mGWASR Tutorial

Contents

Overview

Introduction

mGWASR contains the R functions and libraries underlying the mGWAS-Explorer web server. After installing and loading the package, users will be able to reproduce the same results from their local computers using the corresponding R command history downloaded from mGWAS-Explorer, thereby achieving maximum flexibility and reproducibility.

Following installation and loading of *mGWASR*, users will be able to reproduce web server results from their local computers using the corresponding R command history downloaded from mGWAS-Explorer, thereby achieving maximum flexibility and reproducibility.

Installation

Step 1. Install package dependencies

To use mGWASR , first install all package dependencies. Ensure that you are able to download packages from bioconductor. To install package dependencies, use the pacman R package (for those with >R 3.5.1). Note that some of these packages may require additional library dependencies that need to be installed prior to their own successful installation.

```
if (!requireNamespace("pacman", quietly = TRUE)) {
  install.packages("pacman")
}
if (!requireNamespace("devtools", quietly = TRUE)) {
  install.packages("devtools")
}

library(pacman)
library(devtools)

# need to update the depdencies
pacman::p_load(RSQLite, igraph, BiocManager, BiocParallel,  pryr,  httr,  reshape,  ggplot2,  RJSONIO,
               mygene, myvariant, RMySQL, MendelianRandomization, VariantAnnotation)
devtools::install_github(c(
  "mglev1n/ldscr",
  "boxiangliu/locuscomparer",
  "explodecomputer/plinkbinr",
  "MRCIEU/genetics.binaRies",
  "mrcieu/gwasvcf",
  "mrcieu/gwasglue"
))
```

Step 2. Install the package

mGWASR is freely available from GitHub. The package documentation, including the vignettes for each module and user manual is available within the downloaded R package file. If all package dependencies were installed, you will be able to install the mGWASR. Due to issues with Latex, some users may find that they are only able to install mGWASR without any documentation (i.e. vignettes).

Install the package directly from github using the *devtools* package. Open R and enter:

```
# Install mGWASR WITHOUT documentation
devtools::install_github("xia-lab/mGWASR", build = TRUE, build_opts = c("--no-resave-data", "--no-manual

# Install mGWASR WITH documentation
devtools::install_github("xia-lab/mGWASR", build = TRUE, build_opts = c("--no-resave-data", "--no-manual
```

Tips for using the mGWASR package

1) The first function that you will use in every module is the `InitDataObjects` function, which constructs the *dataSet* object that stores user's data for further processing and analysis.
2) The mGWASR package directly creates data files/tables/analysis/networks outputs in your current working directory.
3) Every command must be run in sequence, please do not skip any commands as this will result in errors downstream.
4) Each main function in mGWASR is documented. Use the *?Function* format to open its documentation. For instance, use `?mGWASR::QueryExposure` to find out more about this function.

Download SQLite database and PLINK binary file

The mGWASR package requires access to a SQLite database containing curated mGWAS information and relies on PLINK for genetic analysis.

1. **SQLite Database**: Contains 313,720 significant SNP-metabolite associations from 65 metabolite GWAS studies
2. **PLINK**: Required for genetic analysis operations

Option 1: Automatic Setup (Recommended)

The easiest way to use mGWASR is to set environment variables that allow automatic downloading and configuration of dependencies.

```
# Load the library
library(mGWASR)
#### Set up file path
Sys.setenv(SQLITE_PATH = "/path/to/sqlite") #Example: "~/mGWAS_depend/"
Sys.setenv(PLINK_PATH = "/path/to/plink")

# Initiate the dataSet object for storing processed data

mSet <- InitDataObjects("met2snp")
```

When you run `InitDataObjects`, mGWASR will: 1. Check if the SQLite database exists in the specified directory 2. If not found, automatically download it (~200MB) 3. Check if PLINK exists in the specified directory 4. If not found, automatically download and make it executable

Option 2: Manual Setup

If you prefer to manage dependencies manually:

1. Download the SQLite database:

   ```r
   # Create a directory to store the database
   dir.create("~/mgwas_data", showWarnings = FALSE, recursive = TRUE)

   # Download the database file
   download.file(
     "https://www.xialab.ca/rest/sqlite/mgwas_202201.sqlite",
     "~/mgwas_data/mgwas_202201.sqlite",
     mode = "wb"
   )
   ```

