

Package ‘GWASinspector’

September 25, 2019

Type Package

Title Comprehensive and Easy to Use Quality Control of GWAS Results

Version 1.2

Date 2019-09-24

Author Alireza Ani [aut, cre],
Peter J. van der Most [aut],
Ahmad Vaez [aut],
Ilja M. Nolte [aut]

Maintainer Alireza Ani <a.ani@umcg.nl>

Depends R (>= 3.2)

Imports ini (>= 0.3), futile.logger (>= 1.4), data.table (>= 1.10),
hash (>= 2.2), tools (>= 3.0), ggplot2 (>= 3.0), knitr (>=
1.1), rmarkdown (>= 0.9), gridExtra, grid, RSQLite, kableExtra
(>= 0.8)

Suggests xlsx (>= 0.5), parallel (>= 3.0)

VignetteBuilder knitr

URL <http://GWASinspector.com>

Description When evaluating the results of a genome-wide association study (GWAS), it is important to perform a quality control to ensure that the results are valid, complete, correctly formatted, and, in case of meta-analysis, consistent with other studies that have applied the same analysis. This package was developed to facilitate and streamline this process and provide the user with a comprehensive report.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-09-25 14:30:02 UTC

R topics documented:

check.database	2
compare.GWASs	3
find.variants	3
get.config	4
get.headerTranslation	5
GWASinspector	5
inspect	6
inspect.example	7
man.plot	7
reformat.columns	8
system.check	9

Index	10
--------------	-----------

check.database	<i>Check the data in the reference database</i>
----------------	---

Description

This function displays the summary of the database, including how many tables are in the database file, number of data rows for each data table and the first row of each table.

Usage

```
check.database(path)
```

Arguments

path character. full path to the database file (*.sqlite)

Value

This function returns a data table including the summary of the specified database. This is necessary to check the consistency and validity of an unknown or new database file.

Note

First column include the names of the tables in database
 Second column is the number of rows in each table
 Next columns are the first row of each table

Examples

```
check.database(system.file("extdata", "sample_db.sqlite", package = "GWASinspector"))
```

compare.GWASs	<i>Compare GWAS result files</i>
---------------	----------------------------------

Description

This function compares the key metrics of previously inspected files. This allows the user to check that the results of these studies are comparable (important when running a meta-analysis) and that there are no significant anomalies.

Usage

```
compare.GWASs(input.file.list, output.path)
```

Arguments

input.file.list	list, full path of the RDS object files. Note that inspect() only produces such files if the object_file parameter is set to TRUE in the configuration file.
output.path	character, full path to the folder where output files should be saved.

Value

Key metrics report of previously inspected files are generated and saved in the specified output folder.

find.variants	<i>Search variants in the reference database</i>
---------------	--

Description

This function searches for a list of variants inside the reference database. rsID or hID of the variants can be used for the lookup.

Usage

```
find.variants(db.path, input.vector, column.name = "hID",
              reorder = FALSE, output.path = NULL)
```

Arguments

db.path	character, full path to the database file (*.sqlite).
input.vector	character, list of variant IDs to be searched in the database
column.name	character, this can be either hID (CHR:POS:type) or ID (rsID). default: hID
reorder	logical, if TRUE output is reordered like the input vector
output.path	character, full path to the excel file.

Value

A data table is returned after searching for input variants in the database.

Examples

```
find.variants(system.file("extdata", "sample_db.sqlite", package = "GWASInspector"),  
              c('rs568405545', 'rs367896724') , column.name = 'ID')
```

```
find.variants(system.file("extdata", "sample_db.sqlite", package = "GWASInspector"),  
              c('1:10506:1', '1:10511:1') , column.name = 'hID')
```

get.config

Save a sample configuration file

Description

Save a sample configuration file for running GWASInspector package. This templates should be edited and then used for running the QC. User can save multiple copies to be used for different sets of files. Default name is *config.ini*, which can be changed while saving the file or afterwards.

Usage

```
get.config(dir.path = NULL)
```

Arguments

dir.path Path to the folder for saving a sample configuration file.

Value

Saves a sample configuration file (config.ini) in the specified folder.

Examples

```
get.config(tempdir())
```

get.headerTranslation *Save a sample header translation table file*

Description

This template file is used to translate a dataset's column names (the header) into the standard names used by GWASInspector. The file contains a two-column table, with the left column containing the standard column-names and the right the alternatives. Both the standard and alternative columns must be fully capitalized. This is a text file which includes most common variable/header names and can be edited according to user specifications. The default name of this file is `*alt_headers.txt*`. configuration file should be edited if this name is changed by user (`**header_translations**` property).

Usage

```
get.headerTranslation(dir.path = NULL)
```

Arguments

`dir.path` Path to the folder for saving a header-translation table file.

Value

Saves a sample header-translation table file in the specified folder.

Examples

```
get.headerTranslation(tempdir())
```

GWASInspector

Comprehensive and Easy to Use Quality Control of GWAS Results

Description

When evaluating the results of a genome-wide association study (GWAS), it is important to perform a quality control to ensure that the results are valid, complete, correctly formatted, and, in case of meta-analysis, consistent with other studies in the same analysis. This package was developed to facilitate and streamline this process and provide the user with a comprehensive report.

Details

Check out our website for more help and support <http://GWASInspector.com>.

Functions

This package includes functions for checking prerequisite libraries on the local machine, setting up QC parameters and running the QC on GWAS result files.

