1 Guide to running the code.

There are two major files for running the causal analysis. Firstly, run_MVMR.py, is the file for getting the causal genes for cases the user can supply the LD-matrix. This file can be downloaded and run on the terminal as

```
python3 run_MVMR.py "/home/user/file.csv" "/home/user/ld.csv"
```

After python3, the first argument should be the *file.csv* file containing the SNPs to exposure effects and SNPs to outcome effects. The first column is always the SNP ID's (the ID is irrelevant if you are providing the LD-matrix but should still be filled with default values), the last column is always the SNPs to outcome effect. Every other column in between is treated as an exposure variable. The separator to be used is comma. As an example:

```
SNPs,gene1,gene2,outcome
rs11191416,0.5,0.37,0.079
rs7098825,0.34,0.0,0.078
rs17115100,0.4,0.54,0.05
```

The second argument is ld.csv file for the LD matrix. Please make sure the ordering of the SNPs is same as in the exposure *file.csv* file. This has the format:

```
1.0 0.9 0.8
0.9 1.0 0.7
0.8 0.7 1.0
```

The delimiter is space. The results are saved as .csv file in the same directory as the one given for the exposure *file.csv* file. The code automatically prunes for SNPs in perfect-LD and keeps only the first occurring SNPs. If you would like to keep the most significant SNPs amongst perfect LD SNPs then please order the SNPs in decreasing order of significance in both .csv files.

Secondly, MVMR_withoutLD.py is the file which has the same purpose as run_MVMR.py except here you have to run the file on the terminal as

```
python3 MVMR_withoutLD.py "/home/user/file.csv"
```

i.e. without the file with the LD-matrix. The LD-matrix is generated using the TwoSampleMR function ld_matrix. Please make sure all SNPs belong to the LD panel.

1.1 Error messages

You can get the following errors while using the code:

Error Message : You require at least as many instruments as exposures to run this analysis. In this case you cannot run the analysis. The code prunes for SNPs in perfect LD so it may happen that you have more SNPs in the dataset you provided but they end up getting removed in the pruning and you get this error.

R[write to console]: The following variants are not present in the LD reference panel rs28789513. In this case you should remove the mentioned SNPs (rs28789513) from the dataset. You can only get this error if you do not provide an LD-matrix and it is generated using the R-package TwoSampleMR.

1.2 Guide to software requirements

To use this code, download the code files and ensure that you have the dependencies explained below. To run the code files, you have to type:

```
python3 run_MVMR.py "/home/user/file.csv" "/home/user/ld.csv"
```

Here, python3 should refer to at least Python 3.5 and depends on your specific installation of Python.

Before running the code files, check whether you have the following requirements and install them if necessary:

```
 Python 3.5 or later
```

You can check this by typing 'python' (or a more specific command as explained above) in the command line. For further support, in particular how to install python please visit `https://www.python.org/`.

This Python version has the packages numpy (version 1.11.0 or later), scipy, pandas, sys and rpy2 (version 2.9.4 or later).

You can check this by starting this python version (check it especially carefully if you have multiple Python versions on your system) and typing

```
import numpy
import pandas
import rpy2
import sys
```

You would also need statistical programming language R 3.2.0 or later. You can check this by typing 'R' in the command line. To install R, please visit `https://www.r-project.org/`. If you cannot provide the LD-matrix file, you need to have the package TwoSampleMR installed. You can install and call them in R using the following commands

```
install.packages("devtools")
remotes::install_github("MRCIEU/TwoSampleMR")
remotes::install_github("MRCIEU/MRInstruments")
```

and

```
library(devtools)
library(TwoSampleMR)
library(MRInstruments)
```

To get the LD-matrix, run the following commands in R

```
snps <- list('rs7776079','rs36049381','rs9349379')
matrix <- ld_matrix(snps, with_alleles = FALSE, pop = "EUR")
```

In *snps*, add the SNPs you wish to get the LD-matrix for.

The following code will save the LD-matrix in the format required for our analysis

```
write.table(matrix,file= "ld.csv",sep=" ",quote=F,col.names=F,row.names=F)
```

2 Data

We have used this method to estimate the causal effect of genes which are shared on a locus on outcome Coronary Artery Disease using summary statistics from genome wide association studies (GWAS). We used two different studies for outcome data, firstly, ebi-a-GCST003116 with trait as coronary artery disease, from the year 2015, Nikpay M is the author. The study is done on a population of European descent with logOR as the unit and 141217 sample size. Secondly, `finn-b-I9_CHD` with trait as Major coronary heart disease event, from the year 2021, Nikpay M is the author. The study is done on a population of European descent with unit and sample size not available.

For the summary data on eQTL analysis, we have used STARNET for association analysis from instruments (SNPs) to exposures (genes). But since this data is not publicly available, exposure data fom Gtex can be download from `https://gtexportal.org/home/datasets`. Here in the Single-Tissue cis-QTL data, you can download the full summary statistics of the cis-eQTLs mapped in European-American subjects. You can check the alignment of the effect allele from `https://www.gtexportal.org/home/faq#interpretEffectSize`.