2. Download PLINK:

   ```r
   # Download PLINK
   download.file(
     "https://github.com/MRCIEU/genetics.binaRies/raw/refs/heads/master/binaries/Linux/plink",
     "~/mgwas_data/plink",
     mode = "wb"
   )

   # Make PLINK executable
   Sys.chmod("~/mgwas_data/plink", "0755")
   ```

3. Set environment variables and load the library:

   ```r
   Sys.setenv(SQLITE_PATH = "~/mgwas_data")
   Sys.setenv(PLINK_PATH = "~/mgwas_data")
   ```

Obtain statistical associations and biological mappings

Starting from a list of metabolites

```r
SetAnalType("met2snp")
SetVepOpt("myvariant")
SetLDProxy("None")
SetLDR2("0.8")
mSet <- PerformCmpdMapping(mSet, "D-Glucose
Pyruvic acid
Betaine
L-Serine
Trimethylamine
Acetone
Acetic acid
2-Hydroxybutyric acid", "name", "blood", "all");
mSet <- CrossReferencing(mSet, "name");
```

```
mSet <- CreateMappingResultTable(mSetObj)
nms.vec <- "met2snp"
SetMappingType();
QueryMultiList();
# working till here

mSet<-CreateGraph(mSet, "metabolite")
```

Note, for users to prepare a list of metabolites, they can copy and paste a list into a plain text file, which can then be uploaded to R. These steps include: 1) Set working directory to location of your files for analysis; 2) Set your list of features as “cmpdListFile” in Rpkgdown::build_site()

3) Read in the text file for analysis using readChar, saving it as “cmpdListFile”

Starting from a list of item

SNP-phenotype A-phenotype B causal hypothesis

Using a combination of fine-mapping and colocalization analysis, it is possible to quickly determine the presence of genetic sharing (SNPs) between multiple traits such as metabolite levels and disease. This leads to the hypothesis that a certain SNP leads to phenotype B by altering phenotype A, which can help to reveal the common genetic basis of complex traits.

get GWAS ID from ieu OpenGWAS database

Step 1: Obtain OpenGWAS API Token Before you can access the OpenGWAS database, you need to get an API token. This token is required for authentication and allows you to query the database. You can obtain the token from https://api.opengwas.io/.

```
Sys.setenv(OPENGWAS_JWT="your API token") # Replace "your API token" with the actual token you obtained
```

Step 2: Retrieve GWAS Information Once you have the API token, you can use the `ieugwasr` package to fetch GWAS information. This package provides functions to interact with the OpenGWAS database.

```
allGWAS <- ieugwasr::gwasinfo() # This function retrieves a list of all available GWAS studies from the
```

Alternatively, you can load a pre-downloaded table of GWAS IDs from a file included in the R package:

```
allGWAS_file <- system.file("extdata", "2024_allGWAS.csv", package = "mGWASR") # Locate the file within
allGWAS <- read.csv(allGWAS_file,row.names = 1) # Read the CSV file into a data frame
```

Explanation: The allGWAS data frame contains information about all GWAS studies available in the OpenGWAS database, including their IDs, traits, and other metadata.

Find causal SNPs through finemap

The `findcausalSNP` function uses the "Sum of Single Effects" (SuSiE) model to identify causal SNPs based on two criteria: 1. PIP (Posterior Inclusion Probability): The probability that each SNP is included in the model. SNPs with PIP > 0.8 (default) are considered candidate causal SNPs. 2. Credible Sets (CSs): A set of SNPs whose cumulative PIP reaches a certain confidence level (default 95%).

Step 1: Select the Region of Interest

To identify causal SNPs, you need to specify a genomic region of interest. This region should be based on prior knowledge or hypotheses about where causal variants might be located.

```
region <- '11:58780549-62223771' # Example region on chromosome 11
```

Step 2: Search for Causal SNPs

Use the `findcausalSNP` function to search for causal SNPs within the specified region for a specific GWAS study. You need to provide the GWAS ID, the region, and the path to the binary file (`bfile`) containing the genotype data.

Explanation: This function returns a list of SNPs that both meet the threshold above the PIP threshold and are members of CSs.

