

Package ‘otargen’

September 22, 2024

Type Package

Title Access Open Target Genetics

Version 1.1.5

Date 2024-09-21

Maintainer Amir Feizi <afeizi@gmail.com>

Description Interact seamlessly with Open Target Genetics' GraphQL endpoint to query and retrieve tidy data tables, facilitating the analysis of genetic data. For more information about the Open Target Genetics API (<<https://genetics.opentargets.org/api>>).

Depends R (>= 3.1.0)

Imports ghql, janitor, ggiraphExtra, cli, dplyr, ggplot2, ggrepel, jsonlite, rlang, magrittr, stringr, tibble, tidyr

License MIT + file LICENSE

Encoding UTF-8

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

URL <https://amirfeizi.github.io/otargen/>

BugReports <https://github.com/amirfeizi/otargen/issues>

NeedsCompilation no

Author Amir Feizi [aut, cre],
Kamalika Ray [aut]

Repository CRAN

Date/Publication 2024-09-21 22:40:02 UTC

Contents

chemblQuery	2
clinvarQuery	3
colocalisationsForGene	4

compGenomicsQuery	5
geneInfo	6
genesForVariant	7
geneticConstraintQuery	8
getLociGenes	8
gwasColocalisation	9
gwasColocalisationForRegion	11
gwasCredibleSet	12
gwasRegional	13
indexVariantsAndStudiesForTagVariant	14
knownDrugsQuery	15
manhattan	16
mousePhenotypesQuery	17
overlapInfoForStudy	18
pheWAS	19
plot_coloc	20
plot_l2g	21
plot_manhattan	22
plot_phewas	22
qtlColocalisationVariantQuery	23
qtlCredibleSet	24
run_custom_query	25
studiesAndLeadVariantsForGeneByL2G	26
studyInfo	28
studyLocus2GeneTable	29
studyVariants	30
tagVariantsAndStudiesForIndexVariant	31
topOverlappedStudies	33
variantInfo	34
Index	36

chemblQuery

Retrieve ChEMBL data for a specified gene and disease.

Description

This function queries the Open Targets Genetics GraphQL API to retrieve ChEMBL data for a specified gene and disease, including evidence from the ChEMBL datasource.

Usage

```
chemblQuery(ensemblId, efoId, size = 10, cursor = NULL)
```

Arguments

ensemblId	Character: ENSEMBL ID of the target gene (e.g., ENSG00000169174).
efoId	Character: EFO ID of the disease (e.g., EFO_0004911).
size	Integer: Number of records to retrieve (default: 10).
cursor	Character: Cursor for pagination (default: NULL).

Value

Returns a data frame containing ChEMBL data for the specified gene and disease.

Examples

```
## Not run:
result <- chemblQuery(ensemblId = "ENSG00000169174",
  efoId = "EFO_0004911", size = 10)
result <- chemblQuery(ensemblId = "ENSG00000169174",
  efoId = "EFO_0004911", size = 10, cursor = NULL)

## End(Not run)
```

clinvarQuery	<i>Retrieve ClinVar data for a specified gene and disease.</i>
--------------	--

Description

This function queries the Open Targets Genetics GraphQL API to retrieve ClinVar data for a specified gene and disease, including evidence from the NCBI datasource.

Usage

```
clinvarQuery(ensemblId, efoId, size = 10, cursor = NULL)
```

Arguments

ensemblId	Character: ENSEMBL ID of the target gene (e.g., ENSG00000130164).
efoId	Character: EFO ID of the disease (e.g., EFO_0004911).
size	Integer: Number of records to retrieve (default: 10).
cursor	Character: Cursor for pagination (default: NULL).

Value

Returns a data frame containing ClinVar data for the specified gene and disease.

Examples

```
## Not run:
result <- clinvarQuery(ensemblId = "ENSG00000130164",
  efoId = "EFO_0004911", size = 10)
result <- clinvarQuery(ensemblId = "ENSG00000130164",
  efoId = "EFO_0004911", size = 10, cursor = NULL)

## End(Not run)
```

colocalisationsForGene

Retrieves Colocalisation Statistics for a Given Gene

Description

Retrieves Colocalisation statistics for a gene using ENSEMBL gene IDs or gene symbol. Colocalisation analysis is performed between all studies in the Portal with at least one overlapping associated locus. This analysis tests whether two independent associations at the same locus are consistent with having a shared causal variant. The function supports multiple gene IDs as a list. The returned data frame (tibble format) includes studies that have evidence of colocalisation with molecular QTLs for the selected gene(s).

Usage

```
colocalisationsForGene(genes)
```

Arguments

genes Character: Gene ENSEMBL ID (e.g. ENSG00000169174) or gene symbol (e.g. PCSK9). Multiple gene IDs are supported as a character vector.

Value

A data frame (tibble format) including the colocalisation data for the query gene(s).

The output tibble contains the following columns:

- **Study**: Character vector. Study identifier.
- **Trait_reported**: Character vector. Reported trait associated with the colocalisation.
- **Lead_variant**: Character vector. Lead variant associated with the colocalisation.
- **Molecular_trait**: Character vector. Molecular trait associated with the colocalisation.
- **Gene_symbol**: Character vector. Gene symbol associated with the colocalisation.
- **Tissue**: Character vector. Tissue where the colocalisation occurs.
- **Source**: Character vector. Source of the colocalisation data.
- **H3**: Numeric vector. H3 value associated with the colocalisation.
- **log2_H4_H3**: Numeric vector. Log2 ratio of H4 to H3 values.

- Title: Character vector. Title of the study.
- Author: Character vector. Author(s) of the study.
- Has_sumstats: Logical vector. Indicates if the study has summary statistics.
- numAssocLoci: Numeric vector. Number of associated loci in the study.
- nInitial_cohort: Numeric vector. Number of samples in the initial cohort.
- study_nReplication: Numeric vector. Number of samples in the replication cohort.
- study_nCases: Numeric vector. Number of cases in the study.
- Publication_date: Character vector. Publication date of the study.
- Journal: Character vector. Journal where the study was published.
- Pubmed_id: Character vector. PubMed identifier of the study.

Examples

```
## Not run:
result1 <- colocalisationsForGene(gene = c("ENSG00000163946", "ENSG00000169174", "ENSG00000143001"))
result2 <- colocalisationsForGene(gene = "ENSG00000169174")
result3 <- colocalisationsForGene(gene = c("TP53", "TASOR"))
result4 <- colocalisationsForGene(gene = "TP53")
## End(Not run)
```

compGenomicsQuery *Retrieve Comparative Genomics data for a specified gene.*

Description

This function queries the Open Targets Genetics GraphQL API to retrieve comparative genomics data for a specified gene.

Usage

```
compGenomicsQuery(ensemblId)
```

Arguments

ensemblId Character: ENSEMBL ID of the target gene (e.g., ENSG00000169174).

Value

Returns a data frame containing comparative genomics data for the specified gene.

