

Package ‘plinkQC’

November 9, 2018

Type Package

Title Genotype Quality Control with 'PLINK'

Version 0.2.0

URL <https://github.com/HannahVMeyer/plinkQC>

BugReports <https://github.com/HannahVMeyer/plinkQC/issues>

Maintainer Hannah Meyer <hannah@ebi.ac.uk>

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.

Depends R (>= 3.3.0)

Imports methods, optparse, data.table (>= 1.11.0), R.utils, ggplot2, ggforce, ggrepel, cowplot, UpSetR, dplyr, sys

Suggests testthat, knitr, rmarkdown

License MIT + file LICENSE

SystemRequirements plink (1.9)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

VignetteBuilder knitr

NeedsCompilation no

Author Hannah Meyer [aut, cre] (<<https://orcid.org/0000-0003-4564-0899>>)

Repository CRAN

Date/Publication 2018-11-09 14:20:02 UTC

R topics documented:

checkPlink	2
check_ancestry	3
check_het_and_miss	5
check_hwe	7
check_maf	8
check_relatedness	10
check_sex	12
check_snp_missingness	14
cleanData	15
evaluate_check_ancestry	18
evaluate_check_het_and_miss	20
evaluate_check_relatedness	21
evaluate_check_sex	23
overviewPerIndividualQC	25
overviewPerMarkerQC	26
perIndividualQC	27
perMarkerQC	31
relatednessFilter	33
run_check_ancestry	34
run_check_heterozygosity	35
run_check_missingness	36
run_check_relatedness	37
run_check_sex	39
testNumerics	40
Index	41

checkPlink	<i>Check PLINK software access</i>
------------	------------------------------------

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 accesible 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 <code>exec_wait('plink')</code> .
------------	--

Value

Path to PLINK executable.

check_ancestry	<i>Identification of individuals of divergent ancestry</i>
----------------	--

Description

Runs and evaluates results of `plink -pca` on merged genotypes from individuals to be QCed and individuals of reference population of known genotypes. Currently, check ancestry only supports automatic selection of individuals of European descent. It uses information from principal components 1 and 2 returned by `plink -pca` to find the center of the European reference samples (`mean(PC1_europeanRef)`, `mean(PC2_europeanRef)`). It then computes the maximum Euclidean distance (`maxDist`) of the European reference samples from this centre. All study samples whose Euclidean distance from the centre falls outside the circle described by the radius `r=europeanTh*maxDist` are considered non-European and their IDs are returned as failing the ancestry check. `check_ancestry` creates a scatter plot of PC1 versus PC2 colour-coded for samples of the reference populations and the study population.

