

Package ‘BiasCorrector’

March 29, 2021

Title A GUI to Correct Measurement Bias in DNA Methylation Analyses

Version 0.1.4

Date 2021-03-29

Description A GUI to correct measurement bias in DNA methylation analyses. The 'BiasCorrector' package just wraps the functions implemented in the 'R' package 'rBiasCorrection' into a shiny web application in order to make them more easily accessible.

License GPL-3

URL <https://github.com/kapsner/BiasCorrector>

BugReports <https://github.com/kapsner/BiasCorrector/issues>

Depends R (>= 2.10)

Imports data.table, DT, magrittr, rBiasCorrection, shiny, shinydashboard, shinyjs

Suggests lintr, testthat

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Lorenz A. Kapsner [cre, aut, cph]
(<<https://orcid.org/0000-0003-1866-860X>>),
Evgeny A. Moskalev [aut]

Maintainer Lorenz A. Kapsner <lorenz.kapsner@gmail.com>

Repository CRAN

Date/Publication 2021-03-29 12:02:06 UTC

R topics documented:

launch_app	2
module_calibrationfile_server	3
module_calibrationfile_ui	4
module_correctedplots_server	5

module_correctedplots_ui	6
module_correctedstatistics_ui	6
module_correctedstats_server	7
module_experimentalfile_server	8
module_experimentalfile_ui	9
module_fileupload_server	10
module_fileupload_ui	11
module_info_server	12
module_info_ui	13
module_log_server	13
module_log_ui	14
module_modelselection_server	15
module_modelselection_ui	16
module_plotting_server	17
module_plotting_ui	18
module_results_server	18
module_results_ui	19
module_settings_server	20
module_settings_ui	21
module_statistics_server	22
module_statistics_ui	23

Index	24
--------------	-----------

launch_app	<i>Launch BiasCorrector</i>
------------	-----------------------------

Description

Launch BiasCorrector

Usage

```
launch_app(
  port = 3838,
  plotdir = "plots",
  csvdir = "csv",
  logfilename = "biascorrector.log",
  maxfilesize = 100,
  parallel = TRUE
)
```

Arguments

port	The port, BiasCorrector is running on (default: 3838)
plotdir	A character string. Defaults to 'plots'. This directory is being created inside tempdir.

csvdir	A character string. Defaults to 'csv'. This directory is being created inside tempdir.
logfilename	A character string. The name of the logfile (default = biascorrector.log).
maxfilesize	A positive integer. The maximum file size allowed for upload.
parallel	A boolean. If TRUE (default), initializing 'future::plan("multiprocess")' before running the code.

Value

The function returns the BiasCorrector shiny application.

Examples

```
if (interactive()) {
  launch_app()
}
```

```
module_calibrationfile_server
      module_calibrationfile_server
```

Description

module_calibrationfile_server

Usage

```
module_calibrationfile_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: input_re = reactive(input)
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_calibrationfile_server,  
    "moduleCalibrationFile",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_calibrationfile_ui

module_calibrationfile_ui

Description

module_calibrationfile_ui

Usage

```
module_calibrationfile_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "calibration",  
      module_calibrationfile_ui(  
        "moduleCalibrationFile"  
      )  
    )  
  )  
}
```

module_correctedplots_server
module_correctedplots_server

Description

module_correctedplots_server

Usage

```
module_correctedplots_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_correctedplots_server,  
    "moduleCorrectedPlots",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_correctedplots_ui
module_correctedplots_ui

Description

module_correctedplots_ui

Usage

```
module_correctedplots_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "correctedplots",  
      module_correctedplots_ui(  
        "moduleCorrectedPlots"  
      )  
    )  
  )  
}
```

module_correctedstatistics_ui
module_correctedstatistics_ui

Description

module_correctedstatistics_ui

Usage

```
module_correctedstatistics_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "correctedstats",  
      module_correctedstatistics_ui(  
        "moduleCorrectedStats"  
      )  
    )  
  )  
}
```

```
module_correctedstats_server  
                          module_correctedstats_server
```

Description

module_correctedstats_server

Usage

```
module_correctedstats_server(input, output, session, rv, input_re)
```

Arguments

input Shiny server input object
output Shiny server output object
session Shiny session object
rv The global 'reactiveValues()' object, defined in server.R
input_re The Shiny server input object, wrapped into a reactive expression: input_re = reactive(input)

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_correctedstats_server,  
    "moduleCorrectedStats",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_experimentalfile_server
module_experimentalfile_server

Description

module_experimentalfile_server

Usage

```
module_experimentalfile_server(input, output, session, rv, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_experimentalfile_server,  
    "moduleExperimentalFile",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

```
module_experimentalfile_ui  
  module_experimentalfile_ui
```

Description

module_experimentalfile_ui

Usage

```
module_experimentalfile_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "experimental",  
      module_experimentalfile_ui(  
        "moduleExperimentalFile"  
      )  
    )  
  )  
}
```

