# Package: snplinkage (via r-universe)

September 18, 2024

**Title** Single Nucleotide Polymorphisms Linkage Disequilibrium Visualizations

**Version** 1.2.0

**Description** Linkage disequilibrium visualizations of up to several hundreds of single nucleotide polymorphisms (SNPs), annotated with chromosomic positions and gene names. Two types of plots are available for small numbers of SNPs (<40) and for large numbers (tested up to 500). Both can be extended by combining other ggplots, e.g. association studies results, and functions enable to directly visualize the effect of SNP selection methods, as minor allele frequency filtering and TagSNP selection, with a second correlation heatmap. The SNPs correlations are computed on Genotype Data objects from the 'GWASTools' package using the 'SNPRelate' package, and the plots are customizable 'ggplot2' and 'gtable' objects and are annotated using the 'biomaRt' package. Usage is detailed in the vignette with example data and results from up to 500 SNPs of 1,200 scans are in Charlon T. (2019) <doi:10.13097/archive-ouverte/unige:161795>.

**Imports** gdsfmt, ggplot2, gtable, magrittr, stats, utils

**Depends** R (>= 2.15), GWASTools (>= 1.10.1)

**Suggests** biomaRt, cowplot, data.table, dplyr, ggrepel, grid, grDevices, knitr, methods, plyr, reshape2, rmarkdown, SNPRelate, testthat

**biocViews** GeneticVariability, MicroArray, SNP

**URL** https://gitlab.com/thomaschln/snplinkage

**BugReports** https://gitlab.com/thomaschln/snplinkage/-/issues

**VignetteBuilder** knitr

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Repository** https://thomaschln.r-universe.dev

**RemoteUrl** https://gitlab.com/thomaschln/snplinkage

**RemoteRef** HEAD

**RemoteSha** f3ec99994c86e0a4654b0cfb1db1d5cab766853d

# Contents

---

`chisq_pvalues` *Compute Chi-squared p-values*

---

## Description

Compute Chi-squared p-values

## Usage

```
chisq_pvalues(
  m_data,
  response,
  adjust_method = "fdr",
  mlog10_transform = TRUE,
  n_cores = 1,
  ...
)
```

## Arguments

| | |
|---|---|
| `m_data` | Data matrix of observations by variables |
| `response` | Response vector of length the number of observations |
| `adjust_method` | Multiple testing p-value adjustment method. Passed to stats::p.adjust. 'fdr' by default. |
| `mlog10_transform` | Logical, transform p-values by minus log10. True by default. |
| `n_cores` | Number of cores |
| `...` | Passed to stats::chisq.test |

## Value

Chi-squared p-values

## chisq_pvalues_gdata *Compute Chi-squared p-values on a Genotype data object*

### Description

Compute Chi-squared p-values on a Genotype data object

### Usage

```
chisq_pvalues_gdata(
  gdata,
  snp_idxs,
  response_column = "region",
  response_value = "Europe",
  threshold = 2,
  ...
)
```

### Arguments

| | |
|---|---|
| gdata | Genotype data object |
| snp_idxs | SNPs indexes |
| response_column | Response column in gdata scans annotations data frame |
| response_value | Response value. The response vector will be a logical, true if equal to the value, false otherwise. |
| threshold | Keep only associations greater than the threshold |
| ... | Passed to chisq_pvalues |

### Value

SNPs annotation data frame, chi-squared p-values in column pvalues

## crohn *Crohn's disease data*

### Description

The data set consist of 103 common (>5% minor allele frequency) SNPs genotyped in 129 trios from an European-derived population. These SNPs are in a 500-kb region on human chromosome 5q31 implicated as containing a genetic risk factor for Crohn disease.

Imported from the gap R package.

An example use of the data is with the following paper, Kelly M. Burkett, Celia M. T. Greenwood, BradMcNeney, Jinko Graham. Gene genealogies for genetic association mapping, with application to Crohn's disease. Fron Genet 2013, 4(260) doi: 10.3389/fgene.2013.00260

## Usage

```
data(crohn)
```

## Format

A data frame containing 387 rows and 212 columns

## Source

MJ Daly, JD Rioux, SF Schaffner, TJ Hudson, ES Lander (2001) High-resolution haplotype structure in the human genome Nature Genetics 29:229-232

---

# diamond_annots *Get diamond ggplot layer.*

---

## Description

Diamond ggplot layer for ggplot_ld

## Usage

```
diamond_annots(data, x = "x", y = "y", color = "color", size = 0.5)
```

## Arguments

| | |
|---|---|
| data | Data frame of 3 columns defining the diamonds |
| x | Name of the column for horizontal positions |
| y | Name of the column for vertical positions |
| color | Name of the column for color values |
| size | Radius of the diamonds |

## Value

gglayers

## fetch_allele1.default *Fetch allele 1 (default object)*

### Description

Fetch allele 1 (default object)

