

Package ‘SNPtools’

June 28, 2025

Title S4 Tools for Reading and Organizing Genetic Data

Version 0.1.0

Maintainer Vinícius Junqueira <junqueiravinicius@hotmail.com>

Description Provides S4-based structures to encapsulate the import, organization, and processing of genetic data from PLINK and FImpute files, with customizable arguments. Includes tools for combining SNP panels, summarizing genotype data, and facilitating downstream quality control and analysis workflows.

Depends R (>= 4.1.0),
snpStats,
tidyverse

Imports methods,
data.table,
fQC

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Encoding UTF-8

License GPL-3

RoxygenNote 7.3.2

Contents

check.sample.call.rate	2
combinarSNPData	2
FImputeRunner	3
getGeno	4
importAllGenos	4
import_geno_list	5
qcSamples	6
qcSNPs	7
qc_header	8
runFImpute	8

saveFImpute	9
saveFImputeRaw	10
summary,SNPDataLong-method	10

Index	11
--------------	-----------

check.sample.call.rate	<i>Check Sample Call Rate</i>
------------------------	-------------------------------

Description

Identifies samples with call rate below a given threshold.

Usage

check.sample.call.rate(sample.summary, min.call.rate)

Arguments

- sample.summary A data frame with a "Call.rate" column for each sample.
- min.call.rate Minimum acceptable call rate (between 0 and 1).

Value

A character vector with the names of samples to remove.

combinarSNPData	<i>Combine multiple SNPDataLong objects</i>
-----------------	---

Description

This function merges a list of SNPDataLong objects, typically representing different SNP panels or datasets, into a single unified SNPDataLong object. It ensures that all genotype matrices have the same set of SNPs (filling missing SNPs with NA), and merges the marker map information while removing duplicate SNP entries.

Usage

combinarSNPData(lista)

Arguments

- lista A list of SNPDataLong objects to be combined.

Details

The function performs the following steps internally:

1. Computes the union of all SNPs across input objects.
2. Fills missing SNP columns in each genotype matrix with NA-coded columns.
3. Combines genotype matrices by rows (individuals).
4. Merges marker maps, removing duplicates (retaining the first occurrence).
5. Creates and returns a new SNPDataLong object with the combined data.

Value

A single SNPDataLong object containing the combined genotype matrix, merged map, and a concatenated path string.

Examples

```
## Not run:
# Example usage:
combined_data <- combinarSNPData(list(snp_data1, snp_data2, snp_data3))

## End(Not run)
```

FImputeRunner

Build FImputeRunner object

Description

A convenience function to construct a ‘FImputeRunner’ object from basic inputs.

Usage

```
FImputeRunner(geno, map, path, exec_path = "FImpute3", name = "data")
```

Arguments

geno	A SnpMatrix object.
map	A data.frame with SNP metadata (columns: Name, Chromosome, Position).
path	A character string indicating the directory to save FImpute files.
exec_path	Path to the FImpute executable (default = "FImpute3").
name	Name for the dataset (used internally, default = "gen_data").

Value

An object of class ‘FImputeRunner’.

getGeno	<i>Flexible and efficient genotype file reading with autodetection using fread</i>
---------	--

Description

This generic and method allow flexible import of SNP genotype data from Illumina FinalReport files, supporting fast initial column detection using `data.table::fread`, followed by full genotype matrix construction via `snpStats::read.snps.long`.

Usage

```
getGeno(...)
```

Arguments

path	Path to the directory containing FinalReport.txt
fields	A list specifying column indices for sample, SNP, allele1, allele2, and confidence
codes	A character vector with allele codes (e.g., <code>c("A", "B")</code>)
threshold	Confidence threshold for genotype calling
sep	Field separator used in the files
skip	Number of lines to skip at the start of the file
verbose	Logical; if TRUE, displays progress messages
every	Frequency of progress update (number of SNPs)

Value

An `SNPDataLong` object containing the genotype matrix and map, or NULL if an error occurs

importAllGenos	<i>Import and combine multiple genotype configurations</i>
----------------	--

Description

This generic and method import genotype data from multiple configurations defined in an `SNPImportList` object, then combine them into a single unified `SNPDataLong` object.