Usage

```
check_ancestry(indir, name, qcdir = indir, prefixMergedDataset,
  europeanTh = 1.5, refSamples = NULL, refColors = NULL,
  refSamplesFile = NULL, refColorsFile = NULL, refSamplesIID = "IID",
  refSamplesPop = "Pop", refColorsColor = "Color",
  refColorsPop = "Pop", studyColor = "#2c7bb6",
  run.check_ancestry = TRUE, interactive = FALSE, verbose = verbose,
  path2plink = NULL, showPlinkOutput = TRUE)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] prefix of plink files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> .
<code>qcdir</code>	[character] /path/to/directory where <code>prefixMergedDataset.eigenvec</code> results as returned by <code>plink -pca</code> should be saved. Per default <code>qcdir=indir</code> . If <code>run.check_ancestry</code> is <code>FALSE</code> , it is assumed that <code>plink -pca prefixMergedDataset</code> has been run and <code>qcdir/prefixMergedDataset.eigenvec</code> exists. User needs writing permission to <code>qcdir</code> .
<code>prefixMergedDataset</code>	[character] Prefix of merged dataset (study and reference samples) used in <code>plink -pca</code> , resulting in <code>prefixMergedDataset.eigenvec</code> .
<code>europeanTh</code>	[double] Scaling factor of radius to be drawn around center of European reference samples, with study samples inside this radius considered to be of European descent and samples outside this radius of non-European descent. The radius is computed as the maximum Euclidean distance of European reference samples to the centre of European reference samples.

refSamples	[data.frame] Dataframe with sample identifiers [refSamplesIID] corresponding to IIDs in prefixMergedDataset.eigenvec and population identifier [refSamplesPop] corresponding to population IDs [refColorsPop] in refColorsfile/refColors. Either refSamples or refSamplesFile have to be specified.
refColors	[data.frame, optional] Dataframe with population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColorsFile) default colors are used.
refSamplesFile	[character] /path/to/File/with/reference samples. Needs columns with sample identifiers [refSamplesIID] corresponding to IIDs in prefixMergedDataset.eigenvec and population identifier [refSamplesPop] corresponding to population IDs [refColorsPop] in refColorsfile/refColors.
refColorsFile	[character, optional] /path/to/File/with/Population/Colors containing population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColors) default colors for are used.
refSamplesIID	[character] Column name of reference sample IDs in refSamples/refSamplesFile.
refSamplesPop	[character] Column name of reference sample population IDs in refSamples/refSamplesFile.
refColorsColor	[character] Column name of population colors in refColors/refColorsFile
refColorsPop	[character] Column name of reference sample population IDs in refColors/refColorsFile.
studyColor	[character] Colour to be used for study population.
run.check_ancestry	[logical] Should plink -pca be run to determine principal components of merged dataset; if FALSE, it is assumed that plink -pca has been run successfully and qcdir/prefixMergedDataset.eigenvec is present; <code>check_ancestry</code> 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_ancestry) via ggplot2::ggsave(p=p_ancestry, other_arguments) or pdf(outfile) print(p_ancestry) dev.off().
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 accesible 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 <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

Value

Named [list] with i) fail_ancestry, containing a [data.frame] with FID and IID of non-European individuals and ii) p_ancestry, a ggplot2-object 'containing' a scatter plot of PC1 versus PC2 colour-coded for samples of the reference populations and the study population, which can be shown by print(p_ancestry).

Examples

```
## Not run:
indir <- system.file("extdata", package="plinkQC")
name <- "data"
fail_ancestry <- check_ancestry(indir=indir, name=name,
refSamplesFile=paste(indir, "/HapMap_ID2Pop.txt", sep=""),
refColorsFile=paste(indir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE,
run.check_ancestry=FALSE)

## End(Not run)
```

check_het_and_miss	<i>Identification of individuals with outlying missing genotype or heterozygosity rates</i>
--------------------	---

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, interactive = FALSE,
  verbose = FALSE, path2plink = 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, name.het and name.imiss.
qcdir	[character] /path/to/directory where name.het as returned by <code>plink --het</code> and name.imiss as returned by <code>plink --missing</code> will be saved. Per default qcdir=indir. If <code>run.check_het_and_miss</code> is FALSE, it is assumed that <code>plink --missing</code> and <code>plink --het</code> 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 $\text{mean}(\text{het}) \pm \text{hetTh} * \text{sd}(\text{het})$ will be returned as failing heterozygosity check; has to be larger than 0.
run.check_het_and_miss	[logical] Should plink <code>--missing</code> and plink <code>--het</code> be run to determine genotype missingness and heterozygosity rates; if FALSE, it is assumed that plink <code>--missing</code> and plink <code>--het</code> have been run and <code>qcdir/name.imiss</code> and <code>qcdir/name.het</code> are present; <code>check_het_and_miss</code> 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 <code>interactive=FALSE</code> and save the returned plot object (<code>p_het_imiss</code>) via <code>ggplot2::ggsave(p=p_het_imiss, other_arguments)</code> or <code>pdf(outfile, print(p_het_imiss) dev.off())</code> .
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 <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that PATH set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

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"
fail_het_miss <- check_het_and_miss(indir=indir, name=name,
run.check_het_and_miss=FALSE, interactive=FALSE)

## End(Not run)
```

check_hwe	<i>Identification of SNPs showing a significant deviation from Hardy-Weinberg- equilibrium (HWE)</i>
-----------	--

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)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> .
<code>qcdir</code>	[character] /path/to/directory where results will be written to. If <code>perIndividualQC</code> was conducted, this directory should be the same as <code>qcdir</code> specified in <code>perIndividualQC</code> , i.e. it contains <code>name.fail.IDs</code> with IIDs of individuals that failed QC. User needs writing permission to <code>qcdir</code> . Per default, <code>qcdir=indir</code> .
<code>hweTh</code>	[double] Significance threshold for deviation from HWE.
<code>interactive</code>	[logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set <code>interactive=FALSE</code> and save the returned plot object (<code>p_hwe</code>) via <code>ggplot2::ggsave(p=p_hwe, other_arguments)</code> or <code>pdf(outfile) print(p_hwe) dev.off()</code> .
<code>path2plink</code>	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. <code>plink</code> should be accesible as <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that <code>PATH</code> set-up works and PLINK will be found by <code>exec_wait('plink')</code> .