Examples

```
## Not run:
result <- compGenomicsQuery(ensemblId = "ENSG00000169174")

## End(Not run)
```

`geneInfo`*Retrieve gene summary information.*

Description

This function takes a gene ENSEMBL id (e.g. ENSG00000169174) or a gene name (e.g. PCSK9) and returns a table containing details about input genes, such as genomic location, gene structure and its BioType.

Usage

```
geneInfo(gene)
```

Arguments

<code>gene</code>	Character: an ENSEMBL gene identifier (e.g. ENSG00000169174) or gene name (e.g. PCSK9).
-------------------	---

Value

Returns a data frame (tibble format) with the following data structure:

1. id
2. symbol
3. bioType
4. description
5. chromosome
6. tss
7. start
8. end
9. fwdStrand
10. exons

Examples

```
## Not run:  
result <- geneInfo(gene="ENSG00000169174")  
result <- geneInfo(gene="PCSK9")  
  
## End(Not run)
```

genesForVariant	<i>Retrieves variant-surrounding genes with calculated statistics for causal gene prioritization.</i>
-----------------	---

Description

This function retrieves all calculated prioritizing scores for surrounding genes of a specific variant based on the Open Target Genetics locus-to-gene (L2G) ML scoring pipeline. It provides detailed insights into the relationship between genetic variants and genes, allowing users to explore the impact of variants on gene expression, colocalization scores, chromatin interactions, and predicted functional effects. The function returns the information in a structured format, making it easier to analyze and interpret the results.

Usage

```
genesForVariant(variant_id)
```

Arguments

variant_id	A character string specifying the ID of the variant for which to fetch gene information.
------------	--

Value

A list with the following components:

- `v2g`: A data frame with all variant-to-gene information with the following data structure: - `gene.symbol`: character - `variant`: character - `overallScore`: numeric - `gene.id`: character
- `tssd`: A data frame with all details on colocalization scores effect of variant in expression of the genes across tissues with the following data structure: - `gene.symbol`: character - `variant`: character - `typeId`: character - `sourceId`: character - `aggregatedScore`: numeric - `tissues_distance`: integer - `tissues_score`: numeric - `tissues_quantile`: numeric - `tissues_id`: character - `tissues_name`: character
- `qtls`: List of QTL associations between genes and variants across analyzed tissues with the following data structure: - `gene.symbol`: character - `variant`: character - `typeId`: character - `aggregatedScore`: numeric - `tissues_quantile`: numeric - `tissues_beta`: numeric - `tissues_pval`: numeric - `tissues_id`: character - `tissues_name`: character
- `chromatin`: A data frame including all information on chromatin interactions effect involving genes and variants with the following data structure: - `gene.symbol`: character - `variant`: character - `typeId`: character - `sourceId`: character - `aggregatedScore`: numeric - `tissues_quantile`: numeric - `tissues_score`: numeric - `tissues_id`: character - `tissues_name`: character
- `functionalpred`: A data frame including predicted functional effects of variants on genes across tissues with the following data structure: - `gene.symbol`: character - `variant`: character - `typeId`: character - `sourceId`: character - `aggregatedScore`: numeric - `tissues_maxEffectLabel`: character - `tissues_maxEffectScore`: numeric - `tissues_id`: character - `tissues_name`: character

Examples

```
## Not run:
result <- genesForVariant(variant_id = "1_154453788_C_T")
result <- genesForVariant(variant_id = "rs4129267")
print(result)

## End(Not run)
```

geneticConstraintQuery

Retrieve Genetic Constraint data for a specified gene.

Description

This function queries the Open Targets Genetics GraphQL API to retrieve genetic constraint data for a specified gene.

Usage

```
geneticConstraintQuery(ensemblId)
```

Arguments

ensemblId Character: ENSEMBL ID of the target gene (e.g., ENSG00000169174).

Value

Returns a data frame containing genetic constraint data for the specified gene.

Examples

```
## Not run:
result <- geneticConstraintQuery(ensemblId = "ENSG00000169174")

## End(Not run)
```

getLociGenes

Retrieve summary information of the genes in a locus

Description

This function retrieves information for a specified region on a chromosome, including overlapping genes and their associated details such as ID, symbol, biotype, description, transcription start site (TSS), start position, end position, strand direction, and exons structure.

Usage

```
getLociGenes(chromosome, start, end)
```

Arguments

chromosome	Character: Chromosome number as a string.
start	Integer: Start position of the specified region on the chromosome.
end	Integer: End position of the specified region on the chromosome.

Value

Returns a tibble data frame of all the overlapping genes in the specified region with the following data structure:

- *id*: *Character*. ID of the gene.
- *symbol*: *Character*. Symbol of the gene.
- *bioType*: *Character*. Biotype of the gene.
- *description*: *Character*. Description of the gene.
- *chromosome*: *Character*. Chromosome of the gene.
- *tss*: *Integer*. Transcription start site of the gene.
- *start*: *Integer*. Start position of the gene.
- *end*: *Integer*. End position of the gene.
- *fwdStrand*: *Logical*. Strand direction of the gene.
- *exons*: *List*. List of exons of the gene.

Examples

```
## Not run:  
result <- get_genes(chromosome = "2", start = 239634984, end = 241634984)  
  
## End(Not run)
```

`gwasColocalisation` *Retrieve calculated GWAS colocalisation data*

Description

This function retrieves colocalisation data for a specific variant from a study with other GWAS studies. It returns a data frame of the studies that colocalise with the input variant and study, including details on the study and reported trait, index variant, and calculated coloc method (see Ref. below) outputs.

Usage

```
gwasColocalisation(study_id, variant_id)
```

Arguments

study_id	Character: Open Target Genetics generated ID for the GWAS study.
variant_id	Character: Open Target Genetics generated ID for the variant (CHRPOSITION_REFALLELE_ALTALLELE or rsID).

Value

Returns a data frame of the studies that colocalise with the input variant and study. The table consists of the following data structure:

- study.studyId: *Character vector*. Study identifier.
- study.traitReported: *Character vector*. Reported trait associated with the colocalisation.
- study.traitCategory: *Character vector*. Trait category.
- indexVariant.id: *Character vector*. Index variant identifier.
- indexVariant.position: *Integer vector*. Index variant position.
- indexVariant.chromosome: *Character vector*. Index variant chromosome.
- indexVariant.rsId: *Character vector*. Index variant rsID.
- beta: *Numeric vector*. Beta value associated with the colocalisation.
- h3: *Numeric vector*. H3 value associated with the colocalisation.
- h4: *Numeric vector*. H4 value associated with the colocalisation.
- log2h4h3: *Numeric vector*. Log2 ratio of H4 to H3 values.

References

Giambartolomei, Claudia et al. "Bayesian test for colocalisation between pairs of genetic association studies using summary statistics." PLoS genetics vol. 10,5 e1004383. 15 May. 2014, doi:10.1371/journal.pgen.1004383

Examples

```
## Not run:
colocalisation_data <- gwasColocalisation(study_id = "GCST90002357", variant_id = "1_154119580_C_A")
colocalisation_data <- gwasColocalisation(study_id = "GCST90002357", variant_id = "rs2494663")

## End(Not run)
```

`gwasColocalisationForRegion`*Retrieve GWAS colocalisation data for a genomic region .*

Description

By providing a genomic region (chromosome name with start and end position), this function returns information about colocalisation between GWAS studies and associated loci within a specified genomic region. It provides details on the studies that have at least one overlapping associated locus within the region, allowing for the assessment of potential shared causal variants. The query output includes data such as the study identifiers, traits, loci information, and other relevant attributes.

