Package 'GWASinspector'

April 27, 2022

Type Package

Title Comprehensive and Easy to Use Quality Control of GWAS Results

Version 1.5.7.2

Date 2022-04-27

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 (>= 1.12.8),
hash (>= 2.2), tools (>= 3.0), ggplot2 (>= 3.0), knitr (>=
1.1), rmarkdown (>= 0.9), gridExtra, grid, RSQLite, kableExtra
(>= 0.8), methods, R.utils

Suggests xlsx (>= 0.5)

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

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-27 17:30:08 UTC

R topics documented:

---

compare.GWASs   *Compare summary statistics of GWAS result files*

Description

This function compares the key metrics of previously inspected files. This allows the user to check if 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 Inspector result files. This file is in RDS format and will be generated for each GWAS result file during the inspection algorithm.

output.path   character, full path to the folder where output files should be saved.

Value

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

---

demo.inspector   *Runs an example QC*

Description

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

Usage

```
demo.inspector(result.dir)
```

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.

---

get.config   *Copies the template configuration file to the local machine*

Description

This templates should be edited and then used for setting up and running the QC pipeline. Default filename is **config.ini**.

Usage

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

Arguments

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

Value

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

---

get.headerTranslation   *Copies the template header translation table to the local machine*

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 filename is **alt_headers.txt**.

Usage

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

Arguments

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

Value

Copies a sample header-translation table in the specified folder.

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

[setup.inspector](#) This function Imports a QC-configuration file into R by generating a new instance of [Inspector](#) class.

[run.inspector](#) This is the main function for running the algorithm on a set of GWAS result files.

[result.inspector](#) This function displays a brief report about the results of running the Inspector algorithm on a set of GWAS result files.

demo.inspector 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.

system.check Checks if required and optional packages are installed on the system. Although the optional packages do not contribute to the QC itself, having them available will allow for Excel and HTML formatted report files, which are easier to read and interpret.

get.config Copies the template configuration file to the local machine.

get.headerTranslation Copies the template configuration file to the local machine.

compare.GWASs Generates reports and plots for comparing the summary statistics of GWAS result files that are previously inspected with this package.

manhattan.plot Generates the Manhattan plot from a GWAS result file. This function has many features that are described in the package tutorial.

Note

GWASinspector uses the S4 object system of R to conduct the QC. The QC is configured using an configuration (ini) file (check get.config), which is imported into R through setup.inspector and turns into an object of the Inspector class. To perform the QC, process the object with run.inspector. A quick scan of the results can be performed via result.inspector, but the primary outcome of the QC are the report files and graphs generated by run.inspector.

The main product of the QC is the extensive log file (in Excel/HTML format, depending on your settings)

Inspector-class	*An S4 class to represent the methods and parameters for inspecting a list of GWAS study result files.*

Description

An object of this class is created by setup.inspector function. Each section of the configuration file is represented as a list of attributes in this object.

Slots

paths A list of parameters which indicate **Paths** section from configuration file.

supplementaryFiles A list of parameters which indicate **supplementaryFiles** section from configuration file.

input_parameters A list of parameters which indicate **input_parameters** section from configuration file.

output_parameters A list of parameters which indicate **output_parameters** section from configuration file.

remove_chromosomes A list of parameters which indicate **remove_chromosomes** section from configuration file.

plot_specs A list of parameters which indicate **plot_specs** section from configuration file.

filters A list of parameters which indicate **filters** section from configuration file.

debug A list of parameters which indicate **debug** section from configuration file.

input_files A list of files that will be inspected during the run.

created_at The time that object was created.

start_time The time that object was run.

end_time The time that run was finished.

StudyList An object of [StudyList](#) class.

---

manhattan.plot   *Creates the Manhattan plot*

Description

A function to generate Manhattan plots.

Usage

```
manhattan.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.

---

result.inspector	*Displays a brief report after running the QC pipeline*

Description

This function displays a brief report about the results of running the Inspector algorithm on a set of GWAS result files. The full report including plots, cleaned files and summary statistics are generated and saved in the output folder during the algorithm run.

Usage

```
result.inspector(inspector)
```

Arguments

inspector	An instance of [Inspector](#) class. Check [setup.inspector](#) for more details.

Value

A data.table containing a brief report about the results.

---

run.inspector   *Runs the QC pipeline on a set of GWAS result files*

Description

This is the main function of the package for running the QC algorithm on a set of GWAS result files. It requires an object of class Inspector which should be created by `setup.inspector`. Check the package vignette and tutorial for more details on this topic.

Usage

```
run.inspector(inspector, verbose = TRUE, test.run = FALSE)
```

Arguments

inspector	An instance of Inspector class. Check `setup.inspector` for more details.
verbose	logical. If FALSE, no messages will show up in the terminal and are only saved in the log file.
test.run	logical. If TRUE, only the first 1000 lines of each data file are loaded and analyzed; plots and saving the cleaned output dataset are skipped. Default value is FALSE.

Value

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

---

setup.inspector   *Importing a configuration file into R for setting the pipeline parameters*

Description

To run a QC in GWASinspector, copy a template configuration file to your machine using the `get.config` command at first, and edit it to suit your requirements. Next, use the `setup.inspector` function to check the configuration file and import it into R. This will create an object of the inspector class, which can then be processed using `run.inspector`.

Usage

```
setup.inspector(config.file, validate = TRUE)
```

Arguments

config.file	character. Path to a configuration (.ini) file. For a sample configuration file, see `get.config`.
validate	logical. Whether to validate the object.

Value

returns a new instance of [Inspector](#) class.

Study-class *An S4 class to represent an inspected GWAS study result file.*

Description

This class is embedded in the [StudyList](#) class and should not be initiated separately.

Slots

File A list representing GWAS result file specifications

Counts A list representing different variant counts from the GWAS result file.

Correlations A list representing different allele frequency and P-value correlations in the GWAS result file.

Statistics A list representing summary statistics from the GWAS result file.

Successful_run A logical value indicating whether the run was successful or not.

starttime The time that file inspection started.

endtime The time that file inspection ended.

StudyList-class *An S4 class to represent a list of inspected GWAS study result files.*

Description

This class is embedded in the [Inspector](#) class and should not be initiated separately.

Slots

studyList A list of GWAS study result files. Each member of this list is of class [Study](#).

studyCount A numeric value indicating how many items of class [Study](#) are included.

system.check	*Checks which required and optional packages are available*

Description

Checks if required and optional packages are installed on the system. Although the optional packages do not contribute to the QC itself, having them available will allow for Excel and HTML formatted log files, which are easier to read and interpret.

Usage

```
system.check()
```

Value

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

Index