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.

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:
fail_hwe <- check_hwe(indir=indir, qcdir=qcdir, name=name, interactive=FALSE,
verbose=TRUE, path2plink=path2plink)

## 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)
```

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 ($macTh = mafTh \cdot 2 \cdot NrSamples$).
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 <code>ggplot2::ggsave(p=p_maf, other_arguments)</code> or <code>pdf(outfile) print(p_maf) dev.off()</code> .
path2plink	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accesible 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 <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

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:
fail_maf <- check_maf(indir=indir, qcdir=qcdir, name=name, macTh=15,
interactive=FALSE, verbose=TRUE, path2plink=path2plink)

## 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,
  imissTh = 0.03, run.check_relatedness = TRUE, interactive = FALSE,
  verbose = FALSE, path2plink = NULL, showPlinkOutput = TRUE)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> , <code>name.genome</code> and <code>name.imiss</code> .
<code>qcdir</code>	[character] /path/to/directory to where <code>name.genome</code> as returned by <code>plink --genome</code> will be saved. Per default <code>qcdir=indir</code> . If <code>run.check_relatedness</code> is <code>FALSE</code> , it is assumed that <code>plink --missing</code> and <code>plink --genome</code> have been run and <code>qcdir/name.imiss</code> and <code>qcdir/name.genome</code> exist. User needs writing permission to <code>qcdir</code> .

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)
run.check_relatedness	[logical] Should plink <code>--genome</code> be run to determine pairwise IBD of individuals; if FALSE, it is assumed that plink <code>--genome</code> and plink <code>--missing</code> have been run and <code>qcdir/name.imiss</code> and <code>qcdir/name.genome</code> are present; <code>check_relatedness</code> 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 <code>interactive=FALSE</code> and save the returned plot object (<code>p_IBD()</code>) via <code>ggplot2::ggsave(p=p_IBD, other_arguments)</code> or <code>pdf(outfile) print(p_IBD) dev.off()</code> .
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 accesible as <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that PATH set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

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 `highIBDTh` 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. $P(\text{IBD}=2) + 0.5 \cdot P(\text{IBD}=1)$), `PHE` (Pairwise phenotypic code (1, 0, -1 = AA, AU, and UU pairs, respectively)), `DST` (IBS distance, i.e. $(\text{IBS2} + 0.5 \cdot \text{IBS1}) / (\text{IBS0} + \text{IBS1} + \text{IBS2})$), `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'
relatednessQC <- check_relatedness(indir=indir, name=name, interactive=FALSE,
run.check_relatedness=FALSE)

## 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,
path2plink = 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 and name.sexcheck.
qcdir	[character] /path/to/directory to save name.sexcheck as returned by plink <code>-check-sex</code> . Per default qcdir=indir. If run.check_sex is FALSE, it is assumed that plink <code>-check-sex</code> 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 <code>--check-sex</code> be run? if set to FALSE, it is assumed that plink <code>--check-sex</code> has been run and <code>qcdir/name.sexcheck</code> is present; <code>check_sex</code> will fail with missing file error otherwise.
externalSex	[data.frame, optional] Dataframe with sample IDs [<code>externalSexID</code>] and sex [<code>externalSexSex</code>] 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 <code>Sex==SNPSEX</code> automatically corrected: this will directly change the <code>name.bim/.bed/.fam</code> 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 <code>interactive=FALSE</code> and save the returned plot object (<code>p_sexcheck</code>) via <code>ggplot2::ggsave(p=p_sexcheck, other_arguments)</code> or <code>pdf(outfile) print(p_sexcheck) dev.off()</code> .
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 <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that PATH set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

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"
fail_sex <- check_sex(indir=indir, name=name, run.check_sex=FALSE,
interactive=FALSE, verbose=FALSE)

## 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)
```

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 <code>perIndividualQC</code> was conducted, this directory should be the same as qcdir specified in <code>perIndividualQC</code> , 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 <code>ggplot2::ggsave(p=p_lmiss, other_arguments)</code> or <code>pdf(outfile) print(p_lmiss) dev.off()</code> .
path2plink	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. plink should be accesible 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 <code>exec_wait('plink')</code> .
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.

