Package ‘plinkQC’

March 27, 2026

Type Package

Title Genotype Quality Control with 'PLINK'

Version 1.1.0

URL https://meyer-lab-cshl.github.io/plinkQC/

BugReports https://github.com/meyer-lab-cshl/plinkQC/issues

Maintainer Hannah Meyer <hannah.v.meyer@gmail.com>

Description Genotyping arrays enable the direct measurement of an individuals genotype at thousands of markers. 'plinkQC' facilitates genotype quality control for genetic association studies as described by Anderson and colleagues (2010) <doi:10.1038/nprot.2010.116>. It makes 'PLINK' basic statistics (e.g. missing genotyping rates per individual, allele frequencies per genetic marker) and relationship functions accessible from 'R' and generates a per-individual and per-marker quality control report. Individuals and markers that fail the quality control can subsequently be removed to generate a new, clean dataset. Removal of individuals based on relationship status is optimised to retain as many individuals as possible in the study. Additionally, there is a trained classifier to predict genomic ancestry of human samples.

Depends R (>= 3.6.0)

Imports methods, optparse, data.table (>= 1.11.0), R.utils, ggplot2, ggrepel, cowplot, UpSetR, dplyr, igraph (>= 1.2.4), sys, randomForest, stats, tidyr

Suggests testthat, mockery, formatR, knitr, rmarkdown

License MIT + file LICENSE

SystemRequirements plink (1.9)

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Hannah Meyer [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4564-0899>),
Caroline Walter [ctb],
Maha Syed [ctb]

Repository CRAN

Date/Publication 2026-03-27 15:40:02 UTC

Contents

ancestry_prediction *Predicting sample superpopulation ancestry*

Description

Predicts the ancestry of inputted samples using plink2. Projects the samples on to the principal components of the reference dataset and inputs it into a random forest classifier to identify the ancestry.

Usage

```
ancestry_prediction(
  indir,
  qcdir,
  name,
  verbose = FALSE,
  interactive = FALSE,
  path2plink2 = NULL,
  path2load_mat = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  showPlinkOutput = TRUE,
  legend_position = "right",
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  plink2format = FALSE,
  var_format = FALSE,
  rf_path = NULL,
 rf_labels = c("Africa", "America", "Central_South_Asia", "East_Asia", "Europe",
    "Middle_East"),
  excludeAncestry = NULL,
  do.run_ancestry_prediction = TRUE,
  do.evaluate_ancestry_prediction = TRUE,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK 1.9 data file name.bim, name.fam, name.bed `qcdir` [character] /path/to/directory where the plink2 data formations as returned by plink2 –make-pgen will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`name` [character] Prefix of PLINK 1.9 files, i.e. name.bim, name.fam, name.bed

`verbose` [logical] If TRUE, progress info is printed to standard out.

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_ancestry) via ggplot2::ggsave(p=p_ancestry, other_arguments) or pdf(outfile) print(p_ancestry) dev.off().

`path2plink2` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/2.0/`) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`(’plink’).

`path2load_mat` [character] /path/to/directory where loading matrices are kept. This can be downloaded from the github repo. Note that the name of the file before the .eigenvec.allele or .acount must be included in file path.

`legend_text_size`

[integer] Size for legend text.

`legend_title_size`

[integer] Size for legend title.

`axis_text_size` [integer] Size for axis text.

`axis_title_size`

[integer] Size for axis title.

`title_size` [integer] Size for plot title.

`showPlinkOutput`

[logical] If TRUE, plink log and error messages are printed to standard out.

`legend_position`

[character] Legend position for the plot.

`keep_individuals`

[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`

[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`

[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it’s okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`plink2format` [logical] If TRUE, data is in plink2 format (i.e. name.pvar, name.psam, and name.pgen)

`var_format` [logical] If TRUE, variant identifiers are in correct format already and rename_variant_identifiers will not be run

`rf_path` [character] /path/to/model for user inputted model. NULL to use the included pre-trained ancestry classification model

`rf_labels` [list] list of the label names for user inputted model.

`excludeAncestry`
[character] Ancestries to be excluded (if any). Options are: Africa, America, Central_South_Asia, East_Asia, Europe, and Middle_East. Strings must be spelled exactly as shown.

`do.run_ancestry_prediction`
[logical] If TRUE, run `run_ancestry_prediction`.

`do.evaluate_ancestry_prediction`
[logical] If TRUE, run `evaluate_ancestry_prediction`.

`write_multiqc` [logical] If TRUE, will output a multiQC-compatible report file.

Value

Three dataframes and a visualization of the ancestral probabilities. prediction_prob contains the sample IDs and ancestral probabilities from the model. prediction_majority contains the sample IDs and greatest ancestry probabilities from the model. exclude_ancestry contains the list of sample ids with ancestries to be excluded. p_ancestry contains a plot visualizing the ancestry probabilities in a bargraph.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data.hg38"
path2plink <- '/path/to/plink'
path2load_mat <- '/path/to/load_mat/merged_chrs.postQC.train.pca'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
ancestry_prediction(indir=indir, qcdir=qcdir, name=name,
path2plink2 = path2plink2, path2load_mat = path2load_mat)

## End(Not run)
```

checkFiltering *Check and construct PLINK sample and marker filters*

Description

checkFiltering checks that the file names with the individuals and markers to be filtered can be found. If so, it constructs the command for filtering

Usage

```
checkFiltering(
  keep_individuals = NULL,
  remove_individuals = NULL,
  extract_markers = NULL,
  exclude_markers = NULL
)
```

Arguments

`keep_individuals`
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

Value

Vector containing args in sys::exec_wait format to enable filtering on individuals and/or markers.

checkLoadingMat *Checking the path of the loading matrix*

Description

Makes sure that the loading matrix is located at the filepath stored in path2load_mat

Usage

```
checkLoadingMat(path2load_mat)
```

Arguments

`path2load_mat`	[character] /path/to/directory where loading matrices are kept. This can be downloaded from the github repo. Note that the name of the file before the .eigenvec.allele or .acount must be included in file path.

Examples

```
path2load_mat <- '/path/to/loading_mat/merged_chrs.postQC.train.pca'
## Not run:
# the following code is not run on package build, as the path2load_mat on the
# user system is not known.
checkLoadingMat(path2load_mat = path2load_mat)

## End(Not run)
```

checkPlink *Check PLINK software access*

Description

checkPlink checks that the PLINK software (https://www.cog-genomics.org/plink/1.9/) can be found from system call.

Usage

```
checkPlink(path2plink = NULL)
```

Arguments

`path2plink`	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

Value

Path to PLINK 1.9 executable.

`checkPlink2` *Check PLINK2 software access*

Description

checkPlink checks that the PLINK software version 2.0 (https://www.cog-genomics.org/plink/2.0/) can be found from system call.

Usage

```
checkPlink2(path2plink2 = NULL)
```

Arguments

path2plink2	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/2.0/) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

Value

Path to PLINK version 2.0 executable.

`checkRemoveIDs` *Check and construct individual IDs to be removed*

Description

checkRemoveIDs checks that the file names with the individuals to be filtered can be found. It reads the corresponding files, combines the selected individuals into one data.frame and compares these to all individuals in the analysis.

Usage

```
checkRemoveIDs(prefix, remove_individuals = NULL, keep_individuals)
```

Arguments

`prefix` [character] Prefix of PLINK files, i.e. path/2/name.bed, path/2/name.bim and path/2/name.fam.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

Value

data.frame containing family (FID) and individual (IID) IDs of individuals to be removed from analysis.

`checkRF_path` *Checking the path of userinputted random forest*

Description

Makes sure the user inputted random forest is able to be loaded

Usage

```
checkRF_path(rf_path, anc_list)
```

Arguments

`rf_path` [character] /path/to/directory where the rds file for a user inputted classification model is located Note that the name of the file including the .rds or .RDS extenstion must be included

`anc_list` [list] List of the names used for labels for the random forest

Examples

```
rf_path <- '/path/to/model/model.rds'
## Not run:
# the following code is not run on package build, as the path2load_mat on the
# user system is not known.
checkRF_path(rf_path = rf_path)

## End(Not run)
```

check_het_and_miss *Identification of individuals with outlying missing genotype or heterozygosity rates*

Description

Runs and evaluates results from plink –missing (missing genotype rates per individual) and plink –het (heterozygosity rates per individual). Non-systematic failures in genotyping and outlying heterozygosity (hz) rates per individual are often proxies for DNA sample quality. Larger than expected heterozygosity can indicate possible DNA contamination. The mean heterozygosity in PLINK is computed as hz_mean = (N-O)/N, where N: number of non-missing genotypes and O:observed number of homozygous genotypes for a given individual. Mean heterozygosity can differ between populations and SNP genotyping panels. Within a population and genotyping panel, a reduced heterozygosity rate can indicate inbreeding - these individuals will then likely be returned by `check_relatedness` as individuals that fail the relatedness filters. `check_het_and_miss` creates a scatter plot with the individuals' missingness rates on x-axis and their heterozygosity rates on the y-axis.

Usage

```
check_het_and_miss(
  indir,
  name,
  qcdir = indir,
  imissTh = 0.03,
  hetTh = 3,
  run.check_het_and_miss = TRUE,
  label_fail = TRUE,
  highlight_samples = NULL,
  highlight_type = c("text", "label", "color", "shape"),
  highlight_text_size = 3,
  highlight_color = "#c51b8a",
  highlight_shape = 17,
  highlight_legend = FALSE,
  interactive = FALSE,
  verbose = FALSE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  path2plink = NULL,
  showPlinkOutput = TRUE,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.het and name.imiss.

`qcdir` [character] /path/to/directory where name.het as returned by plink –het and name.imiss as returned by plink –missing will be saved. Per default qcdir=indir. If run.check_het_and_miss is FALSE, it is assumed that plink –missing and plink –het have been run and qcdir/name.imiss and qcdir/name.het are present. User needs writing permission to qcdir.

`imissTh` [double] Threshold for acceptable missing genotype rate per individual; has to be proportion between (0,1)

`hetTh` [double] Threshold for acceptable deviation from mean heterozygosity per individual. Expressed as multiples of standard deviation of heterozygosity (het), i.e. individuals outside mean(het) +/- hetTh*sd(het) will be returned as failing heterozygosity check; has to be larger than 0.

`run.check_het_and_miss`
[logical] Should plink –missing and plink –het be run to determine genotype missingness and heterozygosity rates; if FALSE, it is assumed that plink –missing and plink –het have been run and qcdir/name.imiss and qcdir/name.het are present; `check_het_and_miss` will fail with missing file error otherwise.

`label_fail` [logical] Set TRUE, to add fail IDs as text labels in scatter plot.

`highlight_samples`
[character vector] Vector of sample IIDs to highlight in the plot (p_het_imiss); all highlight_samples IIDs have to be present in the IIDs of the name.fam file.

`highlight_type` [character] Type of sample highlight, labeling by IID ("text"/"label") and/or highlighting data points in different "color" and/or "shape". "text" and "label" use ggrepel for minimal overlap of text labels ("text) or label boxes ("label"). Only one of "text" and "label" can be specified.Text/Label size can be specified with highlight_text_size, highlight color with highlight_color, or highlight shape with highlight_shape.

`highlight_text_size`
[integer] Text/Label size for samples specified to be highlighted (highlight_samples) by "text" or "label" (highlight_type).

`highlight_color`
[character] Color for samples specified to be highlighted (highlight_samples) by "color" (highlight_type).

`highlight_shape`
[integer] Shape for samples specified to be highlighted (highlight_samples) by "shape" (highlight_type). Possible shapes and their encoding can be found at: https://ggplot2.tidyverse.org/articles/ggplot2-specs.html#sec:shape-spec

`highlight_legend`
: [logical] Should a separate legend for the highlighted samples be provided; only relevant for highlight_type == "color" or highlight_type == "shape".

`interactive`
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_het_imiss) via ggplot2::ggsave(p=p_het_imiss , other_arguments) or pdf(outfile) print(p_het_imiss) dev.off().

`verbose`
: [logical] If TRUE, progress info is printed to standard out.

`keep_individuals`
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`title_size`
: [integer] Size for plot title.

`path2plink`
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

write_multiqc
: [logical] If TRUE, will output a multiQC-compatible report file.

Details

check_het_and_miss wraps around run_check_missingness, run_check_heterozygosity and evaluate_check_het_and_miss. If run.check_het_and_miss is TRUE, run_check_heterozygosity and run_check_missingness are executed ; otherwise it is assumed that plink –missing and plink –het have been run externally and qcdir/name.het and qcdir/name.imiss exist. check_het_and_miss will fail with missing file error otherwise.

For details on the output data.frame fail_imiss and fail_het, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#imiss and https://www.cog-genomics.org/plink/1.9/formats#het

Value

Named [list] with i) fail_imiss [data.frame] containing FID (Family ID), IID (Within-family ID), MISS_PHENO (Phenotype missing? (Y/N)), N_MISS (Number of missing genotype call(s), not including obligatory missings), N_GENO (Number of potentially valid call(s)), F_MISS (Missing call rate) of individuals failing missing genotype check and ii) fail_het [data.frame] containing FID (Family ID), IID (Within-family ID), O(HOM) (Observed number of homozygotes), E(HOM) (Expected number of homozygotes), N(NM) (Number of non-missing autosomal genotypes), F (Method-of-moments F coefficient estimate) of individuals failing outlying heterozygosity check and iii) p_het_imiss, a ggplot2-object 'containing' a scatter plot with the samples' missingness rates on x-axis and their heterozygosity rates on the y-axis, which can be shown by print(p_het_imiss).

Examples