To extract outcome data for the study of interest, we used the TwoSampleMR Package (package for performing Mendelian randomization using GWAS summary data). In this package you can select instruments for the exposure (in our case we had the STARNET exposure data) and then extract the instruments (SNPs) from the IEU GWAS database for the outcomes of interest. One benefit of using this to extract the instruments using this package is, that for a particular outcome of interest, it searches across studies to extract the SNPs which are in the exposure data. Also by default if a particular requested SNP is not present in the outcome GWAS then a SNP (proxy) that is in LD with the requested SNP (target) will be searched for instead. The *available_outcomes* function returns a table of all the available studies in the database. Each study has a unique ID.

```
library(TwoSampleMR)
available_outcomes <- available_outcomes()
View(available_outcomes)
```

If we want to extract the the SNPs which are in the exposure data for some outcome of interest we can just specify the outcome id and then just use the function to extract the outcome data. The below code specifies these two steps in R:

```
id_outcome <- ebi-a-GCST005195
outcome_dat <- extract_outcome_data(exposure_data$SNP, id_outcome)
```

This outcome data is not harmonized with the effect allele of the Starnet data (a1, alt) and for this we iterate over all variants in the Starnet data and align them according to effect allele of the given study while reversing the sign of the effect in Starnet.

For the validation of the causal genes we used exposure data from Genotype-Tissue Expression (GTEx) project which is a public resource to study tissue-specific gene expression and regulation. Data harmonization for this dataset with the outcome data was done in the similar fashion as before.

Note that if you are using Gtex for exposure datasets, you have to make sure that this minimum information is provided for the extraction of the outcome data and the minimum information required for MR analysis is the following:

```
SNP - rs ID.
beta - The effect size (binary traits log(OR)).
se - The standard error of the effect size.
effect_allele - allele of SNP which has the effect marked in beta.
```

When you download data from Gtex, you would have data in the format such as:

```
variant_id          gene_id            maf  slope slope_se  pval_beta
chr1_64764_C_T_b38  ENSG00000227232.5  0.06 0.5    0.1        1.3e-05
```

To get the outcome data for this exposure data, you would firstly need to replace the variant_id's to rs ID's from the gtex annotation file. The *slope* in the file is *beta.exposure* for the MR-Base package, *slope_se* is *se.exposure*, *pval_beta* is *pval.exposure*, *gene_id* is *exposure*, the *effect_allele.exposure* is the allele C in this example. You have to remember to match the rs ID's of the build b37 as MRBase package uses this build.

For the previous code, the corresponding GTex file (exposure data) for MR-analysis should look like

```
SNP         exposure           maf   beta.exposure se.exposure
rs769952832 ENSG00000227232.5  0.06    0.5            0.1
pval.exposure     effect_allele.exposure   other_allele.exposure
 1.3e-05                       C                           T
```

To save the effort of going from GTEx exposure data to data usable in the MRBase package, we have GTEx exposure data aligned with corresponding rs ID's and structured in the MRBase format, available at https://drive.google.com/drive/folders/14u2dN8k30wnZZkSkAQFN0ndboTFJFH-J?usp=share_link.

After this you can extract the outcome data for both studies as follows

```
id_outcome <- ebi-a-GCST005195
outcome_dat <- extract_outcome_data(exposure_data$SNP, id_outcome)
write.csv(cad_out_dat, "Outcome.csv" ,row.names = FALSE)
```

Now that you have the outcome and exposure data, you can choose a chromosome and genetic variants within 1Mb distance of each other to run the causal analysis.

We have a code *Seperate_chr.py* which can output the exposure and outcome data segregated per chromosome and position. You can give as input the exposure data file (GTEx exposure data) and outcome data file (outcome file after you extract outcome data from the MR-Base package) and the lastly an integer value and this function will take the exposure and output files, segregate both files per chromosome and additionally per position. As an example if you give the position argument to be 2, the code will segregate each file into batches of each chromosome and *additionally* each chromosome batch into batches of SNPs sharing only the first two digits in their base pair position. The output exposure and outcome files will be saved in the same input directory, categorized by their chromosome and first two digits of their base pair position. The input and outcome would both have *comma* as a separator for the *.csv* files.

Lastly you can give the outputs of the last function and use the file *Data_preparation.py* to get the datasets in the format needed for running the causal analysis. The input would be exposure and outcome files with *comma* as a sperator and the output would be a file with the format required for the causal analysis saved in the same input directory with prefix *_prepared.csv.*

3 Comparison to other methods

You can compare your results to other methods in the MVMR community like TWMR https://www.nature.com/articles/s41467-019-10936-0, MVMR https://onlinelibrary.wiley.com/doi/10.1002/gepi.21758, and functions in the MR-Base package https://mrcieu.github.io/TwoSampleMR/.

For our simulations, we used TWMR as from their github page (code found under https://github.com/eleporcu/TWMR), MRBase with the exposure and outcome data and analysing after harmonizing the datasets as metioned in https://mrcieu.github.io/TwoSampleMR/articles/perform_mr.html#multivariable-mr. As for MVMR, we supplied data as mentioned in their vignette and then ran the analysis https://cran.r-project.org/web/packages/MendelianRandomization/vignettes/Vignette_MR.pdf.