Usage

```
importAllGenos(object)
```

Arguments

object An object of class SNPImportList containing import configurations

Value

A single combined SNPDataLong object

import_geno_list	<i>Import multiple genotype datasets from a list of configurations</i>
------------------	--

Description

This function iterates over a list of configuration lists (each specifying parameters such as path, fields, separators, etc.), imports each genotype dataset using `getGeno()`, and then combines them into a single SNPDataLong object.

Usage

```
import_geno_list(config_list)
```

Arguments

config_list A list of configuration lists. Each element must include at least path and fields. Optional elements include codes, threshold, sep, skip, and verbose.

Value

A unified SNPDataLong object containing combined genotype data from all configurations.

Examples

```
## Not run:
configs <- list(
  list(path = "panel1", fields = list(sample = 2, snp = 1, allele1 = 7, allele2 = 8, confidence = 9)),
  list(path = "panel2", fields = list(sample = 2, snp = 1, allele1 = 7, allele2 = 8, confidence = 9), threshold = 0.10
)
combined_data <- import_geno_list(configs)

## End(Not run)
```

qcSamples	<i>Quality control on samples</i>
-----------	-----------------------------------

Description

Applies quality control (QC) procedures to samples in a ‘SNPDataLong’ object, based on heterozygosity and call rate thresholds.

Usage

```
qcSamples(x, ...)

## S4 method for signature 'SNPDataLong'
qcSamples(
  x,
  heterozygosity = NULL,
  smp_cr = NULL,
  action = c("report", "filter", "both")
)
```

Arguments

x	An object of class ‘SNPDataLong’.
heterozygosity	A numeric threshold or range for heterozygosity. Samples outside this threshold are removed.
smp_cr	Minimum acceptable sample call rate (between 0 and 1). Samples below this value are removed.
action	Character string indicating the action to perform. One of: - “report”: only returns a list of samples to remove and those kept; - “filter”: returns a filtered object without reporting; - “both”: performs filtering and returns the filtered object.

Value

Depending on the ‘action’ argument: - “report”: returns a list with removed and kept samples; - “filter”: returns a new ‘SNPDataLong’ object with filtered genotypes; - “both”: returns a list with: - ‘filtered’: the filtered ‘SNPDataLong’ object; - ‘report’: a list of removed and kept samples.

Description

Permite análise de qualidade genotípica com critérios definidos pelo usuário.

Usage

```
qcSNPs(x, ...)
```

Arguments

x	Objeto da classe SNPDataLong
missing_ind	Proporção máxima de dados faltantes permitida por indivíduo (opcional)
missing_snp	Proporção máxima de dados faltantes permitida por SNP (opcional)
min_snp_cr	Mínimo call rate aceitável para SNPs
min_maf	Frequência alélica mínima permitida para SNPs (opcional)
hwe	Identifica SNPs com desvio de HWE
snp_mono	Identifica SNPs monomórficos
remove_chr	Descarta SNPs localizados nos cromossomos listados
action	"report", "filter" ou "both"

Value

Dependendo do argumento action, retorna: - "report": lista com SNPs removidos por critério; - "filter": objeto SNPDataLong com SNPs filtrados; - "both": lista contendo o objeto filtrado e o relatório.

Examples

```
## Not run:
set.seed(123)
mat <- matrix(sample(c(0, 1, 2, NA), 100, replace = TRUE, prob = c(0.4, 0.4, 0.15, 0.05)),
              nrow = 10, ncol = 10)
colnames(mat) <- paste0("snp", 1:10)
rownames(mat) <- paste0("ind", 1:10)
map <- data.frame(Name = colnames(mat), Chrom = 1, Position = 1:10)
x <- new("SNPDataLong", geno = mat, map = map)

qcSNPs(x, min_snp_cr = 0.8, min_maf = 0.05, snp_mono = TRUE, action = "report")
## End(Not run)
```

qc_header	<i>Formatted header message</i>
-----------	---------------------------------

Description

Prints a formatted message with a border for section titles in the console.