```
 ## Not run:
indir <- system.file("extdata", package="plinkQC")
name <- "data"
path2plink <- "path/to/plink"

# whole dataset
fail_het_miss <- check_het_and_miss(indir=indir, name=name,
interactive=FALSE,path2plink=path2plink)

# subset of dataset with sample highlighting
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
fail_het_miss <- check_het_and_miss(indir=indir, name=name,
interactive=FALSE,path2plink=path2plink,
remove_individuals=remove_individuals_file,
highlight_samples=highlight_samples[,2], highlight_type = c("text", "shape"))

## End(Not run)
```

check_hwe *Identification of SNPs showing a significant deviation from Hardy-Weinberg- equilibrium (HWE)*

Description

Runs and evaluates results from plink –hardy. It calculates the observed and expected heterozygote frequencies for all variants in the individuals that passed the `perIndividualQC` and computes the deviation of the frequencies from Hardy-Weinberg equilibrium (HWE) by HWE exact test. The p-values of the HWE exact test are displayed as histograms (stratified by all and low p-values), where the hweTh is used to depict the quality control cut-off for SNPs.

Usage

```
check_hwe(
  indir,
  name,
  qcdir = indir,
  hweTh = 1e-05,
  interactive = FALSE,
  path2plink = NULL,
  verbose = FALSE,
  showPlinkOutput = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9
)
```

Arguments

`indir`	[character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.
`name`	[character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.
`qcdir`	[character] /path/to/directory where results will be written to. If `perIndividualQC` was conducted, this directory should be the same as qcdir specified in `perIndividualQC`, i.e. it contains name.fail.IDs with IIDs of individuals that failed QC. User needs writing permission to qcdir. Per default, qcdir=indir.
`hweTh`	[double] Significance threshold for deviation from HWE.

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_hwe) via ggplot2::ggsave(p=p_hwe, other_arguments) or pdf(outfile) print(p_hwe) dev.off().

`path2plink` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`verbose` [logical] If TRUE, progress info is printed to standard out and specifically, if TRUE, plink log will be displayed.

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
[integer] Size for legend text.

`legend_title_size`
[integer] Size for legend title.

`axis_text_size` [integer] Size for axis text.

`axis_title_size`
[integer] Size for axis title.

`title_size` [integer] Size for plot title.

Details

check_hwe uses plink –remove name.fail.IDs –hardy to calculate the observed and expected heterozygote frequencies per SNP in the individuals that passed the perIndividualQC. It does so without generating a new dataset but simply removes the IDs when calculating the statistics.

For details on the output data.frame fail_hwe, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#hwe.

Value

Named list with i) fail_hwe containing a [data.frame] with CHR (Chromosome code), SNP (Variant identifier), TEST (Type of test: one of: ALL’, ’AFF’, ’UNAFF’, ’ALL(QT)’, ’ALL(NP)’), A1 (Allele 1; usually minor), A2 (Allele 2; usually major), GENO (’/’-separated genotype counts: A1 hom, het, A2 hom), O(HET) (Observed heterozygote frequency E(HET) (Expected heterozygote frequency), P (Hardy-Weinberg equilibrium exact test p-value) for all SNPs that failed the hweTh and ii) p_hwe, a ggplot2-object ’containing’ the HWE p-value distribution histogram which can be shown by (print(p_hwe)).

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# run on all individuals and markers
fail_hwe <- check_hwe(indir=indir, qcdir=qcdir, name=name, interactive=FALSE,
verbose=TRUE, path2plink=path2plink)

# run on subset of individuals and markers
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
extract_markers_file <- system.file("extdata", "extract_markers",
package="plinkQC")
fail_hwe <- check_hwe(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink,
remove_individuals=remove_individuals_file,
extract_markers=extract_markers_file)

## End(Not run)
```

check_maf *Identification of SNPs with low minor allele frequency*

Description

Runs and evaluates results from plink –freq. It calculates the minor allele frequencies for all variants in the individuals that passed the perIndividualQC. The minor allele frequency distributions is plotted as a histogram.

Usage

```
check_maf(
  indir,
  name,
  qcdir = indir,
  macTh = 20,
  mafTh = NULL,
  verbose = FALSE,
  interactive = FALSE,
  path2plink = NULL,
  showPlinkOutput = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9
)
```

Arguments

indir	[character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.
name	[character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.
qcdir	[character] /path/to/directory where results will be written to. If perIndividualQC was conducted, this directory should be the same as qcdir specified in perIndividualQC, i.e. it contains name.fail.IDs with IIDs of individuals that failed QC. User needs writing permission to qcdir. Per default, qcdir=indir.
macTh	[double] Threshold for minor allele cut cut-off, if both mafTh and macTh are specified, macTh is used (macTh = mafTh*2*NrSamples).
mafTh	[double] Threshold for minor allele frequency cut-off.
verbose	[logical] If TRUE, progress info is printed to standard out and specifically, if TRUE, plink log will be displayed.
interactive	[logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_hwe) via ggplot2::ggsave(p=p_maf, other_arguments) or pdf(outfile) print(p_maf) dev.off().

`path2plink` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
[integer] Size for legend text.

`legend_title_size`
[integer] Size for legend title.

`axis_text_size` [integer] Size for axis text.

`axis_title_size`
[integer] Size for axis title.

`title_size` [integer] Size for plot title.

Details

`check_maf` uses plink –remove name.fail.IDs –freq to calculate the minor allele frequencies for all variants in the individuals that passed the perIndividualQC. It does so without generating a new dataset but simply removes the IDs when calculating the statistics.

For details on the output data.frame fail_maf, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#frq.

Value

Named list with i) fail_maf containing a [data.frame] with CHR (Chromosome code), SNP (Variant identifier), A1 (Allele 1; usually minor), A2 (Allele 2; usually major), MAF (Allele 1 frequency), NCHROBS (Number of allele observations) for all SNPs that failed the mafTh/macTh and ii) p_maf, a ggplot2-object 'containing' the MAF distribution histogram which can be shown by (print(p_maf)).

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# run on all individuals and markers
fail_maf <- check_maf(indir=indir, qcdir=qcdir, name=name, macTh=15,
interactive=FALSE, verbose=TRUE, path2plink=path2plink)

# run on subset of individuals and markers
keep_individuals_file <- system.file("extdata", "keep_individuals",
package="plinkQC")
exclude_markers_file <- system.file("extdata", "exclude_markers",
package="plinkQC")
fail_maf <- check_maf(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink,
keep_individuals=keep_individuals_file, exclude_markers=exclude_markers_file)

## End(Not run)
```

`check_relatedness` *Identification of related individuals*

Description

Runs and evaluates results from plink –genome. plink –genome calculates identity by state (IBS) for each pair of individuals based on the average proportion of alleles shared at genotyped SNPs. The degree of recent shared ancestry, i.e. the identity by descent (IBD) can be estimated from the genome-wide IBS. The proportion of IBD between two individuals is returned by plink –genome as PI_HAT. check_relatedness finds pairs of samples whose proportion of IBD is larger than the specified highIBDTh. Subsequently, for pairs of individuals that do not have additional relatives in the dataset, the individual with the greater genotype missingness rate is selected and returned as the individual failing the relatedness check. For more complex family structures, the unrelated individuals per family are selected (e.g. in a parents-offspring trio, the offspring will be marked as fail, while the parents will be kept in the analysis). `check_relatedness` depicts all pair-wise IBD-estimates as histograms stratified by value of PI_HAT.

Usage

```
check_relatedness(
  indir,
  name,
  qcdir = indir,
  highIBDTh = 0.1875,
  filter_high_ldregion = TRUE,
  high_ldregion_file = NULL,
  genomebuild = "hg19",
  imissTh = 0.03,
  run.check_relatedness = TRUE,
  interactive = FALSE,
  verbose = FALSE,
  mafThRelatedness = 0.1,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  showPlinkOutput = TRUE,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.genome and name.imiss.

`qcdir` [character] /path/to/directory to where name.genome as returned by plink –genome will be saved. Per default qcdir=indir. If run.check_relatedness is FALSE, it is assumed that plink –missing and plink –genome have been run and qcdir/name.imiss and qcdir/name.genome exist. User needs writing permission to qcdir.

`highIBDTh` [double] Threshold for acceptable proportion of IBD between pair of individuals.

`filter_high_ldregion`
[logical] Should high LD regions be filtered before IBD estimation; carried out per default with high LD regions for hg19 provided as default via `genomebuild`. For alternative genome builds not provided or non-human data, high LD regions files can be provided via `high_ldregion_file`.

`high_ldregion_file`
[character] Path to file with high LD regions used for filtering before IBD estimation if `filter_high_ldregion` == TRUE, otherwise ignored; for human

genome data, high LD region files are provided and can simply be chosen via genomebuild. Files have to be space-delimited, no column names with the following columns: chromosome, region-start, region-end, region number. Chromosomes are specified without 'chr' prefix. For instance: 1 48000000 52000000 1 2 86000000 100500000 2

genomebuild
: [character] Name of the genome build of the PLINK file annotations, ie mappings in the name.bim file. Will be used to remove high-LD regions based on the coordinates of the respective build. Options are hg18, hg19 and hg38. See @details.

imissTh
: [double] Threshold for acceptable missing genotype rate in any individual; has to be proportion between (0,1)

run.check_relatedness
: [logical] Should plink –genome be run to determine pairwise IBD of individuals; if FALSE, it is assumed that plink –genome and plink –missing have been run and qcdir/name.imiss and qcdir/name.genome are present; check_relatedness will fail with missing file error otherwise.

interactive
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_IBD() via ggplot2::ggsave(p=p_IBD, other_arguments) or pdf(outfile) print(p_IBD) dev.off().

verbose
: [logical] If TRUE, progress info is printed to standard out.

mafThRelatedness
: [double] Threshold of minor allele frequency filter for selecting variants for IBD estimation.

path2plink
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

keep_individuals
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed

from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
[integer] Size for legend text.

`legend_title_size`
[integer] Size for legend title.

`axis_text_size` [integer] Size for axis text.

`axis_title_size`
[integer] Size for axis title.

`title_size` [integer] Size for plot title.

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

`write_multiqc` [logical] If TRUE, will output a multiQC-compatible report file.

Details

`check_relatedness` wraps around `run_check_relatedness` and `evaluate_check_relatedness`. If run.check_relatedness is TRUE, `run_check_relatedness` is executed ; otherwise it is assumed that plink –genome has been run externally and qcdir/name.genome exists. `check_relatedness` will fail with missing file error otherwise.

For details on the output data.frame fail_high_IBD, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#genome.

Value

Named [list] with i) fail_high_IBD containing a [data.frame] of IIDs and FIDs of individuals who fail the IBDTh in columns FID1 and IID1. In addition, the following columns are returned (as originally obtained by plink –genome): FID2 (Family ID for second sample), IID2 (Individual ID for second sample), RT (Relationship type inferred from .fam/.ped file), EZ (IBD sharing expected value, based on just .fam/.ped relationship), Z0 (P(IBD=0)), Z1 (P(IBD=1)), Z2 (P(IBD=2)), PI_HAT (Proportion IBD, i.e. P(IBD=2) + 0.5*P(IBD=1)), PHE (Pairwise phenotypic code (1, 0, -1 = AA, AU, and UU pairs, respectively)), DST (IBS distance, i.e. (IBS2 + 0.5*IBS1) / (IBS0 + IBS1 + IBS2)), PPC (IBS binomial test), RATIO (HETHET : IBS0 SNP ratio (expected value 2)). and ii) failIDs containing a [data.frame] with individual IDs [IID] and family IDs [FID] of individuals failing the highIBDTh iii) p_IBD, a ggplot2-object 'containing' all pair-wise IBD-estimates as histograms stratified by value of PI_HAT, which can be shown by print(p_IBD).