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:
fail_snp_missingness <- check_snp_missingness(qcdir=qcdir, indir=indir,
name=name, interactive=FALSE, verbose=TRUE, path2plink=path2plink)

## 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,
  filterAncestry = TRUE, filterRelated = TRUE,
  filterSNPMissingness = TRUE, lmissTh = 0.01, filterHWE = TRUE,
  hweTh = 1e-05, filterMAF = TRUE, macTh = 20, mafTh = NULL,
  path2plink = NULL, verbose = FALSE, 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).
filterAncestry	[logical] Set to exclude samples that failed ancestry check (via check_ancestry or perIndividualQC). Requires file qcdir/name.fail-ancestry.IDs (automatically created by perIndividualQC if do.check_ancestry 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).
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 accesible 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 <code>exec_wait('plink')</code> .
verbose	[logical] If TRUE, progress info is printed to standard out.
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 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)

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

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

## End(Not run)

```

 evaluate_check_ancestry

Evaluate results from PLINK PCA on combined study and reference data

Description

Evaluates and depicts results of `plink -pca` (via [run_check_ancestry](#) or externally conducted `pca`) on merged genotypes from individuals to be QCed and individuals of reference population of known genotypes. Currently, check ancestry only supports automatic selection of individuals of European descent. It uses information from principal components 1 and 2 returned by `plink -pca` to find the center of the European reference samples (`mean(PC1_europeanRef)`, `mean(PC2_europeanRef)`). It computes the maximum Euclidean distance (`maxDist`) of the European reference samples from this centre. All study samples whose Euclidean distance from the centre falls outside the circle described by the radius `r=europeanTh* maxDist` are considered non-European and their IDs are returned as failing the ancestry check. `check_ancestry` creates a scatter plot of PC1 versus PC2 colour-coded for samples of the reference populations and the study population.

Usage

```
evaluate_check_ancestry(indir, name, prefixMergedDataset, qcdir = indir,
  europeanTh = 1.5, refSamples = NULL, refColors = NULL,
  refSamplesFile = NULL, refColorsFile = NULL, refSamplesIID = "IID",
  refSamplesPop = "Pop", refColorsColor = "Color",
  refColorsPop = "Pop", studyColor = "#2c7bb6", interactive = FALSE)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> .
<code>prefixMergedDataset</code>	[character] Prefix of merged dataset (study and reference samples) used in <code>plink -pca</code> , resulting in <code>prefixMergedDataset.eigenvec</code> .
<code>qcdir</code>	[character] /path/to/directory/with/QC/results containing <code>prefixMergedDataset.eigenvec</code> results as returned by <code>plink -pca</code> . Per default <code>qcdir=indir</code> .
<code>europeanTh</code>	[double] Scaling factor of radius to be drawn around center of European reference samples, with study samples inside this radius considered to be of European descent and samples outside this radius of non-European descent. The radius is computed as the maximum Euclidean distance of European reference samples to the centre of European reference samples.
<code>refSamples</code>	[data.frame] Dataframe with sample identifiers [<code>refSamplesIID</code>] corresponding to IIDs in <code>prefixMergedDataset.eigenvec</code> and population identifier [<code>refSamplePop</code>] corresponding to population IDs [<code>refColorsPop</code>] in <code>refColorsfile/refColors</code> . Either <code>refSamples</code> or <code>refSamplesFile</code> have to be specified.