### Usage

```
## Default S3 method:
fetch_allele1(obj, snps_idx)
```

### Arguments

| | |
|---|---|
| obj | Default object |
| snps_idx | SNPs indexes |

## fetch_allele1.GdsGenotypeReader *Fetch allele 1 (GdsGenotypeReader object)*

### Description

Fetch allele 1 (GdsGenotypeReader object)

### Usage

```
## S3 method for class 'GdsGenotypeReader'
fetch_allele1(obj, snps_idx)
```

### Arguments

| | |
|---|---|
| obj | GenotypeData object |
| snps_idx | SNPs indexes |

### Value

Allele 1

---

`fetch_allele1.GenotypeData`
*Fetch allele 1 (GenotypeData object)*

---

## Description

Fetch allele 1 (GenotypeData object)

## Usage

```
## S3 method for class 'GenotypeData'
fetch_allele1(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeData object |
| `...` | Passed to getAlleleA |

## Value

Allele 1

---

`fetch_allele1.GenotypeDataSubset`
*Fetch allele 1 (GenotypeDataSubset object)*

---

## Description

Fetch allele 1 (GenotypeDataSubset object)

## Usage

```
## S3 method for class 'GenotypeDataSubset'
fetch_allele1(obj, snps_idx)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeDataSubset object |
| `snps_idx` | SNPs indexes |

## Value

Allele 1

## fetch_allele2.default *Fetch allele 2 (default object)*

### Description

Fetch allele 2 (default object)

### Usage

```
## Default S3 method:
fetch_allele2(obj, snps_idx)
```

### Arguments

| | |
|---|---|
| obj | Default object |
| snps_idx | SNPs indexes |

## fetch_allele2.GdsGenotypeReader *Fetch allele 2 (GdsGenotypeReader object)*

### Description

Fetch allele 2 (GdsGenotypeReader object)

### Usage

```
## S3 method for class 'GdsGenotypeReader'
fetch_allele2(obj, snps_idx)
```

### Arguments

| | |
|---|---|
| obj | GenotypeData object |
| snps_idx | SNPs indexes |

### Value

Allele 2

---

`fetch_allele2.GenotypeData` *Fetch allele 2 (GenotypeData object)*

---

## Description

Fetch allele 2 (GenotypeData object)

## Usage

```
## S3 method for class 'GenotypeData'
fetch_allele2(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeData object |
| `...` | Passed to getAlleleB |

## Value

Allele 2

---

`fetch_allele2.GenotypeDataSubset` *Fetch allele 1 (GenotypeDataSubset object)*

---

## Description

Fetch allele 1 (GenotypeDataSubset object)

## Usage

```
## S3 method for class 'GenotypeDataSubset'
fetch_allele2(obj, snps_idx)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeDataSubset object |
| `snps_idx` | SNPs indexes |

## Value

Allele 2

## fetch_gds.default *Fetch GDS (default)*

### Description

Fetch GDS (default)

### Usage

```
## Default S3 method:
fetch_gds(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | Default object |
| ... | Not passed |

## fetch_gds.GdsGenotypeReader *Fetch GDS (GdsGenotypeReader)*

### Description

Fetch GDS (GdsGenotypeReader)

### Usage

```
## S3 method for class 'GdsGenotypeReader'
fetch_gds(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | GdsGenotypeReader object |
| ... | Not passed |

### Value

S4 slot 'handler' of obj

`fetch_gds.GenotypeData` *Fetch GDS (GenotypeData)*

## Description

Fetch GDS (GenotypeData)

## Usage

```
## S3 method for class 'GenotypeData'
fetch_gds(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeData object |
| `...` | Not passed |

## Value

fetch_gds output on S4 slot 'data' of obj

`fetch_gds.GenotypeDataSubset` *Fetch GDS (GenotypeDataSubset)*

## Description

Fetch GDS (GenotypeDataSubset)

## Usage

```
## S3 method for class 'GenotypeDataSubset'
fetch_gds(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeDataSubset object |
| `...` | Not passed |

## gdata_add_gene_annots *gdata_add_gene_annots*

### Description

Add biomaRt gene annotations to Genotype Data object.

### Usage

```
gdata_add_gene_annots(
  gdata,
  snp_idxs,
  rsids_colname = "probe_id",
  biomart_metadb = get_biomart_metadb()
)
```

### Arguments

| | |
|---|---|
| gdata | Genotype Data object |
| snp_idxs | SNP indexes |
| rsids_colname | Column of SNP annotation data frame with rs identifiers |
| biomart_metadb | List with slots snpmart and ensembl, corresponding to the biomart databases to query for SNP identifiers and gene names, respectively. See get_biomart_metadb function. |

### Value

Genotype Data object

## gdata_add_gene_annots_aim_example *gdata_add_gene_annots_aim_example*

### Description

Add ancestry informative markers gene annotations to Genotype Data object. Convenience function for the vignette to avoid querying biomaRt on build.