Examples

```
## Not run:
indir <- system.file("extdata", package="plinkQC")
```

```
name <- 'data'
path2plink <- "path/to/plink"

# whole dataset
relatednessQC <- check_relatedness(indir=indir, name=name, interactive=FALSE,
run.check_relatedness=FALSE, path2plink=path2plink)

# subset of dataset
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
fail_relatedness <- check_relatedness(indir=qcdir, name=name,
remove_individuals=remove_individuals_file, path2plink=path2plink)

## End(Not run)
```

check_sex *Identification of individuals with discordant sex information*

Description

Runs and evaluates results from plink –check-sex. check_sex returns IIDs for individuals whose SNPSEX != PEDSEX (where the SNPSEX is determined by the heterozygosity rate across X-chromosomal variants). Mismatching SNPSEX and PEDSEX IDs can indicate plating errors, sample-mixup or generally samples with poor genotyping. In the latter case, these IDs are likely to fail other QC steps as well. Optionally, an extra data.frame (externalSex) with sample IDs and sex can be provided to double check if external and PEDSEX data (often processed at different centers) match. If a mismatch between PEDSEX and SNPSEX was detected, while SNPSEX == Sex, PEDSEX of these individuals can optionally be updated (fixMixup=TRUE). check_sex depicts the X-chromosomal heterozygosity (SNPSEX) of the individuals split by their (PEDSEX).

Usage

```
check_sex(
  indir,
  name,
  qcdir = indir,
  maleTh = 0.8,
  femaleTh = 0.2,
  run.check_sex = TRUE,
  externalSex = NULL,
  externalFemale = "F",
  externalMale = "M",
  externalSexSex = "Sex",
  externalSexID = "IID",
  fixMixup = FALSE,
  interactive = FALSE,
  verbose = FALSE,
  label_fail = TRUE,
  highlight_samples = NULL,
  highlight_type = c(”text”, ”label”, ”color”, ”shape”),
  highlight_text_size = 3,
  highlight_color = ”#c51b8a”,
  highlight_shape = 17,
  highlight_legend = FALSE,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  showPlinkOutput = TRUE,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam and name.sexcheck.

`qcdir` [character] /path/to/directory to save name.sexcheck as returned by plink –check-sex. Per default qcdir=indir. If run.check_sex is FALSE, it is assumed that plink –check-sex has been run and qcdir/name.sexcheck is present. User needs writing permission to qcdir.

`maleTh` [double] Threshold of X-chromosomal heterozygosity rate for males.

`femaleTh` [double] Threshold of X-chromosomal heterozygosity rate for females.

`run.check_sex` [logical] Should plink –check-sex be run? if set to FALSE, it is assumed that plink –check-sex has been run and qcdir/name.sexcheck is present; `check_sex` will fail with missing file error otherwise.

`externalSex` [data.frame, optional] Dataframe with sample IDs [externalSexID] and sex [externalSexSex] to double check if external and PEDSEX data (often processed at different centers) match.

`externalFemale` [integer/character] Identifier for ’female’ in externalSex.

`externalMale` [integer/character] Identifier for ’male’ in externalSex.

`externalSexSex` [character] Column identifier for column containing sex information in externalSex.

`externalSexID` [character] Column identifier for column containing ID information in externalSex.

`fixMixup` [logical] Should PEDSEX of individuals with mismatch between PEDSEX and Sex while Sex==SNPSEX automatically corrected: this will directly change the name.bim/.bed/.fam files!

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_sexcheck) via ggplot2::ggsave(p=p_sexcheck, other_arguments) or pdf(outfile) print(p_sexcheck) dev.off().

`verbose` [logical] If TRUE, progress info is printed to standard out.

`label_fail` [logical] Set TRUE, to add fail IDs as text labels in scatter plot.

`highlight_samples`
[character vector] Vector of sample IIDs to highlight in the plot (p_sexcheck); all highlight_samples IIDs have to be present in the IIDs of the name.fam file.

`highlight_type` [character] Type of sample highlight, labeling by IID ("text"/"label") and/or highlighting data points in different "color" and/or "shape". "text" and "label" use ggrepel for minimal overlap of text labels ("text) or label boxes ("label"). Only one of "text" and "label" can be specified. Text/Label size can be specified with highlight_text_size, highlight color with highlight_color, or highlight shape with highlight_shape.

`highlight_text_size`
[integer] Text/Label size for samples specified to be highlighted (highlight_samples) by "text" or "label" (highlight_type).

`highlight_color`
[character] Color for samples specified to be highlighted (highlight_samples) by "color" (highlight_type).

`highlight_shape`
[integer] Shape for samples specified to be highlighted (highlight_samples) by "shape" (highlight_type). Possible shapes and their encoding can be found at: `https://ggplot2.tidyverse.org/articles/ggplot2-specs.html#sec:shape-spec`

`highlight_legend`
[logical] Should a separate legend for the highlighted samples be provided; only relevant for highlight_type == "color" or highlight_type == "shape".

`path2plink` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/1.9/`) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column

and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`title_size`
: [integer] Size for plot title.

`showPlinkOutput`
: [logical] If TRUE, plink log and error messages are printed to standard out.

`write_multiqc`
: [logical] If TRUE, will output a multiQC-compatible report file

Details

`check_sex` wraps around `run_check_sex` and `evaluate_check_sex`. If run.check_sex is TRUE, `run_check_sex` is executed ; otherwise it is assumed that plink –check-sex has been run externally and qcdir/name.sexcheck exists. `check_sex` will fail with missing file error otherwise.

For details on the output data.frame fail_sex, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#sexcheck.

Value

Named list with i) fail_sex: [data.frame] with FID, IID, PEDSEX, SNPSEX and Sex (if externalSex was provided) of individuals failing sex check, ii) mixup: dataframe with FID, IID, PEDSEX, SNPSEX and Sex (if externalSex was provided) of individuals whose PEDSEX != Sex and Sex == SNPSEX and iii) p_sexcheck, a ggplot2-object 'containing' a scatter plot of the X-chromosomal heterozygosity (SNPSEX) of the sample split by their (PEDSEX), which can be shown by print(p_sexcheck).

Examples

```
 ## Not run:
indir <- system.file("extdata", package="plinkQC")
name <- "data"

# whole dataset
fail_sex <- check_sex(indir=indir, name=name, run.check_sex=FALSE,
interactive=FALSE, verbose=FALSE)

# subset of dataset with sample highlighting
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
fail_sex <- check_sex(indir=indir, name=name,
interactive=FALSE, path2plink=path2plink,
remove_individuals=remove_individuals_file,
highlight_samples=highlight_samples[,2], highlight_type = c("text", "shape"))

## End(Not run)
```

check_snp_missingness *Identification of SNPs with high missingness rate*

Description

Runs and evaluates results from plink –missing –freq. It calculate the rates of missing genotype calls and frequency for all variants in the individuals that passed the perIndividualQC. The SNP missingness rates (stratified by minor allele frequency) are depicted as histograms.

Usage

```
check_snp_missingness(
  indir,
  name,
  qcdir = indir,
  lmissTh = 0.01,
  interactive = FALSE,
  path2plink = NULL,
  verbose = FALSE,
  showPlinkOutput = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9
)
```

Arguments

indir [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

name [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

qcdir [character] /path/to/directory where results will be written to. If perIndividualQC was conducted, this directory should be the same as qcdir specified in perIndividualQC, i.e. it contains name.fail.IDs with IIDs of individuals that failed QC. User needs writing permission to qcdir. Per default, qcdir=indir.

lmissTh [double] Threshold for acceptable variant missing rate across samples.

interactive [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_lmiss) via ggplot2::ggsave(p=p_lmiss, other_arguments) or pdf(outfile) print(p_lmiss) dev.off().

path2plink [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

verbose [logical] If TRUE, progress info is printed to standard out and specifically, if TRUE, plink log will be displayed.

showPlinkOutput
[logical] If TRUE, plink log and error messages are printed to standard out.

keep_individuals
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`title_size`
: [integer] Size for plot title.

Details

`check_snp_missingness` uses plink –remove name.fail.IDs –missing –freq to calculate rates of missing genotype calls and frequency per SNP in the individuals that passed the `perIndividualQC`. It does so without generating a new dataset but simply removes the IDs when calculating the statistics.

For details on the output data.frame fail_missingness, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#lmiss.

Value

Named list with i) fail_missingness containing a [data.frame] with CHR (Chromosome code), SNP (Variant identifier), CLST (Cluster identifier. Only present with –within/–family), N_MISS (Number of missing genotype call(s), not counting obligatory missings), N_CLST (Cluster size; does not include nonmales on Ychr; Only present with –within/–family), N_GENO (Number of potentially valid call(s)), F_MISS (Missing call rate) for all SNPs failing the lmissTh and ii) p_lmiss, a ggplot2-object 'containing' the SNP missingness histogram which can be shown by (print(p_lmiss)).

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# run on all individuals and markers
fail_snp_missingness <- check_snp_missingness(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink)

# run on subset of individuals and markers
keep_individuals_file <- system.file("extdata", "keep_individuals",
package="plinkQC")
```

```
extract_markers_file <- system.file("extdata", "extract_markers",
package="plinkQC")
fail_snp_missingness <- check_snp_missingness(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink,
keep_individuals=keep_individuals_file, extract_markers=extract_markers_file)

## End(Not run)
```

cleanData *Create plink dataset with individuals and markers passing quality control*

Description

Individuals that fail per-individual QC and markers that fail per-marker QC are removed from indir/name.bim/.bed/.fam and a new, dataset with the remaining individuals and markers is created as qcdir/name.clean.bim/.bed/.fam.

Usage

```
cleanData(
  indir,
  name,
  qcdir = indir,
  filterSex = TRUE,
  filterHeterozygosity = TRUE,
  filterSampleMissingness = TRUE,
  filterRelated = TRUE,
  filterAncestry = TRUE,
  filterSNPMissingness = TRUE,
  lmissTh = 0.01,
  filterHWE = TRUE,
  hweTh = 1e-05,
  filterMAF = TRUE,
  macTh = 20,
  mafTh = NULL,
  path2plink = NULL,
  verbose = FALSE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory where results will be written to. If `perIndividualQC` was conducted, this directory should be the same as qcdir specified in `perIndividualQC`, i.e. it contains name.fail.IDs with IIDs of individuals that failed QC. User needs writing permission to qcdir. Per default, qcdir=indir.

`filterSex` [logical] Set to exclude samples that failed the sex check (via `check_sex` or `perIndividualQC`). Requires file qcdir/name.fail-sexcheck.IDs (automatically created by `perIndividualQC` if do.evaluate_check_sex set to TRUE).

`filterHeterozygosity`
[logical] Set to exclude samples that failed check for outlying heterozygosity rates (via `check_het_and_miss` or `perIndividualQC`). Requires file qcdir/name.fail-het.IDs (automatically created by `perIndividualQC` if do.evaluate_check_het_and_miss set to TRUE).

`filterSampleMissingness`
[logical] Set to exclude samples that failed check for excessive missing genotype rates (via `check_het_and_miss` or `perIndividualQC`). Requires file qcdir/name.fail-imiss.IDs (automatically created by `perIndividualQC` if do.evaluate_check_het_and_miss set to TRUE).

`filterRelated` [logical] Set to exclude samples that failed relatedness check (via `check_relatedness` or `perIndividualQC`). Requires file qcdir/name.fail-IBD.IDs (automatically created by `perIndividualQC` if do.evaluate_check_relatedness set to TRUE).

`filterAncestry` [logical] Set to exclude samples that are excluded for ancestry (via `ancestry_prediction` or `perIndividualQC`). Requires file qcdir/name.exclude-ancestry.IDs (automatically created by `perIndividualQC` if do.evaluate_check_sex set to TRUE).

`filterSNPMissingness`
[logical] Set to exclude markers that have excessive missing rates across samples (via `check_snp_missingness` or `perMarkerQC`). Requires lmissTh to be set.

`lmissTh` [double] Threshold for acceptable variant missing rate across samples.

`filterHWE` [logical] Set to exclude markers that fail HWE exact test (via `check_hwe` or `perMarkerQC`). Requires hweTh to be set.

`hweTh` [double] Significance threshold for deviation from HWE.

`filterMAF` [logical] Set to exclude markers that fail minor allele frequency or minor allele count threshold (via `check_maf` or `perMarkerQC`). Requires mafTh or macTh to be set.

`macTh` [double] Threshold for minor allele cut cut-off, if both mafTh and macTh are specified, macTh is used (macTh = mafTh*2*NrSamples).

`mafTh` [double] Threshold for minor allele frequency cut-off.

`path2plink` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/1.9/`) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

verbose [logical] If TRUE, progress info is printed to standard out.

keep_individuals
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

Value

names [list] with i) passIDs, containing a [data.frame] with family [FID] and individual [IID] IDs of samples that pass the QC, ii) failIDs, containing a [data.frame] with family [FID] and individual [IID] IDs of samples that fail the QC.

Examples