Usage

```
gwasColocalisationForRegion(chromosome, start, end)
```

Arguments

<code>chromosome</code>	String: Chromosome number as a string.
<code>start</code>	Long: Start position of the specified chromosome.
<code>end</code>	Long: End position of the specified chromosome.

Value

Returns a data frame with the following columns:

- `leftVariant.id`: *Character vector*. ID of the left variant.
- `leftVariant.position`: *Integer vector*. Position of the left variant.
- `leftVariant.chromosome`: *Character vector*. Chromosome of the left variant.
- `leftVariant.rsId`: *Character vector*. rsID of the left variant.
- `leftStudy.studyId`: *Character vector*. Study identifier for the left study.
- `leftStudy.traitReported`: *Character vector*. Reported trait associated with the colocalisation in the left study.
- `leftStudy.traitCategory`: *Character vector*. Category of the reported trait in the left study.
- `rightVariant.id`: *Character vector*. ID of the right variant.
- `rightVariant.position`: *Integer vector*. Position of the right variant.
- `rightVariant.chromosome`: *Character vector*. Chromosome of the right variant.
- `rightVariant.rsId`: *Character vector*. rsID of the right variant.
- `rightStudy.studyId`: *Character vector*. Study identifier for the right study.
- `rightStudy.traitReported`: *Character vector*. Reported trait associated with the colocalisation in the right study.
- `rightStudy.traitCategory`: *Character vector*. Category of the reported trait in the right study.

- h3: *Numeric vector*. H3 value associated with the colocalisation.
- h4: *Numeric vector*. H4 value associated with the colocalisation.
- log2h4h3: *Numeric vector*. Log2 ratio of H4 to H3 values associated with the colocalisation.

References

- Giambartolomei, Claudia et al. "Bayesian test for colocalisation between pairs of genetic association studies using summary statistics." PLoS genetics vol. 10,5 e1004383. 15 May. 2014, doi:10.1371/journal.pgen.1004383

Examples

```
## Not run:
result <- gwasColocalisationForRegion(chromosome = "1", start = 153992685, end = 154155116)

## End(Not run)
```

gwasCredibleSet	<i>Retrieve GWAS credible set data</i>
-----------------	--

Description

Provided with a study ID and a lead variant ID, this function returns a data frame consisting of all the associated credible set tag variants with the corresponding statistical data.

Usage

```
gwasCredibleSet(study_id, variant_id)
```

Arguments

study_id	Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).
variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).

Value

Returns a data frame of results from the credible set of variants for a specific lead variant with the following columns:

- tagVariant.id: *Data frame*. A table of IDs of the tag variant.
- tagVariant.rsId: *Character vector*. rsID of the tag variant.
- beta: *Numeric*. Beta value.
- postProb: *Numeric*. Posterior probability.
- pval: *Numeric*. P-value.
- se: *Numeric*. Standard error.

- MultisignalMethod: *Character vector*. Multisignal method.
- logABF: *Numeric*. Logarithm of approximate Bayes factor.
- is95: *Logical*. Indicates if the variant has a 95
- is99: *Logical*. Indicates if the variant has a 99

Examples

```
## Not run:
result <- gwasCredibleSet(study_id="GCST90002357", variant_id="1_154119580_C_A")
result <- gwasCredibleSet(study_id="GCST90002357", variant_id="rs2494663")

## End(Not run)
```

gwasRegional	<i>Retrieve GWAS summary statistics for a genomic region.</i>
--------------	---

Description

For a given study ID and chromosomal region information, this function returns data frame(tibble format) with all variants and their GWAS summary statistics.

Usage

```
gwasRegional(study_id, chromosome, start, end)
```

Arguments

study_id	Character: Open Target Genetics generated ID for the GWAS study.
chromosome	Character: Chromosome number as a string.
start	Integer: Start position of the specified chromosome.
end	Integer: End position of the specified chromosome.

Value

Returns a data table of variant information and p-values with the following columns:

- variant.id: *Character vector*. Variant identifier.
- variant.chromosome: *Character vector*. Chromosome of the variant.
- variant.position: *Integer vector*. Position of the variant.
- pval: *Numeric vector*. P-value.

Examples

```
## Not run:
result <- gwasRegional(study_id = "GCST90002357",
  chromosome = "1", start = 153992685, end = 154155116)

## End(Not run)
```

indexVariantsAndStudiesForTagVariant

Retrieve population-level summary GWAS statistics.

Description

For an input tag variant ID, this function returns a data frame(tibble format) with population-level summary statistics data across various GWAS studies.

Usage

```
indexVariantsAndStudiesForTagVariant(variant_id, pageindex = 0, pagesize = 20)
```

Arguments

variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).
pageindex	Integer: Index of the current page, pagination index ≥ 0 .
pagesize	Integer: Number of records in a page, pagination size > 0 .

Value

Returns a data frame containing the variant associated with the input tag variant. The table consists of the following columns:

- `index_variant`: *Data frame*. Data frame of index variants with the following columns:
 - `id`: *Character vector*. Variant ID.
 - `rsId`: *Character vector*. rsID of the variant.
- `study`: *Data frame*. Data frame of studies with the following columns:
 - `studyId`: *Character vector*. Study identifier.
 - `traitReported`: *Character vector*. Reported trait associated with the colocalisation.
 - `traitCategory`: *Character vector*. Trait category.
- `pval`: *Numeric vector*. P-value.
- `pval_mantissa`: *Numeric vector*. Mantissa of the p-value.
- `pval_exponent`: *Integer vector*. Exponent of the p-value.
- `n_total`: *Integer vector*. Total number of samples.
- `n_cases`: *Integer vector*. Number of cases.
- `overall_r2`: *Numeric vector*. Overall R-squared value.
- `afr1000g_prop`: *Numeric vector*. Proportion in African population in 1000 Genomes.
- `amr1000g_prop`: *Numeric vector*. Proportion in Admixed American population in 1000 Genomes.
- `eas1000g_prop`: *Numeric vector*. Proportion in East Asian population in 1000 Genomes.

- `eur1000g_prop`: *Numeric vector*. Proportion in European population in 1000 Genomes.
- `sas1000g_prop`: *Numeric vector*. Proportion in South Asian population in 1000 Genomes.
- `log10abf`: *Numeric vector*. Log10 ABF (Approximate Bayes Factor).
- `posterior_probability`: *Numeric vector*. Posterior probability.
- `odds_ratio`: *Numeric vector*. Odds ratio.
- `odds_ratio_ci_lower`: *Numeric vector*. Lower confidence interval of the odds ratio.
- `odds_ratio_ci_upper`: *Numeric vector*. Upper confidence interval of the odds ratio.
- `beta`: *Numeric vector*. Beta value.
- `beta_ci_lower`: *Numeric vector*. Lower confidence interval of the beta value.
- `beta_ci_upper`: *Numeric vector*. Upper confidence interval of the beta value.
- `direction`: *Character vector*. Direction of the effect.