### Usage

```
gdata_add_gene_annots_aim_example(gdata, aim_idxs)
```

### Arguments

| | |
|---|---|
| gdata | Genotype Data object |
| aim_idxs | AIM indexes in the example Genotype data object |

### Value

Genotype Data object

---

`gdata_add_gene_annots_hladr_example` *gdata_add_gene_annots_hladr_example*

---

### Description

Add HLA-DR gene annotations to Genotype Data object. Convenience function for the vignette to avoid querying biomaRt on build.

### Usage

```
gdata_add_gene_annots_hladr_example(gdata, hla_dr_idxs)
```

### Arguments

| | |
|---|---|
| `gdata` | Genotype Data object |
| `hla_dr_idxs` | HLA-DR indexes in the example Genotype data object |

### Value

Genotype Data object

---

`gdata_scans_annots` *gdata_scan_annots*

---

### Description

Get scans annotations from a Genotype Data object or a subset.

### Usage

```
gdata_scans_annots(gdata, scan_ids)
```

### Arguments

| | |
|---|---|
| `gdata` | Genotype Data object |
| `scan_ids` | Scan identifiers to subset |

### Value

Scans annotations data frame

## gdata_snps_annots *gdata_snp_annots*

### Description

Get SNPs annotations from a Genotype Data object or a subset.

### Usage

```
gdata_snps_annots(gdata, snp_ids = NULL)
```

### Arguments

| | |
|---|---|
| gdata | Genotype Data object |
| snp_ids | SNP identifiers to subset |

### Value

SNP annotation data frame

## get_biomart_metadb *get_biomart_metadb*

### Description

To query gene names of SNPs, it is necessary to retrieve two objects using biomaRt::useMart. First, the object required to map SNP rs identifiers to ENSEMBL identifiers. Second, the object required to map ENSEMBL identifiers to common gene names. The function returns a list of two slots named snpmart and ensembl corresponding to each one, respectively. Once obtained it is saved to a local file.

### Usage

```
get_biomart_metadb(
  filepath = extdata_filepath("bmart_meta.rds"),
  host = "https://grch37.ensembl.org"
)
```

### Arguments

| | |
|---|---|
| filepath | Path to save the biomaRt objects |
| host | BiomaRt Ensembl host, by default https://grch37.ensembl.org |

### Value

List of slots snpmart and ensembl as detailed above

## get_scan_annot.GenotypeData *Get scans annotations (GenotypeData object)*

### Description

Get scans annotations (GenotypeData object)

### Usage

```
## S3 method for class 'GenotypeData'
get_scan_annot(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | GenotypeData object |
| ... | Not passed |

### Value

Data frame

## get_scan_annot.GenotypeDataSubset *Get scans annotations (GenotypeDataSubset object)*

### Description

Get scans annotations (GenotypeDataSubset object)

### Usage

```
## S3 method for class 'GenotypeDataSubset'
get_scan_annot(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | GenotypeDataSubset object |
| ... | Not passed |

### Value

Data frame

---

`get_snp_annot.GenotypeData` *Get SNPs annotations (GenotypeData object)*

---

## Description

Get SNPs annotations (GenotypeData object)

## Usage

```
## S3 method for class 'GenotypeData'
get_snp_annot(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeData object |
| `...` | Not passed |

## Value

Data frame

---

`get_snp_annot.GenotypeDataSubset` *Get SNPs annotations (GenotypeDataSubset object)*

---

## Description

Get SNPs annotations (GenotypeDataSubset object)

## Usage

```
## S3 method for class 'GenotypeDataSubset'
get_snp_annot(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | GenotypeDataSubset object |
| `...` | Not passed |

## Value

Data frame

`ggplot_associations` *Ggplot associations*

## Description

Get SNPs associations ggplot, either as points or as a linked area. Optionally add labels to most associated points using ggrepel.

## Usage

```
ggplot_associations(
  df_snp,
  pvalue_colname = "pvalues",
  labels_colname = "probe_id",
  n_labels = 10,
  nudge = c(0, 1),
  linked_area = FALSE,
  byindex = linked_area,
  colors = if (linked_area) snp_position_colors(nrow(df_snp)) else "black"
)
```

## Arguments

| | |
|---|---|
| `df_snp` | SNP annotation data frame with columns chromosome, position, and as specified by parameters pvalue_colname and optionally labels_colname. |
| `pvalue_colname` | Column name of df_snp with association values |
| `labels_colname` | Optional column name of df_snp with labels. Set to NULL to remove. |
| `n_labels` | Number of labels of most associated points to display. |
| `nudge` | Nudge parameter passed to ggrepel::geom_label_repel. |
| `linked_area` | Add a linked area to associations points, default FALSE |
| `byindex` | Display by SNP index or chromosomic position (default) |
| `colors` | Colors of SNPs |

## Value

ggplot

## ggplot_ld *Ggplot linkage disequilibrium*

### Description

Display SNP r2 correlations using points or diamonds with text.