```
package.dir <- find.package('plinkQC')
indir <- file.path(package.dir, 'extdata')
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# Run qc on all samples and markers in the dataset
## Run individual QC checks
fail_individuals <- perIndividualQC(indir=indir, qcdir=qcdir, name=name,
refSamplesFile=paste(qcdir, "/HapMap_ID2Pop.txt",sep=""),
refColorsFile=paste(qcdir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE, verbose=FALSE,
path2plink=path2plink)

## Run marker QC checks
fail_markers <- perMarkerQC(indir=indir, qcdir=qcdir, name=name,
path2plink=path2plink)

## Create new dataset of individuals and markers passing QC
ids_all <- cleanData(indir=indir, qcdir=qcdir, name=name, macTh=15,
verbose=TRUE, path2plink=path2plink,
filterRelated=TRUE)

# Run qc on subset of samples and markers in the dataset
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")

fail_individuals <- perIndividualQC(indir=indir, qcdir=qcdir, name=name,
 interactive=FALSE, verbose=FALSE,
highlight_samples = highlight_samples[,2], highlight_type = "label",
remove_individuals = remove_individuals_file, path2plink=path2plink)

## Run marker QC checks
fail_markers <- perMarkerQC(indir=indir, qcdir=qcdir, name=name,
path2plink=path2plink)

## Create new dataset of individuals and markers passing QC
ids_all <- cleanData(indir=indir, qcdir=qcdir, name=name, macTh=15,
verbose=TRUE, path2plink=path2plink,
remove_individuals = remove_individuals_file)

## End(Not run)
```

convert_from_vcf *Converting VCF data files into PLINK v1.9 and PLINK v2.0 data files*

Description

This converts files in the VCF format (i.e. name.vcf) into PLINK v1.9 format (i.e. name.bed, name.bim, and name.fam) and/or PLINK v2.0 format (i.e. name.pvar, name.psam, and name.pgen)

Usage

```
convert_from_vcf(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink2 = NULL,
  gzipped = FALSE,
  makebed = TRUE,
  makepgen = FALSE,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic vcf file, name.vcf or name.vcf.gz

`name` [character] Prefix of VCF files, i.e. name.vcf or name.vcf.gz

`qcdir` [character] /path/to/directory where the plink1.9 or plink2 data formations as returned by plink2 –vcf –make-bed or plink2 –vcf –make-pgen will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink2` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/2.0/`) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`gzipped` [logical] Put as TRUE if the vcf file is gzipped (i.e. name.vcf.gz)

`makebed` [logical] If TRUE, will output PLINK v1.9 files (i.e. name.bed, name.bim, and name.fam)

`makepgen` [logical] If TRUE, will output PLINK v2.0 files (i.e. name.pvar, name.psam, and name.pgen)

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

Value

Creates PLINK v1.9 or PLINK v2.0 datafiles

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
convert_to_plink2(indir=indir, qcdir=qcdir, name=name, path2plink2 = path2plink2)

## End(Not run)
```

`convert_to_plink2` *Converting PLINK v1.9 data files into PLINK v2.0 data files*

Description

This converts files in the PLINK v1.9 format (i.e. name.bim, name.fam, and name.bed) into PLINK v2.0 format (i.e. name.pvar, name.psam, and name.pgen)

Usage

```
convert_to_plink2(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink2 = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK 1.9 data file name.bim, name.fam, name.bed

`name` [character] Prefix of PLINK 1.9 files, i.e. name.bim, name.fam, name.bed

`qcdir` [character] /path/to/directory where the plink2 data formations as returned by plink2 –make-pgen will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink2` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/2.0/`) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`(’plink’).

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`showPlinkOutput`
: [logical] If TRUE, plink log and error messages are printed to standard out.

Value

Creates plink 2.0 datafiles

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
convert_to_plink2(indir=indir, qcdir=qcdir, name=name, path2plink2 = path2plink2)

## End(Not run)
```

`evaluate_ancestry_prediction`

Predicting sample superpopulation ancestry

Description

Predicts the ancestry of inputted samples using plink2. Uses the output of `run_ancestry_prediction` as input in a random forest classifier to predict the genomic ancestry of samples within six continental groups: AFR, AMR, EAS, EUR, CSA, and MID. Genomic data version hg38 with variant identifiers in the format of 1:12345[hg38] is needed for the function to work

Usage

```
evaluate_ancestry_prediction(
  qcdir,
  name,
  verbose = FALSE,
  interactive = FALSE,
  excludeAncestry = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  showPlinkOutput = TRUE,
  legend_position = "right",
  rf_path = NULL,
 rf_labels = c("Africa", "America", "Central_South_Asia", "East_Asia", "Europe",
   "Middle_East"),
  write_multiqc = FALSE
)
```

Arguments

`qcdir` [character] /path/to/directory where name.sscore as returned by plink2 –score is located.

`name` [character] Prefix of file with a .sscore output

`verbose` [logical] If TRUE, progress info is printed to standard out.

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_ancestry) via ggplot2::ggsave(p=p_ancestry, other_arguments) or pdf(outfile) print(p_ancestry) dev.off().

`excludeAncestry` [character] Ancestries to be excluded (if any). Options are: Africa, America, Central_South_Asia, East_Asia, Europe, and Middle_East. Strings must be spelled exactly as shown.

`legend_text_size` [integer] Size for legend text.

`legend_title_size` [integer] Size for legend title.

`axis_text_size` [integer] Size for axis text.

`axis_title_size` [integer] Size for axis title.

`title_size` [integer] Size for plot title.

`showPlinkOutput` [logical] If TRUE, plink log and error messages are printed to standard out.

`legend_position`
: [character] Legend position for the plot.

`rf_path` [character] /path/to/model for user inputted model. NULL to use the included pre-trained ancestry classification model

`rf_labels` [list] list of the label names for user inputted model.

`write_multiqc` [logical] If TRUE, will output a multiQC-compatible report file.

Value

Three dataframes and a visualization of the ancestral probabilities. prediction_prob contains the sample IDs and ancestral probabilities from the model. prediction_majority contains the sample IDs and greatest ancestry probabilities from the model. exclude_ancestry contains the list of sample ids with ancestries to be excluded. p_ancestry contains a plot visualizing the ancestry probabilities in a bargraph.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data.hg38"
path2plink <- '/path/to/plink'
path2load_mat <- '/path/to/load_mat/merged_chrs.postQC.train.pca'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
superpop_classification(indir=indir, qcdir=qcdir, name=name,
path2plink2 = path2plink2, path2load_mat = path2load_mat)

## End(Not run)
```

---

evaluate_check_het_and_miss

Evaluate results from PLINK missing genotype and heterozygosity rate check.

Description

Evaluates and depicts results from plink –missing (missing genotype rates per individual) and plink –het (heterozygosity rates per individual) via `run_check_heterozygosity` and `run_check_missingness` or externally conducted check.) Non-systematic failures in genotyping and outlying heterozygosity (hz) rates per individual are often proxies for DNA sample quality. Larger than expected heterozygosity can indicate possible DNA contamination. The mean heterozygosity in PLINK is computed as hz_mean = (N-O)/N, where N: number of non-missing genotypes and O:observed number of homozygous genotypes for a given individual. Mean heterozygosity can differ between populations and SNP genotyping panels. Within a population and genotyping panel, a reduced heterozygosity rate can indicate inbreeding - these individuals will then be returned by check_relatedness as individuals that fail the relatedness filters. `evaluate_check_het_and_miss` creates a scatter plot with the individuals' missingness rates on x-axis and their heterozygosity rates on the y-axis.

Usage

```
evaluate_check_het_and_miss(
  qcdir,
  name,
  imissTh = 0.03,
  hetTh = 3,
  label_fail = TRUE,
  highlight_samples = NULL,
  highlight_type = c("text", "label", "color", "shape"),
  highlight_text_size = 3,
  highlight_color = "#c51b8a",
  highlight_shape = 17,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  highlight_legend = FALSE,
  interactive = FALSE,
  write_multiqc = FALSE
)
```

Arguments

`qcdir` [character] path/to/directory/with/QC/results containing name.imiss and name.het results as returned by plink –missing and plink –het.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.het and name.imiss.

`imissTh` [double] Threshold for acceptable missing genotype rate in any individual; has to be proportion between (0,1)

`hetTh` [double] Threshold for acceptable deviation from mean heterozygosity in any individual. Expressed as multiples of standard deviation of heterozygosity (het), i.e. individuals outside mean(het) +/- hetTh*sd(het) will be returned as failing heterozygosity check; has to be larger than 0.

`label_fail` [logical] Set TRUE, to add fail IDs as text labels in scatter plot.

`highlight_samples`
[character vector] Vector of sample IIDs to highlight in the plot (p_het_imiss); all highlight_samples IIDs have to be present in the IIDs of the name.fam file.

`highlight_type` [character] Type of sample highlight, labeling by IID ("text"/"label") and/or highlighting data points in different "color" and/or "shape". "text" and "label" use ggrepel for minimal overlap of text labels ("text) or label boxes ("label"). Only one of "text" and "label" can be specified.Text/Label size can be specified with highlight_text_size, highlight color with highlight_color, or highlight shape with highlight_shape.

`highlight_text_size`
[integer] Text/Label size for samples specified to be highlighted (highlight_samples) by "text" or "label" (highlight_type).

`highlight_color`
: [character] Color for samples specified to be highlighted (highlight_samples) by "color" (highlight_type).

`highlight_shape`
: [integer] Shape for samples specified to be highlighted (highlight_samples) by "shape" (highlight_type). Possible shapes and their encoding can be found at: `https://ggplot2.tidyverse.org/articles/ggplot2-specs.html#sec:shape-spec`

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`title_size`
: [integer] Size for plot title.

`highlight_legend`
: [logical] Should a separate legend for the highlighted samples be provided; only relevant for highlight_type == "color" or highlight_type == "shape".

`interactive`
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_het_imiss) via ggplot2::ggsave(p=p_het_imiss , other_arguments) or pdf(outfile) print(p_het_imiss) dev.off().

`write_multiqc`
: [logical] If TRUE, will output a multiQC-compatible report file

Details

All, `run_check_heterozygosity`, `run_check_missingness` and `evaluate_check_het_and_miss` can simply be invoked by `check_het_and_miss`.

For details on the output data.frame fail_imiss and fail_het, check the original description on the PLINK output format page: `https://www.cog-genomics.org/plink/1.9/formats#imiss` and `https://www.cog-genomics.org/plink/1.9/formats#het`

Value

named [list] with i) fail_imiss dataframe containing FID (Family ID), IID (Within-family ID), MISS_PHENO (Phenotype missing? (Y/N)), N_MISS (Number of missing genotype call(s), not including obligatory missings), N_GENO (Number of potentially valid call(s)), F_MISS (Missing call rate) of individuals failing missing genotype check and ii) fail_het dataframe containing FID (Family ID), IID (Within-family ID), O(HOM) (Observed number of homozygotes), E(HOM) (Expected number of homozygotes), N(NM) (Number of non-missing autosomal genotypes), F (Method-of-moments F coefficient estimate) of individuals failing outlying heterozygosity check; iii) p_het_imiss, a ggplot2-object 'containing' a scatter plot with the samples' missingness rates on x-axis and their heterozygosity rates on the y-axis, which can be shown by print(p_het_imiss) and iv) plot_data, a data.frame with the data visualised in p_het_imiss (iii).

Examples

```
qcdir <- system.file("extdata", package="plinkQC")
name <- "data"
## Not run:
fail_het_miss <- evaluate_check_het_and_miss(qcdir=qcdir, name=name,
interactive=FALSE)

#' # highlight samples
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
fail_het_miss <- evaluate_check_het_and_miss(qcdir=qcdir, name=name,
interactive=FALSE, highlight_samples = highlight_samples[,2],
highlight_type = c("text", "color"))

## End(Not run)
```

---

evaluate_check_relatedness

Evaluate results from PLINK IBD estimation.