Examples

```
## Not run:
result <- indexVariantsAndStudiesForTagVariant(variant_id = "1_109274968_G_T")
result <- indexVariantsAndStudiesForTagVariant(variant_id = "rs12740374",
  pageindex = 1, pagesize = 50)

## End(Not run)
```

knownDrugsQuery	<i>Retrieve Known Drugs data for a specified gene.</i>
-----------------	--

Description

This function queries the Open Targets Genetics GraphQL API to retrieve known drugs data for a specified gene.

Usage

```
knownDrugsQuery(ensgId, cursor = NULL, freeTextQuery = NULL, size = 10)
```

Arguments

<code>ensgId</code>	Character: ENSEMBL ID of the target gene (e.g., ENSG00000169174).
<code>cursor</code>	Character: Cursor for pagination (default: NULL).
<code>freeTextQuery</code>	Character: Free text query to filter results (default: NULL).
<code>size</code>	Integer: Number of records to retrieve (default: 10).

Value

Returns a data frame containing known drugs data for the specified gene.

Examples

```
## Not run:
result <- knownDrugsQuery(ensgId = "ENSG00000169174", size = 10)
result <- knownDrugsQuery(ensgId = "ENSG00000169174",
  cursor = NULL, freeTextQuery = NULL, size = 10)

## End(Not run)
```

manhattan

*Retrieve GWAS summary statistics to create a Manhattan plot***Description**

This function retrieves summary statistics for a given GWAS study ID, which are used to generate a Manhattan plot. The Manhattan plot is a graphical representation of genetic association studies, particularly in genome-wide association studies (GWAS). It displays the results of statistical associations between genetic variants and a trait or disease of interest across the genome. This function returns a data frame of the underlying data, which can be used to recreate the Manhattan plot using the `plot_manhattan()` function, or for custom plots and downstream analysis.

Usage

```
manhattan(study_id, pageindex = 0, pagesize = 100)
```

Arguments

<code>study_id</code>	Character: Open Targets Genetics generated ID for the GWAS study.
<code>pageindex</code>	Int: Index of the current page (pagination index ≥ 0).
<code>pagesize</code>	Int: Number of records in a page (pagination size > 0).

Value

Returns a data frame containing the Manhattan associations for the input study ID. The table consists of the following columns:

- `pval_mantissa`: Numeric vector. Mantissa of the p-value.
- `pval_exponent`: Integer vector. Exponent of the p-value.
- `credible_set_size`: Integer vector. Size of the credible set.
- `ld_set_size`: Integer vector. Size of the LD set.
- `total_set_size`: Integer vector. Total size of the set.
- `pval`: Numeric vector. P-value.
- `odds_ratio`: Logical vector. Odds ratio.
- `odds_ratio_ci_lower`: Logical vector. Lower confidence interval of the odds ratio.
- `odds_ratio_ci_upper`: Logical vector. Upper confidence interval of the odds ratio.

- beta: Numeric vector. Beta value.
- beta_ci_lower: Numeric vector. Lower confidence interval of the beta value.
- beta_ci_upper: Numeric vector. Upper confidence interval of the beta value.
- direction: Character vector. Direction of the effect.
- best_genes_score: Numeric vector. Score of the best genes.
- best_genes_gene_id: Character vector. Gene ID of the best genes.
- best_genes_gene_symbol: Character vector. Gene symbol of the best genes.
- best_coloc_genes_score: Numeric vector. Score of the best colocated genes.
- best_coloc_genes_gene_id: Character vector. Gene ID of the best colocated genes.
- best_coloc_genes_gene_symbol: Character vector. Gene symbol of the best colocated genes.
- best_locus2genes_score: Numeric vector. Score of the best locus-to-genes.
- best_locus2genes_gene_id: Character vector. Gene ID of the best locus-to-genes.
- best_locus2genes_gene_symbol: Character vector. Gene symbol of the best locus-to-genes.
- variant_id: Character vector. Variant ID.
- variant_position: Integer vector. Variant position.
- variant_chromosome: Character vector. Variant chromosome.
- variant_rs_id: Character vector. Variant rsID.

Examples

```
## Not run:
result <- manhattan(study_id = "GCST90002357")
result <- manhattan(study_id = "GCST90002357", pageindex = 2, pagesize = 50)

## End(Not run)
```

mousePhenotypesQuery *Retrieve Mouse Phenotypes data for a specified gene.*

Description

This function queries the Open Targets Genetics GraphQL API to retrieve mouse phenotypes data for a specified gene.

Usage

```
mousePhenotypesQuery(ensemblId)
```

Arguments

ensemblId Character: ENSEMBL ID of the target gene (e.g., ENSG00000169174).

Value

Returns a data frame containing mouse phenotypes data for the specified gene.

Examples

```
## Not run:
result <- mousePhenotypesQuery(ensemblId = "ENSG00000169174")

## End(Not run)
```

overlapInfoForStudy *Retrieves overlap info for a study and a list of studies*

Description

For an input study ID and a list of other study IDs, this function returns two elements. One contains the overlap information in a table format, and the other element is the variant intersection set, representing an overlap between two variants of the two given studies.

Usage

```
overlapInfoForStudy(study_id, study_ids = list())
```

Arguments

study_id	Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).
study_ids	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).

Value

A list containing a data frame of overlap information and the variant intersection set. The overlap information table (overlap_info) consists of the following columns:

- studyId: *Character vector*. Study ID.
- traitReported: *Character vector*. Reported trait.
- traitCategory: *Character vector*. Trait category.
- variantIdA: *Character vector*. Variant ID from study A.
- variantIdB: *Character vector*. Variant ID from study B.
- overlapAB: *Integer vector*. Number of overlaps between variants A and B.
- distinctA: *Integer vector*. Number of distinct variants in study A.
- distinctB: *Integer vector*. Number of distinct variants in study B.
- study.studyId: *Character vector*. Study ID from study list.
- study.traitReported: *Character vector*. Reported trait from study list.
- study.traitCategory: *Character vector*. Trait category from study list.

The variant intersection set (variant_intersection_set) is a character vector representing the intersection of variants.

Examples

```
## Not run:
result <- overlapInfoForStudy(study_id = "GCST90002357",
  study_ids = list("GCST90025975", "GCST90025962"))

## End(Not run)
```

pheWAS	<i>Retrieve PheWAS (Phenome Wide Association Studies) data for a variant.</i>
--------	---

Description

PheWAS (*Phenome-wide association study*) is a method that investigates the relationships between genetic variants and traits or phenotypes, helping in the study of their potential influence on multiple traits or diseases concurrently. This function retrieves the traits associated with a given variant in the *UK Biobank*, *FinnGen*, and/or *GWAS Catalog* summary statistics repository (only traits with a p-value less than 0.005 are returned).