### Usage

```
ggplot_ld(
  df_ld,
  diamonds = length(unique(df_ld$SNP_A)) < 40,
  point_size = 120/sqrt(nrow(df_ld)),
  reverse = FALSE,
  reindex = TRUE
)
```

### Arguments

| | |
|---|---|
| df_ld | Data frame with columns SNP_A, SNP_B, and R2. As returned by the snprelate_ld function. |
| diamonds | Should the values be displayed as diamonds or points ? Default is TRUE for less than 40 SNPs. |
| point_size | Size for geom_point. Ignored if diamonds is TRUE. |
| reverse | Reverse the display (horizontal symmetry) |
| reindex | If FALSE, SNPs are positionned following their IDs |

### Value

ggplot

## ggplot_snp_pos *Ggplot SNPs position*

### Description

Get SNPs position ggplot with mappings to combine with other ggplots. Optionally add labels and an upper subset.

### Usage

```
ggplot_snp_pos(
  df_snp,
  upper_subset = NULL,
  labels_colname = NULL,
  colors = snp_position_colors(nrow(df_snp))
)
```

### Arguments

| | |
|---|---|
| df_snp | SNP annotation data frame with a column named position and, if specified, one named as the labels_colname parameter. |
| upper_subset | Subset of df_snp for the positions on the upper side |
| labels_colname | Optional column name of df_snp to use as SNP labels. |
| colors | Colors for each SNP |

### Value

ggplot

---

## gtable_ld *Gtable of linkage disequilibrium and chromosomic positions*

---

### Description

Creates a gtable of linkage disequilibrium and chromosomic positions ggplots. A biplot_subset parameter is available to add a second linkage disequibrium ggplot to visualize the effect of a SNP selection.

### Usage

```
gtable_ld(
  df_ld,
  df_snp,
  biplot_subset = NULL,
  labels_colname = NULL,
  diamonds = length(unique(df_ld$SNP_A)) < 40,
  point_size = ifelse(is.null(biplot_subset), 120, 80)/sqrt(nrow(df_ld)),
  title = "",
  title_biplot = "",
  ...
)
```

### Arguments

| | |
|---|---|
| df_ld | Data frame returned by snprelate_ld |
| df_snp | SNP annotations with columns snpID and position |
| biplot_subset | SNP indexes of the subset for the second ld plot |
| labels_colname | Column name of df_snp to use as SNP labels |
| diamonds | Display the values as diamonds or as points Default is TRUE for less than 40 SNPs. |
| point_size | Size for geom_point. Ignored if diamonds is TRUE. |
| title | Plot title |
| title_biplot | Optional biplot title |
| ... | Passed to ggplot_ld |

## Value

gtable of ggplots

## Examples

```
library(snplinkage)
gds_path <- save_hgdp_as_gds()
gdata <- load_gds_as_genotype_data(gds_path)
qc <- snprelate_qc(gdata, tagsnp = .99)
snp_idxs_8p23 <- select_region_idxs(qc$gdata, chromosome = 8,
  position_min = 11e6, position_max = 12e6)

df_ld <- snprelate_ld(qc$gdata, snps_idx = snp_idxs_8p23, quiet = TRUE)
plt <- gtable_ld(df_ld, df_snp = gdata_snps_annots(qc$gdata))
```

---

# gtable_ld_associations

*Gtable of linkage disequilibrium and associations*

---

## Description

Creates a gtable of a linkage disequilibrium, chromosomic positions, and association scores ggplots.

## Usage

```
gtable_ld_associations(
  df_assocs,
  df_ld,
  pvalue_colname = "pvalues",
  labels_colname = "probe_id",
  n_labels = 5,
  diamonds = nrow(df_assocs) <= 40,
  linked_area = diamonds,
  point_size = 150/nrow(df_assocs),
  colors = snp_position_colors(nrow(df_assocs)),
  ...
)
```

## Arguments

| | |
|---|---|
| df_assocs | SNP annotation data frame with columns chromosome, position, and as specified by parameters pvalue_colname and optionally labels_colname. |
| df_ld | Data frame with columns SNP_A, SNP_B, and R2, as returned by the snprelate_ld function. |
| pvalue_colname | Column name of df_snp with association values |
| labels_colname | Optional column name of df_snp with labels. Set NULL to remove labels. |

| | |
|---|---|
| n_labels | Number of labels of most associated SNPs to display. |
| diamonds | Should the values be displayed as diamonds or points ? Default is TRUE for up to 40 SNPs. |
| linked_area | Add a linked area to associations points. Default same as diamonds. |
| point_size | Point size for ggplot_ld, ignored if diamonds is TRUE. |
| colors | Colors of SNPs |
| ... | Passed to ggplot_associations |