`system.check()` Some R packages or system functions are not necessary for running the algorithm; but, it is suggested to install them to benefit full functionality. Using this function to get a brief information about whether these utilities are present and accessible. After running the command you will see a table indicating package names and their version.

`inspect()` Main function for running the algorithm on GWAS data files.

`check.database()` Displays the summary of a reference database, including how many tables are in the database file, number of data rows for each data table and the first row of each table

`find.variants()` Search for a list of variants in the allele reference panels.

`get.config()` Save a sample configuration file on your computer.

`get.headerTranslation()` Save a sample column header translation table file.

`compare.GWASs()` Compares result files from different analyses. So, there is no need to re-run the analysis on a result file again.

`man.plot()` Generates the Manhattan plot from a result file. This function has many features that are described in the package tutorial.

`inspect.example()` This function runs the algorithm on a fabricated GWAS result file. User should only set the output folder for saving the generated files. The input file and reference dataset are embedded in the package.

<code>inspect</code>	<i>Runs the QC algorithm</i>
----------------------	------------------------------

Description

This is the main function for running the algorithm on GWAS result files.

Usage

```
inspect(config.file = NULL, user.verification = FALSE,
        test.run = FALSE)
```

Arguments

<code>config.file</code>	character. Path to a configuration (.ini) file to configurate the QC. For a sample configuration file, see <code>get.config()</code> . For details on how to edit the config file, see the tutorial.
<code>user.verification</code>	logical. If TRUE, the algorithm will pause and ask the user to verify that it has selected the correct input files for QC. Default vaule is FALSE.
<code>test.run</code>	logical. if TRUE, only the first 1000 lines of each data file are loaded and analysed, and no plots or final dataset is produced.

Value

QC reports from running the algorithm on a single or a series of GWAS result files are generated and saved.

inspect.example	<i>Runs an example QC</i>
-----------------	---------------------------

Description

This function runs the QC algorithm on a fabricated GWAS result file.

Usage

```
inspect.example(result.dir = NULL)
```

Arguments

result.dir character. Path to the output folder for saving QC result files

Value

QC reports from running the algorithm on a sample GWAS file are generated and saved in the specified folder.

Examples

```
inspect.example(tempdir())
```

man.plot	<i>Creates the Manhattan plot</i>
----------	-----------------------------------

Description

A function to generate Manhattan plots.

Usage

```
man.plot(dataset, chr, pvalue, position, fileName,  
plot.title = "Manhattan Plot", plot.subtitle = "",  
p.threshold = 0.01, sig.threshold.log = -log10(5 * 10^-8),  
beta = NULL, std.error = NULL, check.columns = TRUE)
```

Arguments

dataset	Data frame or data table containing the below columns
chr	Name of chromosome column
pvalue	Name of P-value column
position	Name of position column
fileName	Full name and path of file to be saved (file extension should be 'png'). e.g. "c:/users/researcher/study/man_plot.png"
plot.title	Title of the plot, default value is 'Manhattan plot'
plot.subtitle	Subtitle of the plot
p.threshold	Threshold for plotting variants (i.e. p-values > 0.01 will not be plotted). Setting a higher threshold will significantly increase plotting time
sig.threshold.log	The -log10 transformed significance threshold, used for plotting a threshold line (e.g. 8 = 10 ⁻⁸)
beta	(optional) Name of the effect-size column
std.error	(optional) Name of the standard error column
check.columns	Whether to check input columns for invalid values

Value

Generates and saves a Manhattan plot for the provided data.

Examples

```
input.data = read.table(gzfile(system.file("extdata", "sample.txt.gz", package = "GWASinspector")),
                        header = TRUE,
                        stringsAsFactors = FALSE,
                        fill = TRUE)
tmpPlotFile = paste(tempfile(), 'png', sep = '.')
man.plot(dataset = input.data, chr = 'CHR', pvalue = 'PVALUE', position = 'POSITION',
         plot.title = 'Manhattan plot', plot.subtitle = 'This data is fabricated!',
         fileName = tmpPlotFile , p.threshold = '0.5')
```

reformat.columns

Unifies column separators in all data rows

Description

It is common that column separator is different between the header and the rest of the file. This can be problematic for reading the file correctly and should be treated before loading the data. This function is based on Rtools and **awk** function to read and save a new file with equal column separators between all rows of data.

Usage

```
reformat.columns(input.file, output.file = NULL, sep = "\t")
```

Arguments

input.file	path to the input file
output.file	path of the outfile path; if not defined , a tag is added to input file
sep	what character to be used add separator; tab character is default

Value

A new text file is generated from the old one with a consistent column separator character.

system.check	<i>Checks which required and optional packages are available</i>
--------------	--

Description

This functions checks the availability of all required and optional packages for GWASInspector. The optional packages are not mandatory for running the algorithm; but will add useful functionalities.

Usage

```
system.check()
```

Value

System information and required functionalities for the QC algorithm are checked and reported as a data frame.

Examples

```
system.check()
```

Index

`check.database`, 2
`compare.GWASs`, 3

`find.variants`, 3

`get.config`, 4
`get.headerTranslation`, 5
`GWASInspector`, 5
`GWASInspector-package (GWASInspector)`, 5

`inspect`, 6
`inspect.example`, 7

`man.plot`, 7

`reformat.columns`, 8

`system.check`, 9