Usage

```
qc_header(title)
```

Arguments

title	Character string to be printed inside the header box.
-------	---

Value

No return value. Used for side effects (message).

Examples

```
qc_header("Quality Control on Samples")
```

runFImpute	<i>Run FImpute from a FImputeRunner object</i>
------------	--

Description

This function runs the external FImpute software using a 'FImputeRunner' object, ensuring that all required input files are present and the results are imported.

Usage

```
runFImpute(object)

## S4 method for signature 'FImputeRunner'
runFImpute(object)
```

Arguments

object	An object of class 'FImputeRunner'
--------	------------------------------------

Value

An updated 'FImputeRunner' object with the 'results' slot populated.

saveFImpute

*Save genotype and map files in FImpute format***Description**

S4 method to export genotype ('.gen'), map ('.map'), and parameter ('data.par') files compatible with the [FImpute](https://www.aps.uoguelph.ca/~msargol/fimpute/) software.

Usage

```
saveFImpute(object, ...)

## S4 method for signature 'FImputeExport'
saveFImpute(object, ...)

## S4 method for signature 'SNPDataLong'
saveFImpute(object, path = "fimpute_run")
```

Arguments

object	An object of class 'FImputeExport' or 'SNPDataLong'.
...	Further arguments passed to methods.
path	Character. Output directory where files will be written (only for 'SNPDataLong' method; default = "fimpute_run").

Value

No return value. Files are saved to disk.

Examples

```
## Not run:
if (requireNamespace("snpStats", quietly = TRUE)) {
  mat <- matrix(sample(c(0L, 1L, 2L), 50, replace = TRUE), nrow = 5)
  colnames(mat) <- paste0("snp", 1:10)
  rownames(mat) <- paste0("ind", 1:5)

  sm <- new("SnpMatrix", data = as.raw(mat))
  map <- data.frame(Name = colnames(mat), Chromosome = 1, Position = 1:10)
  x <- new("SNPDataLong", geno = sm, map = map)

  saveFImpute(x, path = tempdir())
}

## End(Not run)
```

saveFImputeRaw	<i>Export genotypes and map using basic arguments</i>
----------------	---

Description

Convenience function to export FImpute files directly from a ‘SnpMatrix’ and map ‘data.frame’.

Usage

```
saveFImputeRaw(geno, map, path)
```

Arguments

geno	A ‘SnpMatrix’ object (from the ‘snpStats’ package).
map	A ‘data.frame’ with columns ‘Name’, ‘Chromosome’, and ‘Position’.
path	Path where the files will be saved.

Value

No return value. Files are saved to disk.

summary,SNPDataLong-method	<i>Summary for SNPDataLong objects</i>
----------------------------	--

Description

Provides a detailed summary of an SNPDataLong object, including sample and SNP counts, proportion of missing data, and SNP distribution by chromosome if mapping information is available.

Usage

```
## S4 method for signature 'SNPDataLong'
summary(object, ...)
```

Arguments

object	An object of class SNPDataLong.
--------	---------------------------------

Value

Prints a summary to the console. Returns NULL (invisible).

Index

`check.sample.call.rate`, [2](#)
`combinarSNPData`, [2](#)

`FImputeRunner`, [3](#)

`getGeno`, [4](#)

`import_geno_list`, [5](#)
`importAllGenos`, [4](#)

`qc_header`, [8](#)
`qcSamples`, [6](#)
`qcSamples`, `SNPDataLong`-method
 (`qcSamples`), [6](#)
`qcSNPs`, [7](#)

`runFImpute`, [8](#)
`runFImpute`, `FImputeRunner`-method
 (`runFImpute`), [8](#)

`saveFImpute`, [9](#)
`saveFImpute`, `FImputeExport`-method
 (`saveFImpute`), [9](#)
`saveFImpute`, `SNPDataLong`-method
 (`saveFImpute`), [9](#)
`saveFImputeRaw`, [10](#)
`summary`, `SNPDataLong`-method, [10](#)