Usage

```
pheWAS(variant_id)
```

Arguments

variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).
------------	---

Value

A data frame with PheWAS associations.

The output data frame contains the following columns:

- `totalGWASstudies`: An integer indicating the total number of GWAS studies where the variant is associated.
- `pval`: A numeric value representing the p-value of the association between the variant and the trait.
- `beta`: A numeric value representing the beta value, which represents the effect size of the variant on the trait.
- `oddsRatio`: A numeric value representing the odds ratio, measuring the association between the variant and the trait.
- `nTotal`: An integer indicating the total number of participants in the study.
- `study.studyId`: A character vector representing the study ID.
- `study.source`: A character vector representing the source of the study.
- `study.pmid`: A character vector representing the PubMed ID (PMID) of the study.

- `study.pubDate`: A character vector representing the publication date of the study.
- `study.traitReported`: A character vector representing the reported trait associated with the variant.
- `study.traitCategory`: A character vector representing the category of the trait.

References

Pendergrass, S A et al. “The use of phenome-wide association studies (PheWAS) for exploration of novel genotype-phenotype relationships and pleiotropy discovery.” *Genetic epidemiology* vol. 35,5 (2011): 410-22. doi:10.1002/gepi.20589

Examples

```
## Not run:
result <- pheWAS(variant_id = "1_154549918_C_A")
result <- pheWAS(variant_id = "rs72698179")

## End(Not run)
```

plot_coloc

Scatter plot of colocalisations for gene and reported traits.

Description

Generates a scatter plot using the results from `colocalisationsForGene()` function as an input. The reported trait in each study are shown on the *x-axis* and plotted against their corresponding $-\log_2(H4/H3)$ values on the *y-axis*, indicating the evidence of colocalisation between the molecular QTLs reported in each study and the explored gene. The molecular QTLs are mapped to the colors of the points. If the results of `colocalisationsForGene()` includes the data for multiple genes, they will be plotted in separate panels.

Usage

```
plot_coloc(data, biobank = FALSE)
```

Arguments

<code>data</code>	Data Frame: result of <code>colocalisationsForGene</code> function in data frame format, containing the phewas information for a variant id
<code>biobank</code>	Logical: TRUE and FALSE variable, with the default value of FALSE which will keep the data that are from UKbioBank beside the published GWAS data. In case, this parameter is set to TRUE, only UKbioBank data will be kept which has not been published.

Value

A horizontal bar plot for colocalisation of information.

Examples

```
## Not run:
plot_out <- colocalisationsForGene(genes = "ENSG00000169174") %>%
  plot_coloc(biobank = FALSE)
plot_out <- colocalisationsForGene(genes = "PCSK9") %>%
  plot_coloc(biobank = TRUE)

## End(Not run)
```

plot_l2g	<i>Radar plot for L2G partial scores from studiesAndLeadVariantsForGeneByL2G()</i>
----------	--

Description

This function returns a radar plot to compare the partial scores, important for prioritising the causal genes that are obtained from the `studiesAndLeadVariantsForGeneByL2G()` function. The user can decide to plot only for a specific disease by specifying an EFO ID for the disease argument, otherwise the returned plot will facet based on existing traits/diseases in the outputs from `studiesAndLeadVariantsForGeneByL2G()`.

Usage

```
plot_l2g(data, disease_efo = NULL, l2g_cutoff = 0.5, top_n_disease = 1)
```

Arguments

<code>data</code>	Data frame: result of <code>studiesAndLeadVariantsForGeneByL2G</code> function.
<code>disease_efo</code>	Character: Input EFO id to filter the L2G data for a particular disease.
<code>l2g_cutoff</code>	Numeric: Sets the minimum L2G score threshold for diseases to be considered in the plot.
<code>top_n_disease</code>	Numeric: Determines the number of top diseases to plot for each gene, ranked by L2G score.

Value

A radar plot for the input disease and the genes associated with that disease. The variables shown include L2G score, chromatin interaction, variant pathogenicity and distance.

Examples

```
## Not run:
p <- studiesAndLeadVariantsForGeneByL2G(list("ENSG00000167207", "ENSG00000096968",
  "ENSG00000138821", "ENSG00000125255")) %>% plot_l2g(disease = "EFO_0003767")
p

## End(Not run)
```

plot_manhattan	<i>Plot results from manhattan()</i>
----------------	--------------------------------------

Description

This function generates a Manhattan plot using the statistical summary data obtained from the `manhattan()` function. Top 3 genes (based on p-value) are annotated per chromosome.

Usage

```
plot_manhattan(data)
```

Arguments

<code>data</code>	Data frame containing the necessary columns from <code>manhattan()</code> output for plotting: - <code>variant_position</code> : Variant position - <code>variant_chromosome</code> : Variant chromosome - <code>pval</code> : P-value - <code>variant_id</code> : Variant ID - <code>best_locus2genes_score</code> : Best locus2genes score - <code>best_locus2genes_gene_symbol</code> : Best locus2genes gene symbol
-------------------	---

Value

A Manhattan plot visualizing the GWAS results.

Examples

```
## Not run:
p <- manhattan(study_id = "GCST003044") %>% plot_manhattan()

p

## End(Not run)
```

plot_phewas	<i>Plot PheWAS() results.</i>
-------------	-------------------------------

Description

This plot visualizes which traits are associated with the user's selected variant in the UK Biobank, FinnGen, and/or GWAS Catalog summary statistics repository based on PheWAS analysis. The associated traits are mapped onto the *x-axis*, and their corresponding $-\log_{10}(\text{p-value})$ values are plotted on the *y-axis*. A horizontal line is shown at a p-value cutoff of 0.005 to indicate significant associations. Associations above this cutoff are labeled with the trait's name, and the sources of the associations are color-coded as points.

Usage

```
plot_phewas(  
  data,  
  disease = TRUE,  
  source = c("GCST", "FINNGEN", "NEALE", "SAIGE")  
)
```

Arguments

data	Data Frame: The result of the PheWAS() function in data frame format, containing the PheWAS information for a selected variant ID.
disease	Logical: A logical variable indicating whether to filter the PheWAS data for disease (default: TRUE).
source	Character vector: Choices for the data sources of PheWAS analysis, including FINNGEN, GCST, NEALE (UKBiobank), and SAIGE.

Value

A plot to prioritize variants based on their $-\log_{10}(\text{p-value})$.

Examples

```
## Not run:  
p <- pheWAS(variant_id = "14_87978408_G_A") %>%  
  plot_phewas(disease = TRUE)  
p  
  
## End(Not run)
```

qtlColocalisationVariantQuery

Retrieve QTL colocalisation results of a variant

Description

The colocalisation analysis in Open Target Genetics is performed using the coloc method (Giambartolomei et al., 2014). Coloc is a Bayesian method which, for two traits, integrates evidence over all variants at a locus to evaluate the following hypotheses: - H0: No association with either trait - H1: Association with trait 1, not with trait 2 - H2: Association with trait 2, not with trait 1 - H3: Association with trait 1 and trait 2, two independent SNPs - H4: Association with trait 1 and trait 2, one shared SNP This analysis tests whether two independent associations at the same locus are consistent with having a shared causal variant. Colocalisation of two independent associations from two GWAS studies may suggest a shared causal mechanism.