Find co-localization relationships in large batches

Step 1: Extract GWAS IDs of Interest You may want to investigate co-localization relationships between a specific phenotype and multiple other phenotypes. First, extract the GWAS IDs of the phenotypes you are interested in.

```
pheno2 <- allGWAS[stringr::str_detect(allGWAS$id, "met-a|met-c"),'id'] # Extract IDs of metabolite-rela
```

Step 2: Search for Co-localization Relationships Use the `batch_coloc` function to search for co-localization relationships between the primary phenotype and multiple secondary phenotypes. Specify the SNPs or chr:pos and the region around them.

```
batch_coloc('ebi-a-GCST007515', pheno2, rsid_or_pos = c('rs174537','rs174547','rs1535'),region = 500000
```

Explanation: Co-localization analysis helps determine whether the same genetic variant influences multiple traits. The `region` parameter defines the genomic window around the SNPs to search for co-localization. A typical region is 500kb to 2Mb.

Alternatively, you can perform single co-localization and plot the results for visualization. This is useful for detailed examination of specific co-localization relationships.

```
single_res <- colocal('ebi-a-GCST007515',pheno2[1],rsid_or_pos = 'rs174537',region = 500000,plot = T)
#Alternatively, you can use a VCF file
#single_res <- colocal('pheno1_vcf_file','pheno2_vcf_file',rsid_or_pos = casusalsnp[1],region = 500000,
```

Perform two-sample Mendelian randomization analysis

Mendelian randomization (MR) is a method that uses genetic variants as instrumental variables to establish a causal relationship between an exposure (e.g., metabolite) and an outcome (e.g., disease). Two-sample MR (2SMR) uses summary statistics from genome-wide association studies (GWAS) for both the exposure and the outcome. In this tutorial, we will guide you through performing two-sample MR in R using the *mGWASR* package.

Load and prepare exposure and outcome data

In two-sample MR, you need summary statistics from separate GWAS for exposure and outcome. The dataset for the exposure should contain SNPs that are significantly associated with the exposure. The dataset for the outcome should contain the effects of these SNPs on the outcome. The datasets must have columns for SNP, effect size, and standard error.

Load the exposure and outcome data:

```
mSet<-InitDataObjects("mrmodule")
QueryExposure("Acetic acid")
QueryOutcome("ukb-b-4575")
```

Perform two-sample MR

```
PerformMRAnalysis("no_ldclump", "true", "0.8", "true", "0.3", "2")
```

Here's a brief explanation of the commonly used methods you can choose:

1) **Inverse Variance Weighted (IVW) Method** (`mr_ivw`): The basic idea here is to use each genetic variant as a separate instrumental variable and to combine the individual causal effect estimates using a meta-analysis approach. The main assumption is that the genetic variants are valid instrumental variables, i.e., they are associated with the exposure, not associated with confounders of the exposure-outcome relationship, and affect the outcome only through the exposure.

- **Pros**: It's the most common method, easy to understand, and provides precise estimates under ideal conditions.
- **Cons**: It is sensitive to invalid instrumental variables (pleiotropy). In such cases, the results may be biased.

2) **Weighted Median Method** (`mr_weighted_median`): This method calculates the median of the ratio estimates, but each ratio estimate is weighted by the inverse of its variance. This method provides a consistent estimate of the causal effect even when up to 50% of the information comes from invalid instrumental variables (those that violate the MR assumptions). If the weighted median estimate is similar to the standard MR estimate, it suggests that the results are robust to violations of the MR assumptions by some of the genetic variants.

- **Pros**: It provides a valid causal effect estimate even when up to 50% of the weight in the analysis comes from invalid instrumental variables.
- **Cons**: It is less efficient (i.e., has larger standard errors) than the IVW method when all genetic variants are valid instrumental variables.

3) **MR-Egger Method** (`mr_egger_regression`): This method is similar to the IVW approach, but allows the intercept term in the regression model to be non-zero, which can be an indication of pleiotropy. This analysis allows for the potential existence of pleiotropy, where a single gene or genetic variant affects more than one trait. The intercept from the MR-Egger regression provides a measure of the average pleiotropic effect across all genetic variants. A non-zero intercept can indicate directional pleiotropy, which could bias the MR estimates. The slope in MR-Egger regression provides a causal estimate that is corrected for pleiotropy.

- **Pros**: It can provide an unbiased estimate even when all genetic variants are invalid, as long as the pleiotropic effects of the genetic variants are uncorrelated with their associations with the exposure.
- **Cons**: It is less efficient than the IVW and weighted median methods and may have low statistical power.