Description

Evaluates and depicts results from plink –genome on the LD pruned dataset (via run_check_relatedness or externally conducted IBD estimation). plink –genome calculates identity by state (IBS) for each pair of individuals based on the average proportion of alleles shared at genotyped SNPs. The degree of recent shared ancestry, i.e. the identity by descent (IBD) can be estimated from the genome-wide IBS. The proportion of IBD between two individuals is returned by –genome as PI_HAT. evaluate_check_relatedness finds pairs of samples whose proportion of IBD is larger than the specified highIBDTh. Subsequently, for pairs of individual that do not have additional relatives in the dataset, the individual with the greater genotype missingness rate is selected and returned as the individual failing the relatedness check. For more complex family structures, the unrelated individuals per family are selected (e.g. in a parents-offspring trio, the offspring will be marked as fail, while the parents will be kept in the analysis). evaluate_check_relatedness depicts all pair-wise IBD-estimates as histograms stratified by value of PI_HAT.

Usage

```
evaluate_check_relatedness(
  qcdir,
  name,
  highIBDTh = 0.1875,
  imissTh = 0.03,
  interactive = FALSE,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  verbose = FALSE,
  write_multiqc = FALSE
)
```

Arguments

qcdir [character] path/to/directory/with/QC/results containing name.imiss and name.genome results as returned by plink –missing and plink –genome.

name [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.genome and name.imiss.

highIBDTh [double] Threshold for acceptable proportion of IBD between pair of individuals.

imissTh [double] Threshold for acceptable missing genotype rate in any individual; has to be proportion between (0,1)

interactive [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_IBD() via ggplot2::ggsave(p=p_IBD, other_arguments) or pdf(outfile) print(p_IBD) dev.off().

legend_text_size [integer] Size for legend text.

legend_title_size [integer] Size for legend title.

axis_text_size [integer] Size for axis text.

axis_title_size [integer] Size for axis title.

title_size [integer] Size for plot title.

verbose [logical] If TRUE, progress info is printed to standard out.

write_multiqc [logical] If TRUE, will output a multiQC-compatible report file.

Details

Both run_check_relatedness and evaluate_check_relatedness can simply be invoked by check_relatedness.

For details on the output data.frame fail_high_IBD, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#genome.

Value

a named [list] with i) fail_high_IBD containing a [data.frame] of IIDs and FIDs of individuals who fail the IBDTh in columns FID1 and IID1. In addition, the following columns are returned (as originally obtained by plink –genome): FID2 (Family ID for second sample), IID2 (Individual ID for second sample), RT (Relationship type inferred from .fam/.ped file), EZ (IBD sharing expected value, based on just .fam/.ped relationship), Z0 (P(IBD=0)), Z1 (P(IBD=1)), Z2 (P(IBD=2)), PI_HAT (Proportion IBD, i.e. P(IBD=2) + 0.5*P(IBD=1)), PHE (Pairwise phenotypic code (1, 0, -1 = AA, AU, and UU pairs, respectively)), DST (IBS distance, i.e. (IBS2 + 0.5*IBS1) / (IBS0 +

IBS1 + IBS2)), PPC (IBS binomial test), RATIO (HETHET : IBS0 SNP ratio (expected value 2)). and ii) failIDs containing a [data.frame] with individual IDs [IID] and family IDs [FID] of individuals failing the highIBDTh; iii) p_IBD, a ggplot2-object 'containing' all pair-wise IBD-estimates as histograms stratified by value of PI_HAT, which can be shown by print(p_IBD and iv) plot_data, a data.frame with the data visualised in p_IBD (iii).

Examples

```
qcdir <- system.file("extdata", package="plinkQC")
name <- 'data'
## Not run:
relatednessQC <- evaluate_check_relatedness(qcdir=qcdir, name=name,
interactive=FALSE)

## End(Not run)
```

evaluate_check_sex *Evaluate results from PLINK sex check.*

Description

Evaluates and depicts results from plink –check-sex (via run_check_sex or externally conducted sex check). Takes file qcdir/name.sexcheck and returns IIDs for samples whose SNPSEX != PEDSEX (where the SNPSEX is determined by the heterozygosity rate across X-chromosomal variants). Mismatching SNPSEX and PEDSEX IDs can indicate plating errors, sample-mixup or generally samples with poor genotyping. In the latter case, these IDs are likely to fail other QC steps as well. Optionally, an extra data.frame (externalSex) with sample IDs and sex can be provided to double check if external and PEDSEX data (often processed at different centers) match. If a mismatch between PEDSEX and SNPSEX was detected while SNPSEX == Sex, PEDSEX of these individuals can optionally be updated (fixMixup=TRUE). evaluate_check_sex depicts the X-chromosomal heterozygosity (SNPSEX) of the samples split by their (PEDSEX).

Usage

```
evaluate_check_sex(
  qcdir,
  name,
  maleTh = 0.8,
  femaleTh = 0.2,
  externalSex = NULL,
  fixMixup = FALSE,
  indir = qcdir,
  externalFemale = "F",
  externalMale = "M",
  externalSexSex = "Sex",
  externalSexID = "IID",
  verbose = FALSE,
  label_fail = TRUE,
  highlight_samples = NULL,
  highlight_type = c("text", "label", "color", "shape"),
  highlight_text_size = 3,
  highlight_color = "#c51b8a",
  highlight_shape = 17,
  highlight_legend = FALSE,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE,
  interactive = FALSE,
  write_multiqc = FALSE
)
```

Arguments

`qcdir` [character] /path/to/directory containing name.sexcheck as returned by plink –check-sex.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam and name.sexcheck.

`maleTh` [double] Threshold of X-chromosomal heterozygosity rate for males.

`femaleTh` [double] Threshold of X-chromosomal heterozygosity rate for females.

`externalSex` [data.frame, optional] with sample IDs [externalSexID] and sex [externalSexSex] to double check if external and PEDSEX data (often processed at different centers) match.

`fixMixup` [logical] Should PEDSEX of individuals with mismatch between PEDSEX and Sex, with Sex==SNPSEX automatically corrected: this will directly change the name.bim/.bed/.fam files!

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files; only required of fixMixup==TRUE. User needs writing permission to indir.

`externalFemale` [integer/character] Identifier for 'female' in externalSex.

`externalMale` [integer/character] Identifier for 'male' in externalSex.

`externalSexSex` [character] Column identifier for column containing sex information in externalSex.

`externalSexID` [character] Column identifier for column containing ID information in externalSex.

`verbose` [logical] If TRUE, progress info is printed to standard out.

label_fail [logical] Set TRUE, to add fail IDs as text labels in scatter plot.

highlight_samples

[character vector] Vector of sample IIDs to highlight in the plot (p_sexcheck); all highlight_samples IIDs have to be present in the IIDs of the name.fam file.

highlight_type [character] Type of sample highlight, labeling by IID ("text"/"label") and/or highlighting data points in different "color" and/or "shape". "text" and "label" use ggrepel for minimal overlap of text labels ("text) or label boxes ("label"). Only one of "text" and "label" can be specified. Text/Label size can be specified with highlight_text_size, highlight color with highlight_color, or highlight shape with highlight_shape.

highlight_text_size

[integer] Text/Label size for samples specified to be highlighted (highlight_samples) by "text" or "label" (highlight_type).

highlight_color

[character] Color for samples specified to be highlighted (highlight_samples) by "color" (highlight_type).

highlight_shape

[integer] Shape for samples specified to be highlighted (highlight_samples) by "shape" (highlight_type). Possible shapes and their encoding can be found at: https://ggplot2.tidyverse.org/articles/ggplot2-specs.html#sec:shape-spec

highlight_legend

[logical] Should a separate legend for the highlighted samples be provided; only relevant for highlight_type == "color" or highlight_type == "shape".

legend_text_size

[integer] Size for legend text.

legend_title_size

[integer] Size for legend title.

axis_text_size [integer] Size for axis text.

axis_title_size

[integer] Size for axis title.

title_size [integer] Size for plot title.

path2plink [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

keep_individuals

[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals

[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column

and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_sexcheck) via ggplot2::ggsave(p=p_sexcheck, other_arguments) or pdf(outfile) print(p_sexcheck) dev.off().

`write_multiqc` [logical] If TRUE, will output a multiQC-compatible report file.

Details

Both `run_check_sex` and `evaluate_check_sex` can simply be invoked by `check_sex`.

For details on the output data.frame fail_sex, check the original description on the PLINK output format page: https://www.cog-genomics.org/plink/1.9/formats#sexcheck.

Value

named list with i) fail_sex: dataframe with FID, IID, PEDSEX, SNPSEX and Sex (if externalSex was provided) of individuals failing sex check; ii) mixup: dataframe with FID, IID, PEDSEX, SNPSEX and Sex (if externalSex was provided) of individuals whose PEDSEX != Sex and Sex == SNPSEX; iii) p_sexcheck, a ggplot2-object 'containing' a scatter plot of the X-chromosomal heterozygosity (SNPSEX) of the individuals split by their (PEDSEX), which can be shown by print(p_sexcheck) and iv) plot_data, a data.frame with the data visualised in p_sexcheck (iii).

Examples

```
qcdir <- system.file("extdata", package="plinkQC")
name <- "data"
path2plink <- '/path/to/plink'
## Not run:
fail_sex <- evaluate_check_sex(qcdir=qcdir, name=name, interactive=FALSE,
verbose=FALSE, path2plink=path2plink)
```

```
# highlight samples
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
fail_sex <- evaluate_check_sex(qcdir=qcdir, name=name, interactive=FALSE,
verbose=FALSE, path2plink=path2plink,
highlight_samples = highlight_samples[,2],
highlight_type = c("label", "color"), highlight_color = "darkgreen")

## End(Not run)
```

---

overviewPerIndividualQC

Overview of per sample QC

Description

overviewPerIndividualQC depicts results of perIndividualQC as intersection plots (via upset) and returns dataframes indicating which QC checks individuals failed or passed.

Usage

```
overviewPerIndividualQC(results_perIndividualQC, interactive = FALSE)
```

Arguments

results_perIndividualQC
: [list] Output of perIndividualQC i.e. named [list] with i) sample_missingness containing a [vector] with sample IIDs failing the selected missingness threshold imissTh, ii) highIBD containing a [vector] with sample IIDs failing the selected relatedness threshold highIBDTh, iii) outlying_heterozygosity containing a [vector] with sample IIDs failing selected the heterozygosity threshold hetTh, iv) mismatched_sex containing a [vector] with the sample IIDs failing the sex-check based on SNPSEX and selected femaleTh/maleTh, and v) p_sampleQC, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots of check_sex, check_het_and_miss, and check_relatedness.

interactive
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_overview) via ggplot2::ggsave(p=p_overview, other_arguments) or pdf(outfile) print(p_overview) dev.off().

Value

Named [list] with i) nr_fail_samples: total number of samples [integer] failing perIndividualQC, ii) fail_QC containing a [data.frame] with samples that failed QC steps (excluding ancestry) with IID, FID, all QC steps applied by perIndividualQC (max=4), with entries=0 if passing the QC and entries=1 if failing that particular QC and iii) fail_QC_and_ancestry_exclusion containing a [data.frame] with samples that are excluded for ancestry and QC checks with IID, FID, QC_fail and

Ancestry_exclusion, with entries=0 if passing and entries=1 if failing that check, iii) p_overview, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
## Not run:
fail_individuals <- perIndividualQC(qcdir=qcdir, indir=indir, name=name,
refSamplesFile=paste(qcdir, "/HapMap_ID2Pop.txt",sep=""),
refColorsFile=paste(qcdir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE, verbose=FALSE,
do.run_check_het_and_miss=FALSE, do.run_check_relatedness=FALSE,
do.run_check_sex=FALSE)

overview <- overviewPerIndividualQC(fail_individuals)

## End(Not run)
```

---

overviewPerMarkerQC *Overview of per marker QC*

Description

overviewPerMarkerQC depicts results of perMarkerQC as an intersection plot (via upset) and returns a dataframe indicating which QC checks were failed or passed.

Usage

```
overviewPerMarkerQC(results_perMarkerQC, interactive = FALSE)
```

Arguments

results_perMarkerQC
: [list] Output of perIndividualQC i.e. named [list] with i) fail_list, a named [list] with 1. SNP_missingness, containing SNP IDs failing the missingness threshold lmissTh, 2. hwe, containing SNP IDs failing the HWE exact test threshold hweTh and 3. maf, containing SNPs failing the MAF threshold mafTh/MAC threshold macTh and ii) p_markerQC, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots of check_snp_missingness, check_hwe and check_maf

interactive
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_overview) via ggplot2::ggsave(p=p_overview, other_arguments) or pdf(outfile) print(p_overview) dev.off().

Value

Named [list] with i) nr_fail_markers: total number of markers [integer] failing perMarkerQC, ii) fail_QC containing a [data.frame] with markers that failed QC steps: marker rsIDs in rows, columns are all QC steps applied by perMarkerQC (max=3), with entries=0 if passing the QC and entries=1 if failing that particular QC.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
# All quality control checks
## Not run:
fail_markers <- perMarkerQC(qcdir=qcdir, indir=indir, name=name,
interactive=FALSE, verbose=TRUE, path2plink=path2plink)
overview <- overviewPerMarkerQC(fail_markers)

## End(Not run)
```

`perIndividualQC` *Quality control for all individuals in plink-dataset*

Description

perIndividualQC checks the samples in the plink dataset for their total missingness and heterozygosity rates, the concordance of their assigned sex to their SNP sex, their relatedness to other study individuals and their genetic ancestry.