Usage

```
qtlColocalisationVariantQuery(study_id, variant_id)
```

Arguments

study_id	Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).
variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).

Value

Returns a data frame of the colocalisation information for a lead variant in a specific study. The output is a tidy data frame with the following data structure:

- qtlStudyName: *Character vector*. QTL study name.
- phenotypeId: *Character vector*. Phenotype ID.
- gene.id: *Character vector*. Gene ID.
- gene.symbol: *Character vector*. Gene symbol.
- name: *Character vector*. Tissue name.
- indexVariant.id: *Character vector*. Index variant ID.
- indexVariant.rsId: *Character vector*. Index variant rsID.
- beta: *Numeric*. Beta value.
- h4: *Numeric*. h4 value.
- h3: *Numeric*. h3 value.
- log2h4h3: *Numeric*. Log2(h4/h3) value.

Examples

```
## Not run:
result <- qtlColocalisationVariantQuery(study_id = "GCST90002357", variant_id = "1_154119580_C_A")
result <- qtlColocalisationVariantQuery(study_id = "GCST90002357", variant_id = "rs2494663")

## End(Not run)
```

qtlCredibleSet

Retrieve calculated QTL summary statistics for credible variant set.

Description

In Open Targets Genetics, the lead variants are expanded into a more comprehensive set of candidate causal variants referred to as the tag variants. This function retrieves calculated summary statistics for tag variants included in a lead variant colocalization analysis for a given study (which links a top loci with a trait). The user can filter the results by desired biofeature (e.g tissue, cell type,...) the function obtains tag variant information.

Usage

```
qtlCredibleSet(study_id, variant_id, gene, biofeature)
```

Arguments

study_id	Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).
variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).
gene	Character: Gene ENSEMBL ID (e.g. ENSG00000169174) or gene symbol (e.g. PCSK9).
biofeature	Character: Represents either a tissue, cell type, aggregation type, protein type, etc.

Value

Returns a data frame of results from the QTL credible set of variants consisting of the following columns:

- tagVariant.id: *Character vector*. Tag variant ID.
- tagVariant.rsId: *Character vector*. Tag variant rsID.
- pval: *Numeric*. P-value.
- se: *Numeric*. Standard error.
- beta: *Numeric*. Beta value.
- postProb: *Numeric*. Posterior probability.
- MultisignalMethod: *Character vector*. Multisignal method.
- logABF: *Numeric*. Logarithm of approximate Bayes factor.
- is95: *Logical*. Indicates if the variant has a 95
- is99: *Logical*. Indicates if the variant has a 99

Examples

```
## Not run:
result <- qtlCredibleSet(study_id = "Braineac2", variant_id = "1_55053079_C_T",
  gene = "ENSG00000169174", biofeature = "SUBSTANTIA_NIGRA")
result <- qtlCredibleSet(study_id = "Braineac2", variant_id = "rs7552841",
  gene = "PCSK9", biofeature = "SUBSTANTIA_NIGRA")

## End(Not run)
```

run_custom_query

Running custom GraphQL queries

Description

Running custom GraphQL queries

Usage

```
run_custom_query(variableList, query, query_name)
```

Arguments

`variableList` is a list format which includes the key value pair list of genes/variants/study ids to be queries.

`query` is a GraphQL desired query body to be run.

`query_name` is a string format of the query name

Value

a flatten json file format

Examples

```
## Not run:
otarget::run_custom_query (variableList, query, query_name)

## End(Not run)
```

```
studiesAndLeadVariantsForGeneByL2G
```

Retrieve "locus-to-gene" (L2G) model summary data for a gene.

Description

The "locus-to-gene" (L2G) model derives features to prioritize likely causal genes at each GWAS locus based on genetic and functional genomics features. The main categories of predictive features are:

- Distance: Distance from credible set variants to the gene.
- Molecular QTL colocalization: Colocalization with molecular QTLs.
- Chromatin interaction: Interactions, such as promoter-capture Hi-C.
- Variant pathogenicity: Pathogenicity scores from VEP (Variant Effect Predictor).

Usage

```
studiesAndLeadVariantsForGeneByL2G(gene, l2g = NA, pvalue = NA, vtype = NULL)
```

Arguments

gene	Character: Gene ENSEMBL ID (e.g. ENSG00000169174) or gene symbol (e.g. PCSK9). This argument can take a list of genes too.
l2g	Numeric: Locus-to-gene (L2G) cutoff score. (Default: NA)
pvalue	Character: P-value cutoff. (Default: NA)
vtype	Character: Most severe consequence to filter the variant types, including "intergenic_variant", "upstream_gene_variant", "intron_variant", "missense_variant", "5_prime_UTR_variant", "non_coding_transcript_exon_variant", "splice_region_variant". (Default: NULL)

Details

The function also provides additional filtering parameters to narrow the results based following parameters (see below)

Value

Returns a data frame containing the input gene ID and its data for the L2G model. The table consists of the following columns:

- `yProbaModel`: *Numeric*. L2G score.
- `yProbaDistance`: *Numeric*. Distance.
- `yProbaInteraction`: *Numeric*. Chromatin interaction.
- `yProbaMolecularQTL`: *Numeric*. Molecular QTL.
- `yProbaPathogenicity`: *Numeric*. Pathogenicity.
- `pval`: *Numeric*. P-value.
- `beta.direction`: *Character*. Beta direction.
- `beta.betaCI`: *Numeric*. Beta confidence interval.
- `beta.betaCILower`: *Numeric*. Lower bound of the beta confidence interval.
- `beta.betaCIUpper`: *Numeric*. Upper bound of the beta confidence interval.
- `odds.oddsCI`: *Numeric*. Odds ratio confidence interval.
- `odds.oddsCILower`: *Numeric*. Lower bound of the odds ratio confidence interval.
- `odds.oddsCIUpper`: *Numeric*. Upper bound of the odds ratio confidence interval.
- `study.studyId`: *Character*. Study ID.
- `study.traitReported`: *Character*. Reported trait.
- `study.traitCategory`: *Character*. Trait category.
- `study.pubDate`: *Character*. Publication date.
- `study.pubTitle`: *Character*. Publication title.
- `study.pubAuthor`: *Character*. Publication author.
- `study.pubJournal`: *Character*. Publication journal.
- `study.pmid`: *Character*. PubMed ID.