refColors	[data.frame, optional] Dataframe with population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColorsFile) default colors are used.
refSamplesFile	[character] /path/to/File/with/reference samples. Needs columns with sample identifiers [refSamplesIID] corresponding to IIDs in prefixMergedDataset.eigenvec and population identifier [refSamplesPop] corresponding to population IDs [refColorsPop] in refColorsfile/refColors. If both refSamplesFile and refSamples are not NULL, refSamplesFile information is used.
refColorsFile	[character, optional] /path/to/File/with/Population/Colors containing population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColors) default colors for are used. If both refColorsFile and refColors are not NULL, refColorsFile information is used.
refSamplesIID	[character] Column name of reference sample IDs in refSamples/refSamplesFile.
refSamplesPop	[character] Column name of reference sample population IDs in refSamples/refSamplesFile.
refColorsColor	[character] Column name of population colors in refColors/refColorsFile
refColorsPop	[character] Column name of reference sample population IDs in refColors/refColorsFile.
studyColor	[character] Colour to be used for study population if plot is TRUE.
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().

Details

Both `run_check_ancestry` and `evaluate_check_ancestry` can simply be invoked by `check_ancestry`.

Value

Named [list] with i) `fail_ancestry`, containing a [data.frame] with FID and IID of non-European individuals and ii) `p_ancestry`, a ggplot2-object 'containing' a scatter plot of PC1 versus PC2 colour-coded for samples of the reference populations and the study population, which can be shown by `print(p_ancestry)`.

Examples

```
## Not run:
qcdir <- system.file("extdata", package="plinkQC")
name <- "data"
fail_ancestry <- evaluate_check_ancestry(indir=qcdir, name=name,
refSamplesFile=paste(qcdir, "/HapMap_ID2Pop.txt", sep=""),
refColorsFile=paste(qcdir, "/HapMap_PopColors.txt", sep=""),
prefixMergedDataset="data.HapMapIII", interactive=FALSE)

## 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,
  interactive = FALSE)
```

Arguments

<code>qcdir</code>	[character] path/to/directory/with/QC/results containing name.imiss and name.het results as returned by <code>plink --missing</code> and <code>plink --het</code> .
<code>name</code>	[character] Prefix of PLINK files, i.e. name.bed, name.bim, name.fam, name.het and name.imiss.
<code>imissTh</code>	[double] Threshold for acceptable missing genotype rate in any individual; has to be proportion between (0,1)
<code>hetTh</code>	[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.
<code>interactive</code>	[logical] Should plots be shown interactively? When choosing this option, make sure you have X-forwarding/graphical interface available for interactive plotting. Alternatively, set <code>interactive=FALSE</code> and save the returned plot object (<code>p_het_imiss</code>) via <code>ggplot2::ggsave(p=p_het_imiss, other_arguments)</code> or <code>pdf(outfile) print(p_het_imiss) dev.off()</code> .

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 dataframe `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 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

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

## 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, verbose = 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().
verbose	[logical] If TRUE, progress info is printed to standard out.

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. $P(\text{IBD}=2) + 0.5 * P(\text{IBD}=1)$), PHE (Pairwise phenotypic code (1, 0, -1 = AA, AU, and UU pairs, respectively)), DST (IBS distance, i.e. $(\text{IBS}_2 + 0.5 * \text{IBS}_1) / (\text{IBS}_0 + \text{IBS}_1 + \text{IBS}_2)$), 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

```
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, path2plink = NULL,
  showPlinkOutput = TRUE, interactive = FALSE)
```

Arguments

<code>qcdir</code>	[character] /path/to/directory containing <code>name.sexcheck</code> as returned by <code>plink --check-sex</code> .
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> and <code>name.sexcheck</code> .
<code>maleTh</code>	[double] Threshold of X-chromosomal heterozygosity rate for males.
<code>femaleTh</code>	[double] Threshold of X-chromosomal heterozygosity rate for females.
<code>externalSex</code>	[data.frame, optional] with sample IDs [<code>externalSexID</code>] and sex [<code>externalSexSex</code>] to double check if external and PEDSEX data (often processed at different centers) match.
<code>fixMixup</code>	[logical] Should PEDSEX of individuals with mismatch between PEDSEX and Sex, with <code>Sex==SNPSEX</code> automatically corrected: this will directly change the <code>name.bim/.bed/.fam</code> files!
<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files; only required of <code>fixMixup==TRUE</code> . User needs writing permission to <code>indir</code> .
<code>externalFemale</code>	[integer/character] Identifier for 'female' in <code>externalSex</code> .
<code>externalMale</code>	[integer/character] Identifier for 'male' in <code>externalSex</code> .
<code>externalSexSex</code>	[character] Column identifier for column containing sex information in <code>externalSex</code> .