Usage

```
perIndividualQC(
  indir,
  name,
  qcdir = indir,
  dont.check_sex = FALSE,
  do.run_check_sex = TRUE,
  do.evaluate_check_sex = TRUE,
  maleTh = 0.8,
  femaleTh = 0.2,
  externalSex = NULL,
  externalMale = "M",
  externalSexSex = "Sex",
  externalSexID = "IID",
  externalFemale = "F",
  fixMixup = FALSE,
  dont.check_het_and_miss = FALSE,
  do.run_check_het_and_miss = TRUE,
  do.evaluate_check_het_and_miss = TRUE,
  imissTh = 0.03,
  hetTh = 3,
  dont.check_relatedness = FALSE,
  do.run_check_relatedness = TRUE,
  do.evaluate_check_relatedness = TRUE,
  highIBDTh = 0.1875,
  mafThRelatedness = 0.1,
  filter_high_ldregion = TRUE,
  high_ldregion_file = NULL,
  genomebuild = "hg38",
  label_fail = TRUE,
  highlight_samples = NULL,
  highlight_type = c("text", "label", "color", "shape"),
  highlight_text_size = 3,
  highlight_color = "#c51b8a",
  highlight_shape = 17,
  highlight_legend = FALSE,
  interactive = FALSE,
  verbose = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  subplot_label_size = 9,
  title_size = 9,
  path2plink = NULL,
  showPlinkOutput = TRUE,
  path2plink2 = NULL,
  dont.ancestry_prediction = FALSE,
  do.run_ancestry_prediction = TRUE,
  do.evaluate_ancestry_prediction = TRUE,
  excludeAncestry = NULL,
  path2load_mat = NULL,
  plink2format = FALSE,
  var_format = FALSE,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory where results will be saved. Per default, qcdir=indir. If do.evaluate_[analysis] is set to TRUE and do.run_[analysis] is FALSE, `perIndividualQC` expects the analysis-specific plink output files in qcdir i.e. do.check_sex expects name.sexcheck, do.evaluate_check_het_and_miss expects name.het and name.imiss, do.evaluate_check_relatedness expects name.genome and name.imiss and Setting do.run_[analysis] TRUE will execute the checks and create the required files. User needs writing permission to qcdir.

`dont.check_sex` [logical] If TRUE, no sex check will be conducted; short for do.run_check_sex=FALSE and do.evaluate_check_sex=FALSE. Takes precedence over do.run_check_sex and do.evaluate_check_sex.

`do.run_check_sex`
[logical] If TRUE, run `run_check_sex`

`do.evaluate_check_sex`
[logical] If TRUE, run `evaluate_check_sex`

`maleTh` [double] Threshold of X-chromosomal heterozygosity rate for males.

`femaleTh` [double] Threshold of X-chromosomal heterozygosity rate for females.

`externalSex` [data.frame, optional] Dataframe with sample IDs [externalSexID] and sex [externalSexSex] to double check if external and PEDSEX data (often processed at different centers) match.

`externalMale` [integer/character] Identifier for 'male' in externalSex.

`externalSexSex` [character] Column identifier for column containing sex information in externalSex.

`externalSexID` [character] Column identifier for column containing ID information in externalSex.

`externalFemale` [integer/character] Identifier for 'female' in externalSex.

`fixMixup` [logical] Should PEDSEX of individuals with mismatch between PEDSEX and Sex while Sex==SNPSEX automatically corrected: this will directly change the name.bim/.bed/.fam files!

`dont.check_het_and_miss`
[logical] If TRUE, no heterozygosity and missingness check will be conducted; short for do.run_check_heterozygosity=FALSE, do.run_check_missingness=FALSE and do.evaluate_check_het_and_miss=FALSE. Takes precedence over do.run_check_heterozygosity, do.run_check_missingness and do.evaluate_check_het_and_miss.

`do.run_check_het_and_miss`
[logical] If TRUE, run `run_check_heterozygosity` and `run_check_missingness`

`do.evaluate_check_het_and_miss`
[logical] If TRUE, run `evaluate_check_het_and_miss`.

`imissTh` [double] Threshold for acceptable missing genotype rate in any individual; has to be proportion between (0,1)

`hetTh` [double] Threshold for acceptable deviation from mean heterozygosity per individual. Expressed as multiples of standard deviation of heterozygosity (het), i.e. individuals outside mean(het) +/- hetTh*sd(het) will be returned as failing heterozygosity check; has to be larger than 0.

`dont.check_relatedness`
[logical] If TRUE, no relatedness check will be conducted; short for do.run_check_relatedness=FALSE and do.evaluate_check_relatedness=FALSE. Takes precedence over do.run_check_relatedness and do.evaluate_check_relatedness.

`do.run_check_relatedness`
[logical] If TRUE, run `run_check_relatedness`.

`do.evaluate_check_relatedness`
[logical] If TRUE, run `evaluate_check_relatedness`.

`highIBDTh` [double] Threshold for acceptable proportion of IBD between pair of individuals.

`mafThRelatedness`
[double] Threshold of minor allele frequency filter for selecting variants for IBD estimation.

`filter_high_ldregion`
[logical] Should high LD regions be filtered before IBD estimation; carried out per default with high LD regions for hg19 provided as default via `genomebuild`. For alternative genome builds not provided or non-human data, high LD regions files can be provided via `high_ldregion_file`.

`high_ldregion_file`
[character] Path to file with high LD regions used for filtering before IBD estimation if `filter_high_ldregion` == TRUE, otherwise ignored; for human genome data, high LD region files are provided and can simply be chosen via `genomebuild`. Files have to be space-delimited, no column names with the following columns: chromosome, region-start, region-end, region number. Chromosomes are specified without 'chr' prefix. For instance: 1 48000000 52000000 1 2 86000000 100500000 2

`genomebuild` [character] Name of the genome build of the PLINK file annotations, ie mappings in the name.bim file. Will be used to remove high-LD regions based on the coordinates of the respective build. Options are hg18, hg19 and hg38. See @details.

`label_fail` [logical] Set TRUE, to add fail IDs as text labels in scatter plot.

`highlight_samples`
[character vector] Vector of sample IIDs to highlight in the plot (p_sexcheck); all highlight_samples IIDs have to be present in the IIDs of the name.fam file.

`highlight_type` [character] Type of sample highlight, labeling by IID ("text"/"label") and/or highlighting data points in different "color" and/or "shape". "text" and "label" use ggrepel for minimal overlap of text labels ("text) or label boxes ("label"). Only one of "text" and "label" can be specified. Text/Label size can be specified with highlight_text_size, highlight color with highlight_color, or highlight shape with highlight_shape.

`highlight_text_size`
[integer] Text/Label size for samples specified to be highlighted (highlight_samples) by "text" or "label" (highlight_type).

`highlight_color`
[character] Color for samples specified to be highlighted (highlight_samples) by "color" (highlight_type).

`highlight_shape`
: [integer] Shape for samples specified to be highlighted (highlight_samples) by "shape" (highlight_type). Possible shapes and their encoding can be found at: https://ggplot2.tidyverse.org/articles/ggplot2-specs.html#sec:shape-spec

`highlight_legend`
: [logical] Should a separate legend for the highlighted samples be provided; only relevant for highlight_type == "color" or highlight_type == "shape".

`interactive`
: [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_sampleQC) via ggplot2::ggsave(p=p_sampleQC, other_arguments) or pdf(outfile) print(p_sampleQC) dev.off(). If TRUE, i) depicts the X-chromosomal heterozygosity (SNPSEX) of the samples split by their PEDSEX (if do.evaluate_check_sex is TRUE), ii) creates a scatter plot with samples' missingness rates on x-axis and their heterozygosity rates on the y-axis (if do.evaluate_check_het_and_miss is TRUE), and iii) depicts all pair-wise IBD-estimates as histogram (if do.evaluate_check_relatedness is TRUE) .

`verbose`
: [logical] If TRUE, progress info is printed to standard out.

`keep_individuals`
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`subplot_label_size`
: [integer] Size of the subplot labeling.

`title_size`
: [integer] Size for plot title.

`path2plink`
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`showPlinkOutput`
: [logical] If TRUE, plink log and error messages are printed to standard out.

`path2plink2`
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/2.0/) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`dont.ancestry_prediction`
: [logical] If TRUE, no ancestry prediction will be conducted; short for do.run_ancestry_prediction=FALSE and do.evaluate_ancestry_prediction=FALSE. Takes precedence over do.run_ancestry_prediction and do.evaluate_ancestry_prediction

`do.run_ancestry_prediction`
: [logical] If TRUE, run `run_ancestry_prediction`.

`do.evaluate_ancestry_prediction`
: [logical] If TRUE, run `evaluate_ancestry_prediction`.

`excludeAncestry`
: [character] Ancestries to be excluded (if any). Options are: Africa, America, Central_South_Asia, East_Asia, Europe, and Middle_East. Strings must be spelled exactly as shown.

`path2load_mat`
: [character] /path/to/directory where loading matrices are kept. This can be downloaded from the github repo. Note that the name of the file before the .eigenvec.allele or .acount must be included in file path.

`plink2format`
: [logical] If TRUE, data is also avaliable in plink2 format (i.e. name.pvar, name.psam, and name.pgen)

`var_format`
: [logical] If TRUE, variant identifiers are in correct format already and rename_variant_identifiers will not be run

`write_multiqc`
: [logical] If TRUE, will output a multiQC-compatible report file

Details

perIndividualQC wraps around the individual QC functions `check_sex`, `check_het_and_miss`, and `check_relatedness`. For details on the parameters and outputs, check these function documentations. For detailed output for fail IIDs (instead of simple IID lists), run each function individually.

Value

Named [list] with i) fail_list, a named [list] with 1. sample_missingness containing a [vector] with sample IIDs failing the missingness threshold imissTh, 2. highIBD containing a [vector] with sample IIDs failing the relatedness threshold highIBDTh, 3. outlying_heterozygosity containing a [vector] with sample IIDs failing the heterozygosity threshold hetTh, 4. mismatched_sex containing a [vector] with the sample IIDs failing the sexcheck based on SNPSEX and femaleTh/maleTh and 5. ancestry containing a dataframe of sample ids and ancestry probablities predicted by a classifier ii) p_sampleQC, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots of `check_sex`, `check_het_and_miss`, and `check_relatedness`, which can be shown by print(p_sampleQC). List entries contain NULL if that specific check was not chosen.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"

# All quality control checks
## Not run:
# whole dataset
fail_individuals <- perIndividualQC(indir=indir, qcdir=qcdir, name=name,
refSamplesFile=paste(qcdir, "/HapMap_ID2Pop.txt",sep=""),
refColorsFile=paste(qcdir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE, verbose=FALSE,
do.run_check_het_and_miss=FALSE, do.run_check_relatedness=FALSE,
do.run_check_sex=FALSE)

# Only check sex and missingness/heterozygosity
fail_sex_het_miss <- perIndividualQC(indir=indir, qcdir=qcdir, name=name,
dont.check_relatedness=TRUE,
interactive=FALSE, verbose=FALSE)

# subset of dataset with sample highlighting
highlight_samples <- read.table(system.file("extdata", "keep_individuals",
package="plinkQC"))
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
individual_qc <- perIndividualQC(indir=indir, qcdir=qcdir, name=name,
refSamplesFile=paste(qcdir, "/HapMap_ID2Pop.txt",sep=""),
refColorsFile=paste(qcdir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE, verbose=FALSE,
path2plink=path2plink,
remove_individuals=remove_individuals_file,
highlight_samples=highlight_samples[,2],
highlight_type = c("text", "color"), highlight_color="goldenrod")

## End(Not run)
```

perMarkerQC *Quality control for all markers in plink-dataset*

Description

perMarkerQC checks the markers in the plink dataset for their missingness rates across samples, their deviation from Hardy-Weinberg-Equilibrium (HWE) and their minor allele frequencies (MAF). Per default, it assumes that IDs of individuals that have failed perIndividualQC have been written to qcdir/name.fail.IDs and removes these individuals when computing missingness rates, HWE p-values and MAF. If the qcdir/name.fail.IDs file does not exist, a message is written to stdout but the analyses will continue for all samples in the name.fam/name.bed/name.bim dataset. Depicts i) SNP missingness rates (stratified by minor allele frequency) as histograms, ii) p-values of HWE exact test (stratified by all and low p-values) as histograms and iii) the minor allele frequency distribution as a histogram.