4) **Wald ratio** (`mr_wald_ratio`): The Wald Ratio is one of the most straightforward methods. In the Wald ratio method, each genetic variant is used as an instrumental variable, and the causal effect of the exposure on the outcome is estimated as the ratio of the genetic association with the outcome to the genetic association with the exposure. This is done for each genetic variant, and then the results can be combined across genetic variants, for example by taking the average.

- **Pros**: The Wald ratio is simple to calculate and easy to interpret. It does not require complex statistical modeling or software.
- **Cons**: Assumptions: The Wald ratio assumes that the genetic variant is associated with the exposure, does not affect the outcome through any pathway other than the exposure (no horizontal pleiotropy), and is not associated with any confounders of the exposure-outcome relationship. If these assumptions are violated, the Wald ratio may give biased estimates. Sensitivity to weak instruments: The Wald ratio can give imprecise estimates if the genetic variant is weakly associated with the exposure, which can be a particular problem when there is only one genetic variant available as an instrumental variable.

The choice of method will depend on the specific context of the analysis. If you strongly believe that the genetic variants are valid instrumental variables, the IVW or Wald ratio method would be good choices. If you are unsure about the validity of the instrumental variables, the weighted median or MR-Egger method might be more appropriate. However, it's often a good idea to use several methods and compare the results to assess the robustness of your findings. If the different methods give similar results, this can increase your confidence in the causal effect estimate.

To perform the two-sample MR analysis, choose an appropriate MR method (e.g., inverse variance weighted [IVW] method). Run the analysis using the selected method:

Perform directionality analysis

Perform sensitivity analysis

Sensitivity analyses in Two-Sample MR are essential to validate the main findings and to assess the robustness of the MR estimates against violations of the key assumptions. Here are the most common sensitivity analyses in Two-Sample MR and how to interpret their results:

- Heterogeneity test: Testing for heterogeneity in a 2SMR analysis involves assessing whether the causal effect estimates from different SNPs are consistent with each other.

  The `mr_heterogeneity()` function returns a list with the following components:

  - Q: The Cochran's Q statistic, which is a measure of the weighted sum of squared differences between individual SNP estimates and the overall MR estimate. A large Q statistic relative to its degrees of freedom suggests heterogeneity.
  - Q_df: The degrees of freedom, which is equal to the number of SNPs minus 1.
  - Q_pval: The p-value for the Q statistic. A low p-value (e.g., less than 0.05) suggests that there is significant heterogeneity.

- Pleiotropy test: This test is based on MR-Egger regression and specifically tests the null hypothesis that the intercept from the MR-Egger regression is equal to zero. If the null hypothesis is rejected, it suggests that there is horizontal pleiotropy, meaning that the genetic variants used as instrumental variables affect the outcome through pathways that are not mediated by the exposure.

  The `mr_pleiotropy_test()` function returns a list with the following components:

  - pval: The p-value for the intercept from the MR-Egger regression. A low p-value (e.g., less than 0.05) suggests that there is significant horizontal pleiotropy.
  - egger_intercept: The intercept from the MR-Egger regression, which provides a measure of the average pleiotropic effect across all genetic variants.

- intercept_stderr: The standard error of the intercept.

If the p-value (pval) is less than a certain significance level (e.g., 0.05), it suggests that there is significant pleiotropy. This means that at least some of the genetic variants used as instrumental variables are affecting the outcome through pathways that are not mediated by the exposure, which could bias the MR estimates.

Interpret results

Review the table results

The output will display the causal estimate, its standard error, and the p-value. A significant p-value ($<$ 0.05) suggests a causal relationship between exposure and outcome.

Scatter plot

The scatter plot in a MR analysis is typically used to visualize the relationship between genetic associations with the exposure (on the x-axis) and the outcome (on the y-axis). Each point on the plot represents a single genetic variant (also known as a single nucleotide polymorphism or SNP).

Here's how you interpret the scatter plot:

1) Slope and Linearity: The slope of the line fitted through the points is an estimate of the causal effect of the exposure on the outcome. If the points are scattered around a straight line with a positive slope, it suggests that an increase in exposure is associated with an increase in the outcome, indicating a positive causal effect. Conversely, if the points are scattered around a straight line with a negative slope, it suggests that an increase in exposure is associated with a decrease in the outcome, indicating a negative causal effect.

2) Scatter around the line: If the points are closely clustered around the line, it suggests that the genetic variants are good instruments (i.e., they satisfy the assumptions of MR). If the points are widely scattered, it suggests potential violation of the MR assumptions, such as presence of pleiotropy (where a single gene affects more than one trait), or measurement error.