## Value

gtable

---

`gtable_ld_associations_combine`

*Build gtable by combining ggplots*

---

## Description

Build gtable by combining ggplots

## Usage

```
gtable_ld_associations_combine(ggplots, diamonds)
```

## Arguments

| | |
|---|---|
| ggplots | List of ggplots |
| diamonds | Does the LD visualization use diamond-type layout |

## Value

gtable of ggplots

## Examples

```
library(snplinkage)

# example rnaseq data frame, 20 variables of 20 patients
m_rna = matrix(runif(20 ^ 2), nrow = 20)

# pair-wise correlation matrix
m_ld = cor(m_rna) ^ 2

# keep only upper triangle and reshape to data frame
m_ld[lower.tri(m_ld, diag = TRUE)] = NA
df_ld = reshape2::melt(m_ld) |> na.omit()
```

```
# rename for SNPLinkage
names(df_ld) = c('SNP_A', 'SNP_B', 'R2')

# visualize with ggplot_ld
gg_ld = ggplot_ld(df_ld)

# let's imagine the 20 variables came from 3 physically close regions
positions = c(runif(7, 10e5, 15e5), runif(6, 25e5, 30e5),
              runif(7, 45e5, 50e5)) |> sort()

# build the dataframe
df_snp_pos = data.frame(position = positions)
df_snp_pos$label = c(rep('HLA-A', 7), rep('HLA-B', 6), rep('HLA-C', 7))

gg_pos_biplot = ggplot_snp_pos(df_snp_pos, labels_colname = 'label',
                               upper_subset = TRUE)

# let's assume HLA-B is more associated with the outcome than the other genes
pvalues = c(runif(7, 1e-3, 1e-2), runif(6, 1e-8, 1e-6), runif(7, 1e-3, 1e-2))
log10_pvals = -log10(pvalues)

# we can reuse the df_snp_pos object
df_snp_pos$pvalues = log10_pvals

# add the chromosome column
df_snp_pos$chromosome = 6

gg_assocs = ggplot_associations(df_snp_pos, labels_colname = 'label',
                                linked_area = TRUE, nudge = c(0, 0.5),
                                n_labels = 12)

l_ggs = list(pos = gg_pos_biplot, ld = gg_ld, pval = gg_assocs)
gt_ld = gtable_ld_associations_combine(l_ggs, diamonds = TRUE)
grid::grid.draw(gt_ld)
```

---

`gtable_ld_associations_gdata` *Gtable of linkage disequilibrium and associations using a GenotypeData object*

---

## Description

Compute linkage disequilibrium using snprelate_ld on the set of SNPs in the associations data frame and call gtable_ld_associations. Creates a gtable of a linkage disequilibrium, chromosomic positions, and association scores ggplots.

## Usage

```
gtable_ld_associations_gdata(
```

```
  df_assocs,
  gdata,
  pvalue_colname = "pvalues",
  labels_colname = "probe_id",
  diamonds = nrow(df_assocs) <= 40,
  window = 15,
  ...
)
```

## Arguments

| | |
|---|---|
| df_assocs | SNP annotation data frame with columns chromosome, position, and as specified by parameters pvalue_colname and optionally labels_colname. |
| gdata | GenotypeData object, as returned by load_gds_as_genotype_data |
| pvalue_colname | Column name of df_snp with association values |
| labels_colname | Optional column name of df_snp with labels. Set NULL to remove labels. |
| diamonds | Should the values be displayed as diamonds or points ? Default is TRUE for up to 40 SNPs. |
| window | Window size for snprelate_ld. Forced to the total number of SNPs if diamonds is FALSE |
| ... | Passed to gtable_ld_associations |

## Value

gtable

## Examples

```
library(snplinkage)
gds_path <- save_hgdp_as_gds()
gdata <- load_gds_as_genotype_data(gds_path)
qc <- snprelate_qc(gdata, tagsnp = .99)

snp_idxs_mhc <- select_region_idxs(qc$gdata,
  chromosome = 6, position_min = 29e6, position_max = 33e6)
df_assocs <- chisq_pvalues_gdata(qc$gdata, snp_idxs_mhc)

df_top_aim <- subset(df_assocs, rank(-pvalues, ties.method = 'first') <= 20)

#qc$gdata <- gdata_add_gene_annots(qc$gdata, rownames(df_top_aim))
qc$gdata <- gdata_add_gene_annots_aim_example(qc$gdata, rownames(df_top_aim))

plt <- gtable_ld_associations_gdata(df_top_aim, qc$gdata,
  labels_colname = 'gene')
```

gtable_ld_gdata *Gtable of linkage disequilibrium and positions using a GenotypeData object*

