Package: gwasglue (via r-universe)

September 24, 2024

Title GWAS summary data sources connected to analytical tools

Version 0.0.0.9000

Description Many tools exist that use GWAS summary data for colocalisation, fine mapping, Mendelian randomization, visualisation, etc. This package is a conduit that connects R packages that can retrieve GWAS summary data to various tools for analysing those data.

URL https://github.com/mrcieu/gwasglue

BugReports https://github.com/mrcieu/gwasglue/issues

Depends R (>= 3.6.0), gwasvcf, ieugwasr

Imports dplyr, testthat, mr.raps, MendelianRandomization, MRPRESSO, RadialMR, MRMix, TwoSampleMR, magrittr, susieR

Suggests knitr, rmarkdown, finemapr, covr

Remotes mrcieu/gwasvcf, rondolab/MR-PRESSO, mrcieu/ieugwasr, mrcieu/TwoSampleMR, mrcieu/MRInstruments, WSpiller/RadialMR, gqi/MRMix, stephenslab/susieR, variani/finemapr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Repository https://mrcieu.r-universe.dev

RemoteUrl https://github.com/MRCIEU/gwasglue

RemoteRef HEAD

RemoteSha c2d5660eed389e1a9b3e04406b88731d642243f1

Contents

`clump_gwasvcf` *Perform LD clumping*

Description

<full description>

Usage

```
clump_gwasvcf(
  vcf,
  clump_kb = 1000,
  clump_r2 = 0.001,
  clump_p = 5e-08,
  pop = NULL,
  bfile = NULL,
  plink_bin = NULL,
  access_token = NULL
)
```

Arguments

`vcf`	VCF file or VCF object
`clump_kb`	Clumping kb window. Default is very strict, 10000
`clump_r2`	Clumping r2 threshold. Default is very strict, 0.001
`clump_p`	Clumping sig level for index variants. Default = 1 (i.e. no threshold)
`pop`	Super-population to use as reference panel. Default = "EUR". Options are EUR, SAS, EAS, AFR, AMR. 'legacy' also available - which is a previously used verison of the EUR panel with a slightly different set of markers
`bfile`	If this is provided then will use the API. Default = NULL
`plink_bin`	If null and bfile is not null then will detect packaged plink binary for specific OS. Otherwise specify path to plink binary. Default = NULL
`access_token`	Google OAuth2 access token. Used to authenticate level of access to data

Value

data frame of clumped results

`cojo_cond` *Perform conditional analysis using GCTA COJO*

Description

For a list of fine-mapped rsids, will assign to regions and generate conditionally independent summary stats for each rsid

Usage

```
cojo_cond(
  vcffile,
  bfile,
  snplist,
  pop,
  gcta = genetics.binaRies::get_gcta_binary(),
  workdir = tempdir(),
  threads = 1
)
```

Arguments

`vcffile`	Path to vcffile
`bfile`	LD reference panel
`snplist`	List of rsids
`pop`	EUR, ASN or AFR
`gcta`	Path to gcta binary. For convenience can use default=genetics.binaRies::get_gcta_binary()
`workdir`	Location to store temporary files. Default=tempdir()
`threads`	Number of parallel threads. Default=1

Value

List of independent summary stats

`cojo_sumstat_file` *Write vcf file to cojo sumstat file*

Description

Write vcf file to cojo sumstat file

Usage

```
cojo_sumstat_file(vcffile, outfile)
```

Arguments

`vcffile`	Path to vcf file
`outfile`	Path to output file

Value

vcf object

`coloc_to_gassocplot` *Convert coloc dataset to gassocplot dataset*

Description

Convert coloc dataset to gassocplot dataset

Usage

```
coloc_to_gassocplot(coloclist, bfile = NULL, plink_bin = NULL)
```

Arguments

`coloclist`	Output from *_to_coloc
`bfile`	If number of SNPs > 500 then need to provide your own LD reference panel. Provide plink dataset here.
`plink_bin`	If number of SNPs > 500 then need to provide your own LD reference panel. Provide plink executable here

Value

List to feed into gassocplot

`gwasvcf_to_coloc` *Generate coloc dataset from vcf files*

Description

Generate coloc dataset from vcf files

Usage

```
gwasvcf_to_coloc(vcf1, vcf2, chrompos)
```

Arguments

`vcf1`	VCF object or path to vcf file
`vcf2`	VCF object or path to vcf file
`chrompos`	Character of chr:pos1-pos2

Value

List of datasets to feed into coloc

`gwasvcf_to_finemapr` *Generate data for fine mapping analysis*

Description

For a given region and VCF file, extracts the variants in the region along with LD matrix from a reference panel

Usage

```
gwasvcf_to_finemapr(
  region,
  vcf,
  bfile,
  plink_bin = genetics.binaRies::get_plink_binary(),
  threads = 1
)
```

Arguments

`region`	Region of the genome to extract eg 1:109317192-110317192". Can be array
`vcf`	Path to VCF file or VCF object
`bfile`	LD reference panel
`plink_bin`	Path to plink. Default = genetics.binaRies::get_plink_binary()
`threads`	Number of threads to run in parallel. Default=1