externalSexID	[character] Column identifier for column containing ID information in external-Sex.
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 accesible 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 <code>exec_wait('plink')</code> .
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 <code>ggplot2::ggsave(p=p_sexcheck, other_arguments)</code> or <code>pdf(outfile) print(p_sexcheck) dev.off()</code> .

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 and 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)`.

Examples

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

## 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`, v) `ancestry` containing a vector with sample IIDs failing the ancestry check based on the selected `europeanTh` and vi) `p_sampleQC`, a `ggplot2`-object 'containing' a sub-paneled plot with the QC-plots of `check_sex`, `check_het_and_miss`, `check_relatedness` and `check_ancestry`.

`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): samples IIDs in rows, columns are 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` containing a [data.frame] with samples that failed ancestry and QC checks: samples IIDs in rows, columns are `QC_fail` and `Ancestry_fail`, with `entries=0` if passing and `entries=1` if failing that check.

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, do.run_check_ancestry=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,
dont.check_ancestry = FALSE, do.run_check_ancestry = TRUE,
do.evaluate_check_ancestry = TRUE, prefixMergedDataset,
europeanTh = 1.5, refSamples = NULL, refColors = NULL,
refSamplesFile = NULL, refColorsFile = NULL, refSamplesIID = "IID",
refSamplesPop = "Pop", refColorsColor = "Color",
refColorsPop = "Pop", studyColor = "#2c7bb6", interactive = FALSE,
verbose = TRUE, path2plink = 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 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 do.evaluate_check_ancestry expects prefixMergeData.eigenvec. If these files are not present perIndividualQC will fail with missing file error. 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.
dont.check_ancestry	[logical] If TRUE, no ancestry check will be conducted; short for do.run_check_ancestry=FALSE and do.evaluate_check_ancestry=FALSE. Takes precedence over do.run_check_ancestry and do.evaluate_check_ancestry.
do.run_check_ancestry	[logical] If TRUE, run run_check_ancestry .
do.evaluate_check_ancestry	[logical] If TRUE, run evaluate_check_ancestry .
prefixMergedDataset	[character] Prefix of merged dataset (study and reference samples) used in plink -pca, resulting in prefixMergedDataset.eigenvec.
europeanTh	[double] Scaling factor of radius to be drawn around center of European reference samples, with study samples inside this radius considered to be of European descent and samples outside this radius of non-European descent. The radius is computed as the maximum Euclidean distance of European reference samples to the centre of European reference samples.
refSamples	[data.frame] Dataframe with sample identifiers [refSamplesIID] corresponding to IIDs in prefixMergedDataset.eigenvec and population identifier [refSamplesPop] corresponding to population IDs [refColorsPop] in refColorsfile/refColors. Either refSamples or refSamplesFile have to be specified.
refColors	[data.frame, optional] Dataframe with population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColorsFile) default colors are used.
refSamplesFile	[character] /path/to/File/with/reference samples. Needs columns with sample identifiers [refSamplesIID] corresponding to IIDs in prefixMergedDataset.eigenvec and population identifier [refSamplesPop] corresponding to population IDs [refColorsPop] in refColorsfile/refColors.
refColorsFile	[character, optional] /path/to/File/with/Population/Colors containing population IDs in column [refColorsPop] and corresponding colour-code for PCA plot in column [refColorsColor]. If not provided and is.null(refColors) default colors for are used.
refSamplesIID	[character] Column name of reference sample IDs in refSamples/refSamplesFile.
refSamplesPop	[character] Column name of reference sample population IDs in refSamples/refSamplesFile.
refColorsColor	[character] Column name of population colors in refColors/refColorsFile
refColorsPop	[character] Column name of reference sample population IDs in refColors/refColorsFile.