## Description

Compute linkage disequilibrium using snprelate_ld on a set of SNP indexes and call gtable_ld. Two parameters are available to compute and compare minor allele frequency filtering and TagSNP selection by displaying two LD plots with their positions in the center. The maf and r2 parameters are used similarly and as follows: - compare baseline with MAF 5 gtable_ld(gdata, snps_idx, maf = 0.05) - compare baseline with TagSNP r2 = 0.8 gtable_ld(gdata, snps_idx, r2 = 0.8) - compare 5 gtable_ld(gdata, snps_idx, maf = c(0.05, 0.05), r2 = 0.8) - compare MAF 5 gtable_ld(gdata, snps_idx, maf = c(0.05, 0.1), r2 = c(0.8, 0.6))

## Usage

```
gtable_ld_gdata(
  gdata,
  snps_idx,
  maf = NULL,
  r2 = NULL,
  diamonds = length(snps_idx) < 40,
  window = 15,
  autotitle = TRUE,
  autotitle_bp = TRUE,
  double_title = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| gdata | GenotypeData object returned by load_gds_as_genotype_data |
| snps_idx | SNPs indexes to select |
| maf | Minor allele frequency threshold(s), see description |
| r2 | TagSNP r2 threshold(s), see description |
| diamonds | Display the values as diamonds or as points Default is TRUE for less than 40 SNPs. |
| window | Window size for snprelate_ld. Forced to the total number of SNPs if diamonds is FALSE |
| autotitle | Set title to feature selection method(s), number of SNPs and chromosome |
| autotitle_bp | Set biplot title to feature selection method(s), number of SNPs and chromosome |
| double_title | Logical, if false (default) keep only biplot title |
| ... | Passed to gtable_ld |

## Value

gtable of ggplots

## Examples

```
library(snplinkage)
gds_path <- save_hgdp_as_gds()
gdata <- load_gds_as_genotype_data(gds_path)
qc <- snprelate_qc(gdata, tagsnp = .99)

snp_idxs_1p13_large <- select_region_idxs(qc$gdata, chromosome = 1,
  position_min = 114e6, n_snps = 100)
plt <- gtable_ld_gdata(qc$gdata, snp_idxs_1p13_large)
```

---

# gtable_ld_grobs *Build gtable by combining ggplots*

---

## Description

Build gtable by combining ggplots

## Usage

```
gtable_ld_grobs(plots, labels_colname, title)
```

## Arguments

| | |
|---|---|
| plots | List of ggplots |
| labels_colname | Does the SNP position plot contain labels |
| title | Title text string |

## Value

gtable of ggplots

## Examples

```
library(snplinkage)

# example rnaseq data frame, 20 variables of 20 patients
m_rna = matrix(runif(20 ^ 2), nrow = 20)

# pair-wise correlation matrix
m_ld = cor(m_rna) ^ 2

# keep only upper triangle and reshape to data frame
m_ld[lower.tri(m_ld, diag = TRUE)] = NA
```

```
df_ld = reshape2::melt(m_ld) |> na.omit()

# rename for SNPLinkage
names(df_ld) = c('SNP_A', 'SNP_B', 'R2')

# visualize with ggplot_ld
gg_ld = ggplot_ld(df_ld)
# let's imagine the 20 variables came from 3 physically close regions
positions = c(runif(7, 10e5, 15e5), runif(6, 25e5, 30e5),
              runif(7, 45e5, 50e5)) |> sort()

# build the dataframe
df_snp_pos = data.frame(position = positions)
df_snp_pos$label = c(rep('HLA-A', 7), rep('HLA-B', 6), rep('HLA-C', 7))
gg_snp_pos = ggplot_snp_pos(df_snp_pos, labels_colname = 'label')

l_ggs = list(snp_pos = gg_snp_pos, ld = gg_ld)
gt_ld = gtable_ld_grobs(l_ggs, labels_colname = TRUE,
                        title = 'RNASeq correlations')
grid::grid.draw(gt_ld)
```

---

`is_snp_first_dim.default` *Is SNP first dimension (default)*

---

## Description

Is SNP first dimension (default)

## Usage

```
## Default S3 method:
is_snp_first_dim(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | Default object |
| `...` | Not passed |

## Value

NA

## is_snp_first_dim.gds.class *Is SNP first dimension (GDS object)*

### Description

Is SNP first dimension (GDS object)

### Usage

```
## S3 method for class 'gds.class'
is_snp_first_dim(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | GDS object |
| ... | Not passed |

### Value

Logical, TRUE if SNP is first dimension

## is_snp_first_dim.GdsGenotypeReader *Is SNP first dimension (GdsGenotypeReader object)*

### Description

Is SNP first dimension (GdsGenotypeReader object)

### Usage