Usage

```
perMarkerQC(
  indir,
  qcdir = indir,
  name,
  do.check_snp_missingness = TRUE,
  lmissTh = 0.01,
  do.check_hwe = TRUE,
  hweTh = 1e-05,
  do.check_maf = TRUE,
  macTh = 20,
  mafTh = NULL,
  interactive = FALSE,
  verbose = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  legend_text_size = 5,
  legend_title_size = 7,
  axis_text_size = 5,
  axis_title_size = 7,
  title_size = 9,
  subplot_label_size = 9,
  path2plink = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`qcdir` [character] /path/to/directory where results will be written to. If `perIndividualQC` was conducted, this directory should be the same as qcdir specified in `perIndividualQC`, i.e. it contains name.fail.IDs with IIDs of individuals that failed QC. User needs writing permission to qcdir. Per default, qcdir=indir.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`do.check_snp_missingness`
[logical] If TRUE, run `check_snp_missingness`.

`lmissTh` [double] Threshold for acceptable variant missing rate across samples.

`do.check_hwe` [logical] If TRUE, run `check_hwe`.

`hweTh` [double] Significance threshold for deviation from HWE.

`do.check_maf` [logical] If TRUE, run `check_maf`.

`macTh` [double] Threshold for minor allele cut cut-off, if both mafTh and macTh are specified, macTh is used (macTh = mafTh*2*NrSamples).

`mafTh` [double] Threshold for minor allele frequency cut-off.

`interactive` [logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set interactive=FALSE and save the returned plot object (p_marker) via ggplot2::ggsave(p=p_marker, other_arguments) or pdf(outfile) print(p_marker) dev.off().

`verbose` [logical] If TRUE, progress info is printed to standard out.

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#snp`. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#snp`. Default: NULL, i.e. no filtering on markers.

`legend_text_size`
: [integer] Size for legend text.

`legend_title_size`
: [integer] Size for legend title.

`axis_text_size`
: [integer] Size for axis text.

`axis_title_size`
: [integer] Size for axis title.

`title_size`
: [integer] Size for plot title.

`subplot_label_size`
: [integer] Size of the subplot labeling.

`path2plink`
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

`showPlinkOutput`
: [logical] If TRUE, plink log and error messages are printed to standard out.

Details

perMarkerQC wraps around the marker QC functions check_snp_missingness, check_hwe and check_maf. For details on the parameters and outputs, check these function documentations.

Value

Named [list] with i) fail_list, a named [list] with 1. SNP_missingness, containing SNP IDs [vector] failing the missingness threshold lmissTh, 2. hwe, containing SNP IDs [vector] failing the HWE exact test threshold hweTh and 3. maf, containing SNPs Ids [vector] failing the MAF threshold mafTh/MAC threshold macTh and ii) p_markerQC, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots of check_snp_missingness, check_hwe and check_maf, which can be shown by print(p_markerQC). List entries contain NULL if that specific check was not chosen.

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
# All quality control checks
## Not run:
# run on all markers and individuals
fail_markers <- perMarkerQC(indir=indir, qcdir=qcdir, name=name,
interactive=FALSE, verbose=TRUE, path2plink=path2plink)

# run on subset of individuals and markers
keep_individuals_file <- system.file("extdata", "keep_individuals",
package="plinkQC")
```

```
extract_markers_file <- system.file("extdata", "extract_markers",
package="plinkQC")
fail_markers <- perMarkerQC(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink,
keep_individuals=keep_individuals_file, extract_markers=extract_markers_file)

## End(Not run)
```

pruning_ld *Pruning of SNPs in Linkage Disequilibrium*

Description

Runs plink –indep-pairwise to remove SNPs in linkage disequilibrium. It excludes variants that found in a high linkage disequilbirum loci.

Usage

```
pruning_ld(
  indir,
  name,
  qcdir = indir,
  path2plink = NULL,
  filter_high_ldregion = TRUE,
  high_ldregion_file = NULL,
  genomebuild = "hg38",
  window_size = 50,
  step_size = 5,
  r_2 = 0.2,
  showPlinkOutput = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  verbose = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.genome and name.imiss.

`qcdir` [character] /path/to/directory to where name.genome as returned by plink –genome will be saved. Per default qcdir=indir. If run.check_relatedness is FALSE, it is assumed that plink –missing and plink –genome have been run and qcdir/name.imiss and qcdir/name.genome exist. User needs writing permission to qcdir.

path2plink
: [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

filter_high_ldregion
: [logical] Should high LD regions be filtered before IBD estimation; carried out per default with high LD regions for hg19 provided as default via genomebuild. For alternative genome builds not provided or non-human data, high LD regions files can be provided via high_ldregion_file.

high_ldregion_file
: [character] Path to file with high LD regions used for filtering before IBD estimation if filter_high_ldregion == TRUE, otherwise ignored; for human genome data, high LD region files are provided and can simply be chosen via genomebuild. Files have to be space-delimited, no column names with the following columns: chromosome, region-start, region-end, region number. Chromosomes are specified without 'chr' prefix. For instance: 1 48000000 52000000 1 2 86000000 100500000 2

genomebuild
: [character] Name of the genome build of the PLINK file annotations, ie mappings in the name.bim file. Will be used to remove high-LD regions based on the coordinates of the respective build. Options are hg18, hg19 and hg38. See @details.

window_size
: [integer] The size of the window (in variant count) in which variants in the window are pruned

step_size
: [integer] The variant count to shift the window

r_2
: [float] The threshold in which variant pairs with a squared correlation above the threshold are removed

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

keep_individuals
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

verbose
: [logical] If TRUE, progress info is printed to standard out.

Value

Files with a .pruned with the pruned SNPS

Examples

```
## Not run:
indir <- system.file("extdata", package="plinkQC")
name <- 'data'
path2plink <- "path/to/plink"

# whole dataset
relatednessQC <- check_relatedness(indir=indir, name=name, interactive=FALSE,
run.check_relatedness=FALSE, path2plink=path2plink)

# subset of dataset
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
fail_relatedness <- check_relatedness(indir=qcdir, name=name,
remove_individuals=remove_individuals_file, path2plink=path2plink)

## End(Not run)
```

relatednessFilter *Remove related individuals while keeping maximum number of individuals*

Description

relatednessFilter takes a data.frame with pair-wise relatedness measures of samples and returns pairs of individual IDs that are related as well as a list of suggested individual IDs to remove. relatednessFilter finds pairs of samples whose relatedness estimate is larger than the specified relatednessTh. Subsequently, for pairs of individual that do not have additional relatives in the dataset, the individual with the worse otherCriterionMeasure (if provided) or arbitrarily individual 1 of that pair is selected and returned as the individual failing the relatedness check. For more complex family structures, the unrelated individuals per family are selected (e.g. in a simple case of a parents-offspring trio, the offspring will be marked as fail, while the parents will be kept in the analysis). Selection is achieved by constructing subgraphs of clusters of individuals that are related. relatednessFilter then finds the maximum independent set of vertices in the subgraphs of related individuals. If all individuals are related (i.e. all maximum independent sets are 0), one individual of that cluster will be kept and all others listed as failIDs.

Usage

```
relatednessFilter(
  relatedness,
  otherCriterion = NULL,
  relatednessTh,
  otherCriterionTh = NULL,
  otherCriterionThDirection = c("gt", "ge", "lt", "le", "eq"),
  relatednessIID1 = "IID1",
  relatednessIID2 = "IID2",
  relatednessFID1 = NULL,
  relatednessFID2 = NULL,
  relatednessRelatedness = "PI_HAT",
  otherCriterionIID = "IID",
  otherCriterionMeasure = NULL,
  verbose = FALSE
)
```

Arguments

`relatedness` [data.frame] containing pair-wise relatedness estimates (in column [relatednessRelatedness]) for individual 1 (in column [relatednessIID1] and individual 2 (in column [relatednessIID1]). Columns relatednessIID1, relatednessIID2 and relatednessRelatedness have to present, while additional columns such as family IDs can be present. Default column names correspond to column names in output of plink –genome (https://www.cog-genomics.org/plink/1.9/ibd). All original columns for pair-wise highIBDTh fails will be returned in fail_IBD.

`otherCriterion` [data.frame] containing a QC measure (in column [otherCriterionMeasure]) per individual (in column [otherCriterionIID]). otherCriterionMeasure and otherCriterionIID have to present, while additional columns such as family IDs can be present. IIDs in relatednessIID1 have to be present in otherCriterionIID.

`relatednessTh` [double] Threshold for filtering related individuals. Individuals, whose pair-wise relatedness estimates are greater than this threshold are considered related.

`otherCriterionTh`

[double] Threshold for filtering individuals based on otherCriterionMeasure. If related individuals fail this threshold they will automatically be excluded.

`otherCriterionThDirection`

[character] Used to determine the direction for failing the otherCriterionTh. If 'gt', individuals whose otherCriterionMeasure > otherCriterionTh will automatically be excluded. For pairs of individuals that have no other related samples in the cohort: if both otherCriterionMeasure < otherCriterionTh, the individual with the larger otherCriterionMeasure will be excluded.

`relatednessIID1`

[character] Column name of column containing the IDs of the first individual.

`relatednessIID2`

[character] Column name of column containing the IDs of the second individual.

`relatednessFID1`
: [character, optional] Column name of column containing the family IDs of the first individual; if only relatednessFID1 but not relatednessFID2 provided, or none provided even though present in relatedness, FIDs will not be returned.

`relatednessFID2`
: [character, optional] Column name of column containing the family IDs of the second individual; if only relatednessFID2 but not relatednessFID1 provided, or none provided even though present in relatedness, FIDs will not be returned.

`relatednessRelatedness`
: [character] Column name of column containing the relatedness estimate.

`otherCriterionIID`
: [character] Column name of column containing the individual IDs.

`otherCriterionMeasure`
: [character] Column name of the column containing the measure of the otherCriterion (for instance SNP missingness rate).

`verbose`
: [logical] If TRUE, progress info is printed to standard out.

Value

named [list] with i) relatednessFails, a [data.frame] containing the data.frame relatedness after filtering for pairs of individuals in relatednessIID1 and relatednessIID2, that fail the relatedness QC; the data.frame is reordered with the fail individuals in column 1 and their related individuals in column 2 and ii) failIDs, a [data.frame] with the [IID]s (and [FID]s if provided) of the individuals that fail the relatednessTh.

---

rename_variant_identifiers
Renaming variants

Description

Changes the format of the variant identifier. The default is in the format of chr1:12345[hg38].

Usage

```
rename_variant_identifiers(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink2 = NULL,
  format = "@:#[hg38]",
  showPlinkOutput = TRUE,
  plink2format = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK 2.0 data file name.pgen, name.pvar, name.psam

`name` [character] Prefix of PLINK 2.0 files, i.e. name.pgen, name.pvar, name.psam

`qcdir` [character] /path/to/directory where name.sscore as returned by plink2 –score will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink2` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/2.0/`) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`format` [character] This gives the template to rewrite the variant identifier. A '@' represents the chromosome code, and a '#' represents the base-pair position.