Value

List of datasets for finemapping

gwasvcf_to_TwoSampleMR *Create exposure or outcome data format for TwoSampleMR from vcf*

Description

Create exposure or outcome data format for TwoSampleMR from vcf

Usage

```
gwasvcf_to_TwoSampleMR(vcf, type = "exposure")
```

Arguments

vcf	VCF object
type	="exposure" or "outcome"

Value

data frame

harmonise *Generic harmonisation function*

Description

Assumes ref and alt alleles available for target and reference datasets, and uses chr:pos for alignment

Usage

```
harmonise(
  chr1,
  pos1,
  ref1,
  alt1,
  chr2,
  pos2,
  ref2,
  alt2,
  rsid2 = NULL,
  indel_recode = FALSE,
  strand_flip = FALSE
)
```

Arguments

`chr1`	Vector
`pos1`	Vector
`ref1`	Vector
`alt1`	Vector
`chr2`	Vector
`pos2`	Vector
`ref2`	Vector
`alt2`	Vector
`rsid2`	Optional vector
`indel_recode`	=FALSE. If TRUE then attempts to recode D/I
`strand_flip`	=FALSE. If TRUE then attempts to flip strand when alignment is not otherwise possible

Details

0: stick 1: swap 2: rename indel 3: rename indel and swap 4: flip 5: flip and swap 6: drop (no match) 7: drop (no reference)

Value

Dataframe of outcomes

`harmonise_against_ref` *Harmonise gwas alleles to be same as reference*

Description

Harmonise gwas alleles to be same as reference

Usage

```
harmonise_against_ref(gwas, reference)
```

Arguments

`gwas`	<what param does>
`reference`	<what param does>

Value

data frame with attributes

ieugwasr_to_coloc *Generate coloc dataset from the IEU GWAS database*

Description

Generate coloc dataset from the IEU GWAS database

Usage

```
ieugwasr_to_coloc(id1, id2, chrompos, type1 = NULL, type2 = NULL)
```

Arguments

`id1` ID for trait 1

`id2` ID for trait 2

`chrompos` Character of chr:pos1-pos2

`type1` Provide "cc" or "quant" to override automatic lookup of trait type for trait 1

`type2` Provide "cc" or "quant" to override automatic lookup of trait type for trait 2

Value

List of datasets to feed into coloc

ieugwasr_to_finemapr *Generate data for analysis in various finemapping methods*

Description

Uses the finemapr package https://github.com/variani/finemapr

Usage

```
ieugwasr_to_finemapr(region, id, bfile = NULL, plink_bin = NULL)
```

Arguments

`region` Region of the genome to extract eg 1:109317192-110317192"

`id` Array of GWAS studies to query. See `gwasinfo` for available studies

`bfile` If this is provided then will use the API. Default = NULL

`plink_bin` If null and bfile is not null then will detect packaged plink binary for specific OS. Otherwise specify path to plink binary. Default = NULL

Value

Each id will be a list of z score data, ld matrix, and sample size

`ieugwasr_to_gassocplot` *Generate regional plot for ieugwasr*

Description

Uses James Staley's gassocplot package https://github.com/jrs95/gassocplot

Usage

```
ieugwasr_to_gassocplot(chrpos, id, bfile = NULL, plink_bin = NULL)
```

Arguments

`chrpos`	A window range to plot e.g. 16:3349655-3849655
`id`	Vector of one or more IEU GWAS db study IDs
`bfile`	If number of SNPs > 500 then need to provide your own LD reference panel. Provide plink dataset here.
`plink_bin`	If number of SNPs > 500 then need to provide your own LD reference panel. Provide plink executable here

Value

assoc_plot or stack_assoc_plot if multiple markers given

`ieugwasr_to_TwoSampleMR` *Convert output from query to TwoSampleMR format*

Description

Convert output from query to TwoSampleMR format

Usage

```
ieugwasr_to_TwoSampleMR(x, type = "exposure")
```

Arguments

`x`	Output from ieugwasr query e.g. associations, tophits, phewas
`type`	"exposure" (default) or "outcome"

Value

data frame

is_forward_strand *Check a GWAS dataset against a reference known to be on the forward strand*

Description

Assuming reference data is all on forward strand, check if the GWAS is also. Use some threshold e.g. if more than 90 need to be flipped then it's likely that the dataset is on the forward strand

Usage

```
is_forward_strand(
  gwas_snp,
  gwas_a1,
  gwas_a2,
  ref_snp,
  ref_a1,
  ref_a2,
  threshold = 0.9
)
```

Arguments

`gwas_snp` Vector of SNP names for the dataset being checked

`gwas_a1` Vector of alleles

`gwas_a2` Vector of alleles

`ref_snp` Vector of SNP names for the reference dataset

`ref_a1` Vector of alleles

`ref_a2` Vector of alleles

`threshold` =0.9 If the proportion of allele strands match is above this threshold, then declare the dataset to be on the forward strand

Details

This function can be used to evaluate how strict harmonisation should be The trade off if you assume we are not on the forward strand then palindromic SNPs are dropped within a particular frequency range But you could instead have some small probability of error for whether palindromic SNPs are on the forward strand, and avoid dropping too many variants.