3) Outliers: Outliers are points that lie far from the line of best fit. These might represent genetic variants that have pleiotropic effects, or that are affected by measurement error or linkage disequilibrium (where genetic variants are inherited together more often than would be expected by chance). Outliers can bias the MR estimate, so it's important to consider sensitivity analyses (like MR-Egger regression or leave-one-out analysis) that can help assess the impact of potential outliers on the MR results.

Forest plot

A forest plot in a 2SMR analysis provides a visual representation of the causal effect estimates for each SNP, as well as the overall causal effect estimate.

Here's how you interpret a forest plot in MR analysis:

1) Individual SNP estimates: Each horizontal line in the forest plot represents one SNP. The point on the line indicates the estimated causal effect of the exposure on the outcome for that SNP. The length of the line represents the confidence interval for that estimate.

2) Overall MR estimate: This is usually represented at the bottom of the plot. The center of the point represents the combined estimate of the causal effect across all SNPs (i.e., the overall MR estimate), and the width of the point represents the confidence interval for the overall estimate.

3) Direction of effect: If the point estimates are mostly on the right side of the line of no effect (usually represented by a vertical line at 0), it suggests that the exposure increases the risk of the outcome. If they're mostly on the left side, it suggests that the exposure decreases the risk of the outcome.

4) Heterogeneity: If the individual SNP estimates are widely scattered, it suggests that there is heterogeneity in the causal estimates.

5) Influence of individual SNPs: If removing one SNP from the analysis changes the overall MR estimate substantially, it suggests that the SNP may be exerting undue influence on the results.

Leave-one-out plot

The leave-one-out (LOO) analysis is a sensitivity analysis used in MR. It assesses the influence of each individual SNP on the overall causal effect estimate. In this analysis, the MR is rerun multiple times, each time leaving out one SNP, to see how much the overall estimate changes.

Here's how you interpret a leave-one-out plot in MR analysis:

1) Influence of individual SNPs: Each point on the LOO plot represents the overall MR estimate obtained when that SNP is excluded from the analysis. If removing a particular SNP changes the overall MR estimate substantially, it suggests that this SNP may be exerting undue influence on the results.

2) Consistency of the MR estimate: If all the points on the LOO plot are close to the overall MR estimate, it suggests that the MR estimate is consistent and not unduly influenced by any single SNP.

3) Identifying outlier SNPs: If there are one or more points that are far from the overall MR estimate, these are outlier SNPs that may be violating the assumptions of MR. These SNPs might be worth further investigation to understand why they are outliers.

4) Confidence intervals: The LOO plot usually also shows the confidence intervals for each leave-one-out MR estimate. If these intervals overlap substantially, it suggests that the MR results are robust to the exclusion of individual SNPs.

Funnel plot

A funnel plot in a 2SMR analysis is used to assess potential bias in the causal effect estimates, often due to pleiotropy. The plot is a scatter plot of the precision of each SNP's causal effect estimate (on the y-axis) against the estimate itself (on the x-axis).

Here's how you interpret a funnel plot in MR analysis:

1) Symmetry: In the absence of bias, the plot should be symmetrical around the vertical line that represents the overall causal effect estimate. This is because each SNP's estimate is expected to vary randomly around the true causal effect, with higher precision estimates (those closer to the top of the plot) varying less than lower precision estimates (those closer to the bottom). If the plot is asymmetrical, it suggests that there may be bias in the MR estimates.

2) Direction of bias: If the points on the plot are mostly to the right of the overall estimate, it suggests that there is positive bias (i.e., the estimates are skewed towards positive values). If the points are mostly to the left, it suggests that there is negative bias (i.e., the estimates are skewed towards negative values).

3) Pleiotropy: Asymmetry in the funnel plot can be a sign of pleiotropy, where a gene influences the outcome through more than one pathway. This violates one of the key assumptions of MR and can bias the results.

4) Heterogeneity and outliers: A widely scattered plot or the presence of outlier points far from the vertical line can indicate heterogeneity in the causal estimates or the presence of outlier SNPs, which could be due to pleiotropy, linkage disequilibrium, or measurement error. These SNPs might be worth further investigation.

5) Precision: Points towards the top of the plot are SNPs with higher precision estimates (smaller standard errors), while points towards the bottom are SNPs with lower precision estimates (larger standard errors).