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

`plink2format` [logical] If TRUE, data is in plink2 format (i.e. name.pvar, name.psam, and name.pgen)

Value

Files with a .renamed in them that have the renamed variants

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data.hg38"
path2plink <- '/path/to/plink'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
rename_variant_identifiers(indir=indir, qcdir=qcdir, name=name, path2plink2 = path2plink2)

## End(Not run)
```

`run_ancestry_format` *Running functions to format data for ancestry prediction*

Description

This function runs convert_to_plink2 and rename_variant_identifiers to format the data for the ancestry identification with superpop_classification

Usage

```
run_ancestry_format(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink2 = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE,
  format = "@:#[hg38]",
  plink2format = FALSE,
  var_format = FALSE,
  path2load_mat = NULL,
  write_multiqc = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK 1.9 data file name.bim, name.fam, name.bed

`name` [character] Prefix of PLINK 1.9 files, i.e. name.bim, name.fam, name.bed

`qcdir` [character] /path/to/directory where the plink2 data formations as returned by plink2 –make-pgen will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink2` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/2.0/) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`(’plink’).

`keep_individuals`

[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`

[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it’s okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it’s okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`showPlinkOutput`
: [logical] If TRUE, plink log and error messages are printed to standard out.

`format`
: [character] This gives the template to rewrite the variant identifier. A ’@’ represents the chromosome code, and a ’#’ represents the base-pair position.

`plink2format`
: [logical] If TRUE, data is in plink2 format (i.e. name.pvar, name.psam, and name.pgen)

`var_format`
: [logical] If TRUE, variant identifiers are in correct format already and rename_variant_identifiers will not be run

`path2load_mat`
: [character] /path/to/directory where loading matrices are kept. This can be downloaded from the github repo. Note that the name of the file before the .eigenvec.allele or .acount must be included in file path.

`write_multiqc`
: [logical] If TRUE, will output a multiQC-compatible report file.

Value

Name of file with correct format

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data"
path2plink <- '/path/to/plink'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
run_ancestry_format(indir=indir, qcdir=qcdir,
  name=name, path2plink2 = path2plink2)

## End(Not run)
```

run_ancestry_prediction

Projecting the study data set onto the PC space of the reference dataset

Description

Projects the study dataset onto the PC space of the reference dataset. The output of this function as input in a random forest classifier to predict the genomic ancestry of the samples. Genomic data version hg38 with variant identifiers in the format of 1:12345[hg38] is needed for ancestry identification to work.

Usage

```
run_ancestry_prediction(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink2 = NULL,
  path2load_mat = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  extract_markers = NULL,
  exclude_markers = NULL,
  showPlinkOutput = TRUE,
  plink2format = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK 2.0 data file name.pgen, name.pvar, name.psam

`name` [character] Prefix of PLINK 2.0 files, i.e. name.pgen, name.pvar, name.psam

`qcdir` [character] /path/to/directory where name.sscore as returned by plink2 –score will be saved to. User needs writing permission to qcdir. Per default is qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink2` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/2.0/) i.e. plink 2 should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

`path2load_mat` [character] /path/to/directory where loading matrices are kept. This can be downloaded from: https://github.com/meyer-lab-cshl/plinkQCAncestryData. Note that file names before the .acount or .eigenvec.allele must be included in file path.

keep_individuals
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

plink2format
: [logical] If TRUE, data is in plink2 format (i.e. name.pvar, name.psam, and name.pgen)

Value

A .sscore file with the input data projected onto the reference data PCs

Examples

```
indir <- system.file("extdata", package="plinkQC")
qcdir <- tempdir()
name <- "data.hg38"
path2plink <- '/path/to/plink'
path2load_mat <- '/path/to/load_mat/merged_chrs.postQC.train.pca'
## Not run:
# the following code is not run on package build, as the path2plink on the
# user system is not known.
superpop_classification(indir=indir, qcdir=qcdir, name=name,
path2plink2 = path2plink2, path2load_mat = path2load_mat)

## End(Not run)
```

run_check_heterozygosity

Run PLINK heterozygosity rate calculation

Description

Run plink –het to calculate heterozygosity rates per individual.

Usage

```
run_check_heterozygosity(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory to save name.het as returned by plink –het. User needs writing permission to qcdir. Per default qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

`keep_individuals` [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals` [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column

and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

Details

All, run_check_heterozygosity, run_check_missingness and their evaluation by evaluate_check_het_and_miss can simply be invoked by check_het_and_miss.

Examples

```
indir <- system.file("extdata", package="plinkQC")
name <- 'data'
qcdir <- tempdir()
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# heterozygosity check on all individuals in dataset
run <- run_check_heterozygosity(indir=indir, qcdir=qcdir, name=name,
path2plink=path2plink)

#' # heterozygosity on subset of dataset
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
run <- run_check_heterozygosity(indir=indir, qcdir=qcdir, name=name,
remove_individuals=remove_individuals_file,path2plink=path2plink)

## End(Not run)
```

run_check_missingness *Run PLINK missingness rate calculation*

Description

Run plink –missing to calculate missing genotype rates per individual.

Usage

```
run_check_missingness(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory to save name.imiss as returned by plink –missing. User needs writing permission to qcdir. Per default qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink` [character] Absolute path to PLINK executable (`https://www.cog-genomics.org/plink/1.9/`) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See `https://www.cog-genomics.org/plink/1.9/filter#indiv`. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed

from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

Details

All, run_check_heterozygosity, run_check_missingness and their evaluation by evaluate_check_het_and_miss can simply be invoked by check_het_and_miss.

Examples

```
indir <- system.file("extdata", package="plinkQC")
name <- 'data'
qcdir <- tempdir()
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# missingness check on all individuals in dataset
run <- run_check_missingness(indir=indir, qcdir=qcdir, name=name,
path2plink=path2plink)

# missingness on subset of dataset
remove_individuals_file <- system.file("extdata", "remove_individuals",
package="plinkQC")
run <- run_check_missingness(indir=indir, qcdir=qcdir, name=name,
remove_individuals=remove_individuals_file, path2plink=path2plink)

## End(Not run)
```

---

run_check_relatedness *Run PLINK IBD estimation*

Description

Run LD pruning on dataset with plink –exclude range highldfile –indep-pairwise 50 5 0.2, where highldfile contains regions of high LD as provided by Anderson et (2010) Nature Protocols. Subsequently, plink –genome is run on the LD pruned, maf-filtered data. plink –genome calculates identity by state (IBS) for each pair of individuals based on the average proportion of alleles shared at genotyped SNPs. The degree of recent shared ancestry,i.e. the identity by descent (IBD) can be estimated from the genome-wide IBS. The proportion of IBD between two individuals is returned by –genome as PI_HAT.

Usage

```
run_check_relatedness(
  indir,
  name,
  qcdir = indir,
  highIBDTh = 0.185,
  mafThRelatedness = 0.1,
  path2plink = NULL,
  filter_high_ldregion = TRUE,
  high_ldregion_file = NULL,
  genomebuild = "hg19",
  showPlinkOutput = TRUE,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  verbose = FALSE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory to save name.genome as returned by plink –genome. User needs writing permission to qcdir. Per default qcdir=indir.

`highIBDTh` [double] Threshold for acceptable proportion of IBD between pair of individuals; only pairwise relationship estimates larger than this threshold will be recorded.

`mafThRelatedness`
[double] Threshold of minor allele frequency filter for selecting variants for IBD estimation.

`path2plink` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by `exec`('plink').

`filter_high_ldregion`
[logical] Should high LD regions be filtered before IBD estimation; carried out per default with high LD regions for hg19 provided as default via `genomebuild`. For alternative genome builds not provided or non-human data, high LD regions files can be provided via `high_ldregion_file`.

`high_ldregion_file`
[character] Path to file with high LD regions used for filtering before IBD estimation if `filter_high_ldregion` == TRUE, otherwise ignored; for human genome data, high LD region files are provided and can simply be chosen via

genomebuild. Files have to be space-delimited, no column names with the following columns: chromosome, region-start, region-end, region number. Chromosomes are specified without 'chr' prefix. For instance: 1 48000000 52000000 1 2 86000000 100500000 2

genomebuild
: [character] Name of the genome build of the PLINK file annotations, ie mappings in the name.bim file. Will be used to remove high-LD regions based on the coordinates of the respective build. Options are hg18, hg19 and hg38. See @details.

showPlinkOutput
: [logical] If TRUE, plink log and error messages are printed to standard out.

keep_individuals
: [character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

remove_individuals
: [character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

exclude_markers
: [character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

extract_markers
: [character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

verbose
: [logical] If TRUE, progress info is printed to standard out.

Details

Both run_check_relatedness and its evaluation via evaluate_check_relatedness can simply be invoked by check_relatedness.

The IBD estimation is conducted on LD pruned data and in a first step, high LD regions are excluded. The regions were derived from the high-LD-regions file provided by Anderson et (2010) Nature Protocols. These regions are in NCBI36 (hg18) coordinates and were lifted to GRCh37 (hg19) and GRC38 (hg38) coordinates using the liftOver tool available here: https://genome.ucsc.edu/cgi-bin/hgLiftOver. The 'Minimum ratio of bases that must remap' which was set to 0.5 and the 'Allow multiple output regions' box ticked; for all other parameters, the default options were selected. LiftOver files were generated on July 9,2019. The commands for formatting the files are provided in system.file("extdata", 'liftOver.cmd', package="plinkQC").

Examples

```
indir <- system.file("extdata", package="plinkQC")
name <- 'data'
qcdir <- tempdir()
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
# Relatedness estimation based in all markers in dataset
run <- run_check_relatedness(indir=indir, qcdir=qcdir, name=name,
path2plink=path2plink)

# relatedness estimation on subset of dataset
keep_individuals_file <- system.file("extdata", "keep_individuals",
package="plinkQC")
run <- run_check_relatedness(indir=indir, qcdir=qcdir, name=name,
keep_individuals=keep_individuals_file, path2plink=path2plink)

## End(Not run)
```

`run_check_sex` *Run PLINK sexcheck*

Description

Run plink –sexcheck to calculate the heterozygosity rate across X-chromosomal variants.

Usage

```
run_check_sex(
  indir,
  name,
  qcdir = indir,
  verbose = FALSE,
  path2plink = NULL,
  keep_individuals = NULL,
  remove_individuals = NULL,
  exclude_markers = NULL,
  extract_markers = NULL,
  showPlinkOutput = TRUE
)
```

Arguments

`indir` [character] /path/to/directory containing the basic PLINK data files name.bim, name.bed, name.fam files.

`name` [character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam.

`qcdir` [character] /path/to/directory to save name.sexcheck as returned by plink –check-sex. User needs writing permission to qcdir. Per default qcdir=indir.

`verbose` [logical] If TRUE, progress info is printed to standard out.

`path2plink` [character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accessible as path2plink -h. The full name of the executable should be specified: for windows OS, this means path/plink.exe, for unix platforms this is path/plink. If not provided, assumed that PATH set-up works and PLINK will be found by exec('plink').

`keep_individuals`
[character] Path to file with individuals to be retained in the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples not listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`remove_individuals`
[character] Path to file with individuals to be removed from the analysis. The file has to be a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column. All samples listed in this file will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#indiv. Default: NULL, i.e. no filtering on individuals.

`exclude_markers`
[character] Path to file with makers to be removed from the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All listed variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`extract_markers`
[character] Path to file with makers to be included in the analysis. The file has to be a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces). All unlisted variants will be removed from the current analysis. See https://www.cog-genomics.org/plink/1.9/filter#snp. Default: NULL, i.e. no filtering on markers.

`showPlinkOutput`
[logical] If TRUE, plink log and error messages are printed to standard out.

Details

Both `run_check_sex` and its evaluation `evaluate_check_sex` can simply be invoked by `check_sex`.

Examples

```
indir <- system.file("extdata", package="plinkQC")
name <- 'data'
qcdir <- tempdir()
path2plink <- '/path/to/plink'
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
```

```
# simple sexcheck on all individuals in dataset
run <- run_check_sex(indir=indir, qcdir=qcdir, name=name)

# sexcheck on subset of dataset
keep_individuals_file <- system.file("extdata", "keep_individuals",
package="plinkQC")
run <- run_check_sex(indir=indir, qcdir=qcdir, name=name,
keep_individuals=keep_individuals_file, path2plink=path2plink)

## End(Not run)
```

---

testNumerics *Test lists for different properties of numerics*

Description

Test all elements of a list if they are numeric, positive numbers, integers or proportions (range 0-1).

Usage

```
testNumerics(numbers, positives = NULL, integers = NULL, proportions = NULL)
```

Arguments

numbers	[list] whose elements are tested for being numeric.
positives	[list] whose elements are tested for being positive numbers.
integers	[list] whose elements are tested for being integers.
proportions	[list] whose elements are tested for being proportions. between 0 and 1.

Index