- `study.hasSumstats`: *Logical*. Indicates if the study has summary statistics.
- `study.nCases`: *Integer*. Number of cases in the study.
- `study.numAssocLoci`: *Integer*. Number of associated loci.
- `study.nTotal`: *Integer*. Total number of samples in the study.
- `study.traitEfos`: *Character*. Trait EFOs.
- `variant.id`: *Character*. Variant ID.
- `variant.rsId`: *Character*. Variant rsID.
- `variant.chromosome`: *Character*. Variant chromosome.
- `variant.position`: *Integer*. Variant position.
- `variant.refAllele`: *Character*. Variant reference allele.
- `variant.altAllele`: *Character*. Variant alternate allele.
- `variant.nearestCodingGeneDistance`: *Integer*. Distance to the nearest coding gene.
- `variant.nearestGeneDistance`: *Integer*. Distance to the nearest gene.
- `variant.mostSevereConsequence`: *Character*. Most severe consequence.
- `variant.nearestGene.id`: *Character*. Nearest gene ID.
- `variant.nearestCodingGene.id`: *Character*. Nearest coding gene ID.
- `ensembl_id`: *Character*. Ensembl ID.
- `gene_symbol`: *Character*. Gene symbol.

Examples

```
## Not run:
result <- studiesAndLeadVariantsForGeneByL2G(genes = c("ENSG00000163946",
  "ENSG00000169174", "ENSG00000143001"), l2g = 0.7)
result <- studiesAndLeadVariantsForGeneByL2G(genes = "ENSG00000169174",
  l2g = 0.6, pvalue = 1e-8, vtype = c("intergenic_variant", "intron_variant"))
result <- studiesAndLeadVariantsForGeneByL2G(genes = "TMEM61")

## End(Not run)
```

studyInfo

Retrieve study summary information.

Description

For a given study id, this function returns a data frame of relevant information about the GWAS study, such as PubMed ID, studied trait EFO ID, case/control size, etc.

Usage

```
studyInfo(study_id)
```

Arguments

study_id Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).

Value

Returns a data frame (in tibble format) containing the summary information about a GWAS study. The data frame has the following data structure:

- studyId: *Character*. Study ID.
- traitReported: *Character*. Reported trait.
- source: *Character*. Source.
- traitEfos: *Character*. Trait EFO ID.
- pmid: *Character*. PubMed ID.
- pubDate: *Character*. Publication date.
- pubJournal: *Character*. Publication journal.
- pubTitle: *Character*. Publication title.
- pubAuthor: *Character*. Publication author.
- hasSumstats: *Character*. Indicates if the study has summary statistics.
- ancestryInitial: *Character*. Initial ancestry.
- nInitial: *Character*. Initial sample size.
- nReplication: *Character*. Replication sample size.
- traitCategory: *Character*. Trait category.
- numAssocLoci: *Character*. Number of associated loci.
- nTotal: *Character*. Total sample size.

Examples

```
## Not run:  
result <- studyInfo(study_id = "GCST90002357")  
  
## End(Not run)
```

studyLocus2GeneTable Retrieve the locus-to-gene (L2G) data table for loci genes.

Description

This function fetches the locus-to-gene (L2G) pipeline summary data table for the neighboring genes of a variant in a GWAS study.

Usage

```
studyLocus2GeneTable(study_id, variant_id)
```

Arguments

study_id	Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).
variant_id	Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).

Value

Returns a data frame with the summary statistics of the study and a data table containing various calculated scores and features for any lead variant. The output table has the following data structure:

- studyId: *Character*. Study ID.
- variant.id: *Character*. Variant ID.
- variant.rsId: *Character*. Variant rsID.
- yProbaDistance: *Numeric*. Distance score.
- yProbaModel: *Numeric*. Model score.
- yProbaMolecularQTL: *Numeric*. Molecular QTL score.
- yProbaPathogenicity: *Numeric*. Pathogenicity score.
- yProbaInteraction: *Numeric*. Interaction score.
- hasColoc: *Logical*. Indicates if colocalization data is available.
- distanceToLocus: *Numeric*. Distance to the locus.
- gene.id: *Character*. Gene ID.
- gene.symbol: *Character*. Gene symbol.

Examples

```
## Not run:
result <- studyLocus2GeneTable(study_id = "GCST90002357", variant_id = "1_154119580_C_A")
result <- studyLocus2GeneTable(study_id = "GCST90002357", variant_id = "rs2494663")

## End(Not run)
```

studyVariants	<i>Retrieves all variants for a study.</i>
---------------	--

Description

For an input study ID, this function returns information of all variants across associated loci. The output also includes information about the associated genes within the each loci.

Usage

```
studyVariants(study_id)
```

Arguments

study_id Character: Study ID(s) generated by Open Targets Genetics (e.g GCST90002357).

Value

Returns a list of two data frames.

the first data frame (tibble format) includes the loci data frame with following data structure:

- variant.id: *Character*. Variant ID.
- pval: *Numeric*. P-value.
- variant.nearestCodingGene.symbol: *Character*. Symbol of the nearest coding gene to the variant.
- variant.rsId: *Character*. Variant rsID.
- variant.chromosome: *Character*. Chromosome of the variant.
- variant.position: *Integer*. Position of the variant.
- variant.nearestCodingGeneDistance: *Integer*. Distance to the nearest coding gene.
- credibleSetSize: *Integer*. Size of the credible set.
- ldSetSize: *Integer*. Size of the LD set.
- oddsRatio: *Numeric*. Odds ratio.
- beta: *Numeric*. Beta value.

The second data frame includes gene information with following data structure:

- score: *Numeric*. Gene score.
- gene.id: *Character*. Gene ID.
- gene.symbol: *Character*. Gene symbol.

Examples

```
## Not run:  
result <- studyVariants(study_id = "GCST003155")  
  
## End(Not run)
```

tagVariantsAndStudiesForIndexVariant

Retrieves tag variants and studies for a given index variant.

Description

For an input index variant ID, this function fetches information about the tag variants and associated studies, including scores.

Usage

```
tagVariantsAndStudiesForIndexVariant(variant_id, pageindex = 0, pagesize = 20)
```

Arguments

variant_id	Character: Open Targets Genetics generated ID for a variant (CHRPOSITION_REFALLELE_ALTALLELE or rsId).
pageindex	Integer: Index of the current page for pagination (≥ 0).
pagesize	Integer: Number of records in a page for pagination (> 0).

Value

Returns a data frame containing the variant associations connected to the input index variant. The columns in the data frame are as follows:

- tagVariant.id: *Character*. Tag variant ID.
- tagVariant.chromosome: *Character*. Chromosome of the tag variant.
- tagVariant.rsId: *Character*. rsID of the tag variant.
- tagVariant.position: *Integer*. Position of the tag variant.
- study.studyId: *Character*. Study ID.
- study.traitReported: *Character*. Reported trait of the study.
- study.traitCategory: *Character*. Category of the trait in the study.
- pval: *Numeric*. P-value.
- pvalMantissa: *Numeric*. Mantissa of the p-value.
- pvalExponent: *Integer*. Exponent of the p-value.
- nTotal: *Integer*. Total number of samples.
- nCases: *Integer*. Number of cases in the study.
- overallR2: *Numeric*. Overall R-squared value.
- afr1000GProp: *Numeric*. Proportion in African 1000 Genomes population.
- amr1000GProp: *Numeric*. Proportion in Admixed American 1000 Genomes population.
- eas1000GProp: *Numeric*. Proportion in East Asian 1000 Genomes population.
- eur1000GProp: *Numeric*. Proportion in European 1000 Genomes population.
- sas1000GProp: *Numeric*. Proportion in South Asian 1000 Genomes population.
- oddsRatio: *Numeric*. Odds ratio.
- oddsRatioCILower: *Numeric*. Lower bound of the odds ratio confidence interval.
- oddsRatioCIUpper: *Numeric*. Upper bound of the odds ratio confidence interval.
- posteriorProbability: *Numeric*. Posterior probability.
- beta: *Numeric*. Beta value.
- betaCILower: *Numeric*. Lower bound of the beta value confidence interval.
- betaCIUpper: *Numeric*. Upper bound of the beta value confidence interval.
- direction: *Character*. Direction of the effect.
- log10Abf: *Numeric*. Log base 10 of the approximate Bayes factor.