```
## S3 method for class 'GdsGenotypeReader'
is_snp_first_dim(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | GdsGenotypeReader object |
| ... | Not passed |

### Value

is_snp_first_dim output on S4 slot 'handler'

---

`is_snp_first_dim.GenotypeData`
*Is SNP first dimension (GenotypeData object)*

---

## Description

Is SNP first dimension (GenotypeData object)

## Usage

```
## S3 method for class 'GenotypeData'
is_snp_first_dim(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | Genotype data object |
| `...` | Not passed |

## Value

is_snp_first_dim output on S4 slot 'data'

---

`is_snp_first_dim.MatrixGenotypeReader`
*Is SNP first dimension (MatrixGenotypeReader object)*

---

## Description

Is SNP first dimension (MatrixGenotypeReader object)

## Usage

```
## S3 method for class 'MatrixGenotypeReader'
is_snp_first_dim(obj, ...)
```

## Arguments

| | |
|---|---|
| `obj` | MatrixGenotypeReader object |
| `...` | Not passed |

## Value

TRUE

## is_snp_first_dim.NcdfGenotypeReader

*Is SNP first dimension (NcdfGenotypeReader object)*

### Description

Is SNP first dimension (NcdfGenotypeReader object)

### Usage

```
## S3 method for class 'NcdfGenotypeReader'
is_snp_first_dim(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | NcdfGenotypeReader object |
| ... | Not passed |

### Value

TRUE

## load_gds_as_genotype_data

*Load GDS as Genotype Data*

### Description

Open a connection to a snpgds file (cf. SNPRelate package) as a Genotype Data object.

### Usage

```
load_gds_as_genotype_data(
  gds_file,
  read_snp_annot = TRUE,
  read_scan_annot = TRUE
)
```

### Arguments

| | |
|---|---|
| gds_file | Path of snpgds file |
| read_snp_annot | Read the SNPs' annotations |
| read_scan_annot | Read the scans' annotations |

## Value

Genotype Data object

## Examples

```
library(snplinkage)
gds_path <- save_hgdp_as_gds()
gdata <- load_gds_as_genotype_data(gds_path)
```

---

`parallel_apply` *Separate a matrix in a list of matrices of length the number of cores and apply a function on the columns in parallel*

---

## Description

Separate a matrix in a list of matrices of length the number of cores and apply a function on the columns in parallel

## Usage

```
parallel_apply(m_data, apply_fun, n_cores = 1, ...)
```

## Arguments

| | |
|---|---|
| `m_data` | Data matrix |
| `apply_fun` | Function to apply |
| `n_cores` | Number of cores |
| `...` | Passed to apply_fun |

## Value

apply_fun return

print_qc_as_tex_table *print_qc_as_tex_table*

## Description

Print information about quality control performed by the snprelate_qc function.

## Usage

```
print_qc_as_tex_table(
  gdata_qc,
  label = "qc",
  caption = paste("Quality control and feature selection of the subset of the",
    "human genome diversity project dataset.")
)
```

## Arguments

| | |
|---|---|
| gdata_qc | Genotype Data object object returned by snprelate_qc |
| label | Label of the Tex table |
| caption | Caption of the Tex table |

## Value

Prints knitr::kable object using cat

save_hgdp_as_gds *save_hgdp_as_gds*

## Description

Save the HGDP SNP data text file as a Genomic Data Structure file

## Usage

```
save_hgdp_as_gds(paths = hgdp_filepaths(), outpath = tempfile(), ...)
```

## Arguments

| | |
|---|---|
| paths | Paths of the zip, txt, and gds files |
| outpath | Output GDS file path |
| ... | Passed to save_genotype_data_as_gds |

## Value

Path of the saved gds file

`select_region_idxs` *select_region_idxs*

## Description

Select SNP indexes corresponding to a specific genomic region.

## Usage

```
select_region_idxs(
  gdata,
  chromosome,
  position_min = -Inf,
  position_max = Inf,
  n_snps = 0,
  offset = 0
)
```

## Arguments

| | |
|---|---|
| `gdata` | Genotype Data object |
| `chromosome` | Chromosome to select |
| `position_min` | Minimum base pair position to select |
| `position_max` | Maximum base pair position to select |
| `n_snps` | Maximum number of SNPs to return |
| `offset` | Number of SNPs to offset |

## Value

SNP indexes of Genotype Data object

`snprelate_allele_frequencies`
*Compute allele frequencie and snp missing rate*

## Description

Wrapper over SNPRelate::snpgdsSNPRateFreq

### Usage

```
snprelate_allele_frequencies(
  gdata,
  snps_idx = NULL,
  scans_idx = NULL,
  quiet = FALSE
)
```

### Arguments

| | |
|---|---|
| gdata | A GenotypeData object |
| snps_idx | Vector of snps indices |
| scans_idx | Vector of scans indices |
| quiet | Whether to be quiet |

### Value

A data frame of snps_idx, snps_ids, allele1, allele2, maf, missing where allele1 and allele2 are the rates of the alleles, and maf the minimum of the 2. Missing is the missing rate. N.B: the allele rates are computed on the non missing genotypes, i.e. their sum equals 1.