Batch estimation of heritability and genetic correlation

Perform LDSC analysis in batches conveniently to estimate heritability and genetic correlation Step 1: Download GWAS Sumstats To perform LDSC (Linkage Disequilibrium Score) analysis, you need GWAS summary statistics in the standard VCF format. Use the “download_ieuvcf” function to download the required files, or you can prepare them yourself.

```r
phenos <- allGWAS %>%
  filter(grepl("Type 2 diabetes", trait, ignore.case = TRUE) & population == "European")
download_ieuvcf(phenos$id,'T2D_EUR_vcf/') # Download VCF files for these studies
```

Step 2: Filter Successful Downloads Check which files were successfully downloaded and filter out any failed downloads.

```r
phenos <- download_status %>%
  filter(status == "Success")
```

Step 3: Convert VCF Files to LDSC required Format Convert the downloaded VCF files to the intermediate RDS format required for LDSC analysis.

```r
ldsc_file <- vcf2ldsc('T2D_EUR_vcf/',saveRDS = T,output_dir = 'T2D_EUR_vcf_rds',num_cores = 5)
```

Step 4: Batch Calculate Heritability Use the `batch_ldsc_h2` function to estimate the heritability of each phenotype.

```r
ldsc_h2 <- batch_ldsc_h2('T2D_EUR_vcf_rds/',num_cores = 8)
```

Step 5: Calculate Genetic Correlation Pair all phenotypes and calculate the genetic correlation between each pair.

```r
ldsc_res <- pair_ldsc('T2D_EUR_vcf_rds/',output_dir = 'T2D_EUR_res/',para_plan  = 'multicore',num_cores
```

Step 6: Merge and Filter Results Combine the results and filter them based on specific conditions to create a table for visualization.

```r
com_res <- combine_filter('T2D_EUR_res/',traits = c('ebi-a-GCST005047.vcf',
  'ebi-a-GCST005413.vcf',
  'ieu-a-1090.vcf',
  'ukb-b-13806.vcf',
  'ebi-a-GCST007517.vcf'))
```

Step 7: Plot Correlation Heatmap Visualize the genetic correlation results using a heatmap.

```
plot <- cor_plot(com_res)
```

Explanation: The LDSC analysis function of this package provides an interface for calculating the heritability and genetic correlation of standardized vcf sumstats.

Reproduce the results of the T2D case study

This section is used to reproduce the results of the case study of T2D in the article, which may take a long time.

Step 1: Obtain GWAS ID

Load the GWAS information from the pre-downloaded file.

```
allGWAS_file <- system.file("extdata", "2024_allGWAS.csv", package = "mGWASR")
allGWAS <- read.csv(allGWAS_file,row.names = 1)
```

Step 2: Find Causal SNPs through Fine-Mapping Specify multiple genomic regions of interest. Here are the top 10 regions related to metabolites after data mining in the article.

```
region <- c("11:58780549-62223771", "11:116383348-117747110", "19:44744108-46102697",
  "11:59251804-62201641", "15:58441366-59694116", "1:54226262-56413117",
  "11:116383543-117901740", "16:55903774-57664330", "2:26894985-28598777",
  "8:19492840-20060856")
```

Searching for causal SNPs of a specific GWAS (Type 2 diabetes, ebi-a-GCST007515) in regions of interest.

Step 3: Find Co-localization Relationships in Large Batches Extract GWAS IDs starting with met-a or met-c (metabolite GWAS provided by the IEU) that are to be compared with T2D.

```
pheno2 <- allGWAS[stringr::str_detect(allGWAS$id, "met-a|met-c"),'id']
```

Step 5: Search for Co-localization Relationships Use the `batch_coloc` function to search for co-localization relationships between T2D and hundreds of metabolites.

```
batch_coloc('ebi-a-GCST007515', pheno2, rsid_or_pos = causalsnp,region = 500000,output_dir = 'output_di
```

Step 6: Perform Single Co-localization and Plot Individually examine and visualize results between Type 2 diabetes (ebi-a-GCST007515) and Total fatty acids (met-c-936).

```
single_res <- colocal('ebi-a-GCST007515','met-c-936',rsid_or_pos = 'rs769449',region = 500000,plot = T)
#Alternatively, you can use a VCF file
#single_res <- colocal('pheno1_vcf_file','pheno2_vcf_file',rsid_or_pos = casusalsnp[1],region = 500000,
```