studyColor	[character] Colour to be used for study population.
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), iii) depicts all pair-wise IBD-estimates as histogram (if do.evaluate_check_relatedness is TRUE) and iv) creates a scatter plot of PC1 versus PC2 color-coded for samples of reference populations and study population (if do.check_ancestry is set to TRUE).
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 accesible 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 <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

Details

perIndividualQC wraps around the individual QC functions [check_sex](#), [check_het_and_miss](#), [check_relatedness](#) and [check_ancestry](#). 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 vector with sample IIDs failing the ancestry check based on europeanTh and ii) p_sampleQC, a ggplot2-object 'containing' a sub-paneled plot with the QC-plots of [check_sex](#), [check_het_and_miss](#), [check_relatedness](#) and [check_ancestry](#), 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
# In this examples, run_check* already conducted and outcome files present
# in qcdir, hence dont.check_* all set to FALSE
```

```

## Not run:
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, do.run_check_ancestry=FALSE)

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

## 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,
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 IDs 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 <code>check_snp_missingness</code> .
lmissTh	[double] Threshold for acceptable variant missing rate across samples.
do.check_hwe	[logical] If TRUE, run <code>check_hwe</code> .
hweTh	[double] Significance threshold for deviation from HWE.
do.check_maf	[logical] If TRUE, run <code>check_maf</code> .
macTh	[double] Threshold for minor allele cut cut-off, if both mafTh and macTh are specified, macTh is used ($macTh = mafTh \times 2 \times NrSamples$).
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 <code>interactive=FALSE</code> and save the returned plot object (<code>p_marker</code>) via <code>ggplot2::ggsave(p=p_marker, other_arguments)</code> or <code>pdf(outfile) print(p_marker) dev.off()</code> .
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 accesible as <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that PATH set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
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:
```

```
fail_markers <- perMarkerQC(indir=indir, qcdir=qcdir, name=name,
interactive=FALSE, verbose=TRUE, path2plink=path2plink)

## End(Not run)
```

relatednessFilter	<i>Remove related individuals while keeping maximum number of individuals</i>
-------------------	---

Description

filterRelatedness 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. filterRelatedness 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 parents-offspring trio, the offspring will be marked as fail, while the parents will be kept in the analysis).

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)
```

Arguments

- | | |
|----------------|--|
| relatedness | [data.frame] containing pair-wise relatedness estimates (in column [relatedness-Relatedness]) 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 <code>-genome</code> (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).

Value

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

`run_check_ancestry` *Run PLINK principal component analysis*

Description

Run `plink -pca` to calculate the principal components on merged genotypes of the study and reference dataset.

Usage

```
run_check_ancestry(indir, prefixMergedDataset, qcdir = indir,
  verbose = FALSE, path2plink = NULL, showPlinkOutput = TRUE)
```

Arguments

indir	[character] /path/to/directory containing the basic PLINK data files prefixMergedDataset.bim, prefixMergedDataset.fam and prefixMergedDataset.bed.
prefixMergedDataset	[character] Prefix of merged study and reference data files, i.e. prefixMergedDataset.bed, prefixMergedDataset.bim, prefixMergedDataset.fam.
qcdir	[character] /path/to/directory to save prefixMergedDataset.eigenvec as returned by plink -pca. 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 accesible 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 <code>exec_wait('plink')</code> .
showPlinkOutput	[logical] If TRUE, plink log and error messages are printed to standard out.

Details

Both, `run_check_ancestry` and its evaluation by `evaluate_check_ancestry` can simply be invoked by `check_ancestry`.

Examples

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

## 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, 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 accesible 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 <code>exec_wait('plink')</code> .
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()
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
run <- run_check_heterozygosity(indir=indir, qcdir=qcdir, name=name)

## 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, 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 <code>-missing</code> . 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 <code>exec_wait('plink')</code> .
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()
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
run <- run_check_missingness(indir=indir, qcdir=qcdir, name=name)

## 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 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,
  path2plink = NULL, showPlinkOutput = TRUE, verbose = FALSE)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> .
<code>qcdir</code>	[character] /path/to/directory to save <code>name.genome</code> as returned by <code>plink –genome</code> . User needs writing permission to <code>qcdir</code> . Per default <code>qcdir=indir</code> .
<code>highIBDTh</code>	[double] Threshold for acceptable proportion of IBD between pair of individuals; only pairwise relationship estimates larger than this threshold will be recorded.
<code>path2plink</code>	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. <code>plink</code> should be accessible as <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that <code>PATH</code> set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
<code>showPlinkOutput</code>	[logical] If TRUE, <code>plink</code> log and error messages are printed to standard out.
<code>verbose</code>	[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`.