Value

1 = Forward strand; 2 = Not on forward strand

make_TwoSampleMR_dat *Create a harmonised dataset from lists of vcf files*

Description

This mimics the TwoSampleMR::make_dat function, which automatically looks up exposure and outcome datasets and harmonises them, except this function uses GWAS-VCF datasets instead. The supporting reference datasets can be accessed by UoB users on BC4 using set_bc4_files()

Usage

```
make_TwoSampleMR_dat(
  id1,
  id2,
  proxies = TRUE,
  nthreads = 1,
  vcfdir = options()$gwasglue.vcfdir,
  proxydb = options()$gwasglue.proxydb,
  rsidx = options()$gwasglue.rsidx,
  bfile = options()$gwasglue.bfile,
  action = 1,
  plink_bin = genetics.binaRies::get_plink_binary()
)
```

Arguments

`id1`	Exposure datasets. Either an array of vcf files, or array of IDs if vcfdir is set
`id2`	Outcome datasets. Either an array of vcf files, or array of IDs if vcfdir is set
`proxies`	Lookup proxies? default=TRUE but requires either bfile or proxydb to be set
`nthreads`	Parellelise default=1
`vcfdir`	Location of vcf files if id1 and id2 are just IDs. Defaults to options()$gwasglue.vcfdir
`proxydb`	Location of LD proxy database Default=options()$gwasglue.proxydb
`rsidx`	Location of rsidx index database Default=options()$gwasglue.rsidx
`bfile`	Location of LD reference panel Default=options()$gwasglue.bfile

Value

harmonised dataset

map_variants_to_regions *For a set of variants map to LD regions*

Description

LD regions defined here https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4731402/

Usage

```
map_variants_to_regions(chrpos, pop)
```

Arguments

chrpos	Array of chr:pos
pop	EUR, AFR or ASN

organise_ids *Figure out specific files and IDs depending on what files exist and whethet vcfdir is set*

Description

Figure out specific files and IDs depending on what files exist and whethet vcfdir is set

Usage

```
organise_ids(id, vcfdir)
```

Arguments

id	List of IDs within the vcfdir structure, or a list of GWAS VCF files, or a mixture
vcfdir	Location of GWAS VCF files, or NULL if id is a list of actual files

Value

File paths to all datasets

read_gwas *Read in GWAS dataset*

Description

Read in GWAS dataset

Usage

```
read_gwas(
  filename,
  skip,
  delimiter,
  gzipped,
  snp,
  nea,
  ea,
  ea_af,
  effect,
  se,
  pval,
  n,
  info,
  z
)
```

Arguments

`filename`	<what param does>
`skip`	<what param does>
`delimiter`	<what param does>
`gzipped`	<what param does>
`snp`	<what param does>
`nea`	<what param does>
`ea`	<what param does>
`ea_af`	<what param does>
`effect`	<what param does>
`se`	<what param does>
`pval`	<what param does>
`n`	<what param does>
`info`	<what param does>
`z`	<what param does>

Value

data frame with log attributes

read_reference *Read in reference dataset*

Description

Read in reference dataset

Usage

```
read_reference(
  reference_file,
  rsid = NULL,
  chrompos = NULL,
  remove_dup_rsids = TRUE
)
```

Arguments

`reference_file`	Reference vcf
`rsid`	List of variants to read
`chrompos`	List of chrompos to read
`remove_dup_rsids`	=TRUE Remove duplicates from output

Value

data frame

set_bc4_files *Determine locations of useful reference datasets on bluecrystal4*

Description

This is a convenience function for members at the University of Bristol to automatically set file locations for various reference datasets. It relates only to paths on bc4

Usage

```
set_bc4_files()
```

`susieR_pipeline` *Perform fine mapping pipeline using susieR*

Description

Clumps data, then maps those to LD representative regions. Within each detected LD representative region, fine mapping is performed

Usage

```
susieR_pipeline(
  vcffile,
  bfile,
  plink_bin,
  pop,
  threads = 1,
  clump_kb = 1000,
  clump_r2 = 0.001,
  clump_p = 5e-08,
  ...
)
```

Arguments

`vcffile`	Path to vcf file
`bfile`	Path to ld reference panel
`plink_bin`	Path to plink
`pop`	EUR, ASN or AFR
`clump_kb`	<what param does>
`clump_r2`	<what param does>
`clump_p`	<what param does>
`...`	Optional arguments to be passed to susie_rss

Value

List

write_out *Create format for HPC pipeline*

Description

Takes raw files and aligns them to reference. Important if files don't have chr:pos already

Usage

```
write_out(harmonised, path)
```

Arguments

harmonised	Output from /codeharmonise_against_ref
path	Path to write out json file and txt file

Index