3.5, bg=nodes$color)
  text(nodes$x[1:length(flatten(instru))-1],
nodes$y[1:length(flatten(instru))-1],
       nodes$label1[1:length(flatten(instru))-1], pos=1, offset=1)
```

```
  text(nodes$x[length(flatten(instru))]+1.6,
nodes$y[length(flatten(instru))],
       nodes$label1[length(flatten(instru))], pos=3, offset=1.5, cex =
0.8)
  legend_image <- as.raster(matrix(colorRampPalette(c('black',
'white'))(999), ncol=1))
  par(mar=c(5.1,0,0,0))
  plot(c(0,2.2),c(0,2.2),type = 'n', axes = F,xlab = '', ylab = '',
main = '')
  text(x=1.5, y = c(1,2), labels = c(0, 1000))
  rasterImage(legend_image, 0, 1, 1,2)
  segments(c(0,0,0,1), c(2,2,1,2), x1=c(1,0,1,1), y1=c(2,1,1,1))
  text("Legend", x=0, y=2.1, pos=4)
}



##running instrumental inequality plots for 4 simulations of 100,000
simvis <- c(2, 5, 7, 8)
IV <- c("z1", "z2", " z3", "z4")
# create list of sets of instruments
mylist <- lapply(seq_along(IV), function(i) combn(IV, i, FUN = list))
mylist <- flatten(mylist)
mylist <- c(mylist, "AlleleScore")
instru <- mylist
sapply(1:length(simvis), function(i){

eval(parse(text=sprintf("png('%s_%s_instrumentalinequalitiessim.png',
width=800, height=700)
                        plot_instrumental_inequalities_sims(instru =
instru, k=4,
                        ineqs=unname(heatvis[[4]][,%s]), s=1,
title='Instrumental Inequalities for b_zy = %s simulation')
                        dev.off()", Sys.Date(), i, simvis[i],
simvis[i])))
})
```