Examples

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

## End(Not run)
```

run_check_sex	<i>Run PLINK sexcheck</i>
---------------	---------------------------

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, showPlinkOutput = TRUE)
```

Arguments

<code>indir</code>	[character] /path/to/directory containing the basic PLINK data files <code>name.bim</code> , <code>name.bed</code> , <code>name.fam</code> files.
<code>name</code>	[character] Prefix of PLINK files, i.e. <code>name.bed</code> , <code>name.bim</code> , <code>name.fam</code> .
<code>qcdir</code>	[character] /path/to/directory to save <code>name.sexcheck</code> as returned by <code>plink --check-sex</code> . User needs writing permission to <code>qcdir</code> . Per default <code>qcdir=indir</code> .
<code>verbose</code>	[logical] If TRUE, progress info is printed to standard out.
<code>path2plink</code>	[character] Absolute path to PLINK executable (https://www.cog-genomics.org/plink/1.9/) i.e. <code>plink</code> should be accesible as <code>path2plink -h</code> . The full name of the executable should be specified: for windows OS, this means <code>path/plink.exe</code> , for unix platforms this is <code>path/plink</code> . If not provided, assumed that <code>PATH</code> set-up works and PLINK will be found by <code>exec_wait('plink')</code> .
<code>showPlinkOutput</code>	[logical] If TRUE, <code>plink</code> 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()
# the following code is not run on package build, as the path2plink on the
# user system is not known.
## Not run:
run <- run_check_sex(indir=indir, qcdir=qcdir, name=name)

## End(Not run)
```

testNumerics	<i>Test lists for different properties of numerics</i>
--------------	--

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

check_ancestry, [3](#), [4](#), [16](#), [19](#), [25](#), [30](#), [35](#)
check_het_and_miss, [5](#), [6](#), [16](#), [20](#), [25](#), [30](#), [36](#),
[37](#)
check_hwe, [7](#), [16](#), [26](#), [32](#)
check_maf, [8](#), [16](#), [26](#), [32](#)
check_relatedness, [5](#), [10](#), [11](#), [16](#), [22](#), [25](#), [30](#),
[38](#)
check_sex, [12](#), [12](#), [13](#), [16](#), [24](#), [25](#), [30](#), [39](#)
check_snp_missingness, [14](#), [16](#), [26](#), [32](#)
checkPlink, [2](#)
cleanData, [15](#)

evaluate_check_ancestry, [18](#), [19](#), [29](#), [35](#)
evaluate_check_het_and_miss, [6](#), [20](#), [20](#),
[28](#), [36](#), [37](#)
evaluate_check_relatedness, [11](#), [21](#), [22](#),
[29](#), [38](#)
evaluate_check_sex, [13](#), [23](#), [24](#), [28](#), [39](#)
exec_wait, [2](#), [4](#), [6](#), [7](#), [9](#), [11](#), [13](#), [14](#), [17](#), [24](#), [30](#),
[32](#), [35–39](#)

overviewPerIndividualQC, [25](#)
overviewPerMarkerQC, [26](#)

perIndividualQC, [7–9](#), [14–16](#), [25](#), [26](#), [27](#), [31](#)
perMarkerQC, [16](#), [26](#), [31](#)

relatednessFilter, [33](#)
run_check_ancestry, [18](#), [19](#), [29](#), [34](#), [35](#)
run_check_heterozygosity, [6](#), [20](#), [28](#), [35](#),
[36](#), [37](#)
run_check_missingness, [6](#), [20](#), [28](#), [36](#), [36](#), [37](#)
run_check_relatedness, [11](#), [21](#), [22](#), [29](#), [37](#),
[38](#)
run_check_sex, [13](#), [23](#), [24](#), [28](#), [39](#), [39](#)

testNumerics, [40](#)

upset, [25](#), [26](#)