Examples

```
## Not run:
result <- tagVariantsAndStudiesForIndexVariant(variant_id = "1_109274968_G_T")
result <- tagVariantsAndStudiesForIndexVariant(variant_id = "1_109274968_G_T"
,pageindex = 1, pagesize = 50)

## End(Not run)
```

topOverlappedStudies *Retrieve top studies having overlap in identified loci.*

Description

For a provided study ID, the function, retrieves top studies with overlap in their identified loci with the queried study loci.

Usage

```
topOverlappedStudies(study_id, pageindex = 0, pagesize = 20)
```

Arguments

study_id	Character: Open Targets Genetics generated ID for a GWAS study.
pageindex	Integer: Index of the current page for pagination (≥ 0).
pagesize	Integer: Number of records in a page for pagination (> 0).

Value

Returns a data frame with the top studies containing the following columns:

- `study.studyId`: *Character*. Study ID of the input study.
- `study.traitReported`: *Character*. Reported trait of the input study.
- `study.traitCategory`: *Character*. Category of the trait in the input study.
- `topStudiesByLociOverlap.studyId`: *Character*. Study ID of the top associated studies.
- `topStudiesByLociOverlap.study.studyId`: *Character*. Study ID of the top associated studies.
- `topStudiesByLociOverlap.study.traitReported`: *Character*. Reported trait of the top associated studies.
- `topStudiesByLociOverlap.study.traitCategory`: *Character*. Category of the trait in the top associated studies.
- `topStudiesByLociOverlap.numOverlapLoci`: *Integer*. Number of loci overlapped with the input study.

Examples

```
## Not run:
result <- topOverlappedStudies(study_id = "GCST006614_3")
result <- topOverlappedStudies(study_id = "NEALE2_6177_1", pageindex = 1, pagesize = 50)

## End(Not run)
```

variantInfo

Retrieves information about a variant.

Description

For a given variant ID, this function retrieves information about the variant, including its chromosome, position, reference allele, alternative allele, rsID, nearest gene, most severe consequence, and allele frequencies in different populations from gnomAD database. The Genome Aggregation Database (gnomAD) is a resource developed by an international coalition of investigators, with the goal of aggregating and harmonizing both exome and genome sequencing data from a wide variety of large-scale sequencing projects, and making summary data available for the wider scientific community (see the reference).

Usage

```
variantInfo(variant_id)
```

Arguments

`variant_id` Character: generated ID for variants by Open Targets Genetics (e.g. 1_154119580_C_A) or rsId (rs2494663).

Value

Returns a data frame (in tibble format) containing information about the variant. The data frame has the following structure:

- `chromosome`: *Character*. Chromosome of the variant.
- `position`: *Integer*. Position of the variant.
- `refAllele`: *Character*. Reference allele.
- `altAllele`: *Character*. Alternative allele.
- `rsId`: *Character*. Variant rsID.
- `chromosomeB37`: *Character*. Chromosome of the variant in build 37 coordinates.
- `positionB37`: *Integer*. Position of the variant in build 37 coordinates.
- `id`: *Character*. Variant ID.
- `nearestGene.id`: *Character*. ID of the nearest gene to the variant.
- `nearestGene.symbol`: *Character*. Symbol of the nearest gene to the variant.

- nearestGeneDistance: *Integer*. Distance to the nearest gene.
- nearestCodingGene.id: *Character*. ID of the nearest coding gene to the variant.
- nearestCodingGene.symbol: *Character*. Symbol of the nearest coding gene to the variant.
- nearestCodingGeneDistance: *Integer*. Distance to the nearest coding gene.
- mostSevereConsequence: *Character*. Most severe consequence of the variant.
- caddRaw: *Numeric*. CADD raw score.
- caddPhred: *Numeric*. CADD phred score.
- gnomadAFR: *Numeric*. Allele frequency in the African/African-American population in gnomAD.
- gnomadAMR: *Numeric*. Allele frequency in the Latino/Admixed American population in gnomAD.
- gnomadASJ: *Numeric*. Allele frequency in the Ashkenazi Jewish population in gnomAD.
- gnomadEAS: *Numeric*. Allele frequency in the East Asian population in gnomAD.
- gnomadFIN: *Numeric*. Allele frequency in the Finnish population in gnomAD.
- gnomadNFE: *Numeric*. Allele frequency in the Non-Finnish European population in gnomAD.
- gnomadNFEEST: *Numeric*. Allele frequency in the Estonian population in gnomAD.
- gnomadNFENWE: *Numeric*. Allele frequency in the Northwest European population in gnomAD.
- gnomadNFSEU: *Numeric*. Allele frequency in the Southern European population in gnomAD.
- gnomadNFEONF: *Numeric*. Allele frequency in the Other Non-Finnish European population in gnomAD.
- gnomadOTH: *Numeric*. Allele frequency in other populations in gnomAD.

References

<https://gnomad.broadinstitute.org/>

Examples

```
## Not run:  
result <- variantInfo(variant_id = "rs2494663")  
  
## End(Not run)
```

Index

chemblQuery, [2](#)
clinvarQuery, [3](#)
colocalisationsForGene, [4](#)
compGenomicsQuery, [5](#)

geneInfo, [6](#)
genesForVariant, [7](#)
geneticConstraintQuery, [8](#)
getLociGenes, [8](#)
gwasColocalisation, [9](#)
gwasColocalisationForRegion, [11](#)
gwasCredibleSet, [12](#)
gwasRegional, [13](#)

indexVariantsAndStudiesForTagVariant,
[14](#)

knownDrugsQuery, [15](#)

manhattan, [16](#)
mousePhenotypesQuery, [17](#)

overlapInfoForStudy, [18](#)

pheWAS, [19](#)
plot_coloc, [20](#)
plot_l2g, [21](#)
plot_manhattan, [22](#)
plot_phewas, [22](#)

qtlColocalisationVariantQuery, [23](#)
qtlCredibleSet, [24](#)

run_custom_query, [25](#)

studiesAndLeadVariantsForGeneByL2G, [26](#)
studyInfo, [28](#)
studyLocus2GeneTable, [29](#)
studyVariants, [30](#)

tagVariantsAndStudiesForIndexVariant,
[31](#)

topOverlappedStudies, [33](#)
variantInfo, [34](#)