---

## snprelate_ld *Wrapper for snpgdsLDMat to compute r2*

---

### Description

Wrapper for snpgdsLDMat to compute r2

### Usage

```
snprelate_ld(
  gdata,
  window_size = 0,
  min_r2 = 0,
  snps_idx = NULL,
  scans_idx = NULL,
  threads = 1,
  quiet = FALSE
)
```

### Arguments

| | |
|---|---|
| gdata | A GenotypeData object |
| window_size | Max number of SNPs in LD window, 0 for no window |
| min_r2 | Minimum r2 value to report |

| | |
|---|---|
| snps_idx | Indices of snps to use |
| scans_idx | Indices of scans to use |
| threads | The number of threads to use |
| quiet | Whether to be quiet |

## Value

A data frame with columns SNP_A, SNP_B, R2 for r2 >= min_r2

---

`snprelate_ld_select` *Wrapper for snpgdsLDpruning to select Tag SNPs*

---

## Description

The tagged snp set is (by sliding window) representative and strongly not redundant.

## Usage

```
snprelate_ld_select(
  gdata,
  window_length = 500L,
  min_r2,
  window_size = NA,
  snps_idx = NULL,
  scans_idx = NULL,
  remove.monosnp = FALSE,
  autosome.only = FALSE,
  method = "r",
  threads = 1,
  quiet = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| gdata | A GenotypeData object |
| window_length | Max length in kb of the window |
| min_r2 | Minimum r2 value to report |
| window_size | Max number of SNPs in LD window |
| snps_idx | Indices of snps to use |
| scans_idx | Indices of scans to use |
| remove.monosnp | if TRUE, remove monomorphic SNPs |
| autosome.only | if TRUE, use autosomal SNPs only; if it is a numeric or character value, keep SNPs according to the specified chromosome |

| | |
|---|---|
| method | "composite", "r", "dprime", "corr", see details |
| threads | The number of threads to use, currently ignored |
| quiet | Whether to be quiet |
| ... | Forwarded to SNPRelate::snpgdsLDpruning |

## Value

A list of SNP IDs stratified by chromosomes.

---

snprelate_qc *snprelate_qc*

---

## Description

Quality control using SNPRelate functions.

## Usage

```
snprelate_qc(
  gdata,
  samples_nas = 0.03,
  ibs = 0.99,
  keep_ids = NULL,
  snps_nas = 0.01,
  maf = 0.05,
  tagsnp = 0.8,
  n_cores = 1
)
```

## Arguments

| | |
|---|---|
| gdata | Genotype data object |
| samples_nas | NA threshold for samples, default 3 pct |
| ibs | Samples identity by state threshold, default 99 pct |
| keep_ids | Samples ids to keep even if IBS is higher than threshold. Used for monozygotic twins. |
| snps_nas | NA threshold for SNPs, default 1 pct |
| maf | Minor allele frequency threshold, default 5 pct |
| tagsnp | TagSNP r2 correlation threshold, default 0.8 |
| n_cores | Number of cores |

## Value

List of gdata, Genotype data object, and df_qc, QC info data frame

## Examples

```
library(snplinkage)
gds_path <- save_hgdp_as_gds()
gdata <- load_gds_as_genotype_data(gds_path)
qc <- snprelate_qc(gdata, tagsnp = .99)
```

---

`%<>%` *Assignment pipe*

---

## Description

Pipe an object forward into a function or call expression and update the ‘lhs‘ object with the resulting value. Magrittr imported function, see details and examples in the magrittr package.

## Arguments

| | |
|---|---|
| `lhs` | An object which serves both as the initial value and as target. |
| `rhs` | a function call using the magrittr semantics. |

## Value

None, used to update the value of lhs.

---

`%$%` *Exposition pipe*

---

## Description

Expose the names in ‘lhs‘ to the ‘rhs‘ expression. Magrittr imported function, see details and examples in the magrittr package.

## Arguments

| | |
|---|---|
| `lhs` | A list, environment, or a data.frame. |
| `rhs` | An expression where the names in lhs is available. |

## Value

Result of rhs applied to one or several names of lhs.

`%>%` *Pipe*

## Description

Pipe an object forward into a function or call expression. Magrittr imported function, see details and examples in the magrittr package.

## Arguments

| | |
|---|---|
| `lhs` | A value or the magrittr placeholder. |
| `rhs` | A function call using the magrittr semantics. |

## Value

Result of rhs applied to lhs, see details in magrittr package.

# Index