```
module_fileupload_server
      module_fileupload_server
```

Description

module_fileupload_server

Usage

```
module_fileupload_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {
  rv <- list()
  logfilename <- paste0(tempdir(), "/log.txt")
  shiny::callModule(
    module_fileupload_server,
    "moduleEileUpload",
    rv = rv,
    logfilename = logfilename
  )
}
```

module_fileupload_ui *module_fileupload_ui*

Description

module_fileupload_ui

Usage

```
module_fileupload_ui(id, ...)
```

Arguments

id	A character. The identifier of the shiny object
...	Further arguments, such as 'maxfilesize'

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "fileupload",  
      module_fileupload_ui(  
        "moduleFileUpload",  
        maxfilesize = maxfilesize  
      )  
    )  
  )  
}
```

module_info_server *module_info_server*

Description

module_info_server

Usage

```
module_info_server(input, output, session, rv, input_re)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_info_server,  
    "moduleInfo",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_info_ui	<i>module_info_ui</i>
----------------	-----------------------

Description

module_info_ui

Usage

```
module_info_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {
  shinydashboard::tabItems(
    shinydashboard::tabItem(
      tabName = "info",
      module_info_ui(
        "moduleInfo"
      )
    )
  )
}
```

module_log_server	<i>module_log_server</i>
-------------------	--------------------------

Description

module_log_server

Usage

```
module_log_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>
...	Further arguments, such as 'logfile', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfile <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_log_server,  
    "moduleLog",  
    rv = rv,  
    logfile = logfile  
  )  
}
```

module_log_ui

module_log_ui

Description

module_log_ui

Usage

```
module_log_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {
  shinydashboard::tabItems(
    shinydashboard::tabItem(
      tabName = "log",
      module_log_ui(
        "moduleLog"
      )
    )
  )
}
```

module_modelselection_server

module_modelselection_server

Description

module_modelselection_server

Usage

```
module_modelselection_server(input, output, session, rv, input_re)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: input_re = reactive(input)

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_modelselection_server,  
    "moduleModelSelection",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_modelselection_ui

module_modelselection_ui

Description

module_modelselection_ui

Usage

```
module_modelselection_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "modelselection",  
      module_modelselection_ui(  
        "moduleModelSelection"  
      )  
    )  
  )  
}
```



```

    )
  )
}

```

```

module_plotting_server
      module_plotting_server

```

Description

module_plotting_server

Usage

```
module_plotting_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```

if (interactive()) {
  rv <- list()
  logfilename <- paste0(tempdir(), "/log.txt")
  shiny::callModule(
    module_plotting_server,
    "modulePlotting",
    rv = rv,
    logfilename = logfilename
  )
}

```

module_plotting_ui *module_plotting_ui*

Description

module_plotting_ui

Usage

```
module_plotting_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "plotting",  
      module_plotting_ui(  
        "modulePlotting"  
      )  
    )  
  )  
}
```

module_results_server *module_results_server*

Description

module_results_server

Usage

```
module_results_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: <code>input_re = reactive(input)</code>
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {
  rv <- list()
  logfilename <- paste0(tempdir(), "/log.txt")
  shiny::callModule(
    module_results_server,
    "moduleResults",
    rv = rv,
    logfilename = logfilename
  )
}
```

module_results_ui *module_results_ui*

Description

module_results_ui

Usage

```
module_results_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "results",  
      module_results_ui(  
        "moduleResults"  
      )  
    )  
  )  
}
```

module_settings_server

module_settings_server

Description

module_settings_server

Usage

```
module_settings_server(input, output, session, rv, input_re, ...)
```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: input_re = reactive(input)
...	Further arguments, such as 'logfilename', 'csvdir' and 'plotdir'

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  rv <- list()  
  logfilename <- paste0(tempdir(), "/log.txt")  
  shiny::callModule(  
    module_settings_server,  
    "moduleSettings",  
    rv = rv,  
    logfilename = logfilename  
  )  
}
```

module_settings_ui *module_settings_ui*

Description

module_settings_ui

Usage

```
module_settings_ui(id)
```

Arguments

id A character. The identifier of the shiny object

Value

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {  
  shinydashboard::tabItems(  
    shinydashboard::tabItem(  
      tabName = "settings",  
      module_settings_ui(  
        "moduleSettings"  
      )  
    )  
  )  
}
```

```

    )
  )
}

```

```

module_statistics_server

```

```

    module_statistics_server

```

Description

```

module_statistics_server

```

Usage

```

module_statistics_server(input, output, session, rv, input_re)

```

Arguments

input	Shiny server input object
output	Shiny server output object
session	Shiny session object
rv	The global 'reactiveValues()' object, defined in server.R
input_re	The Shiny server input object, wrapped into a reactive expression: input_re = reactive(input)

Value

The function returns a shiny server module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```

if (interactive()) {
  rv <- list()
  logfilename <- paste0(tempdir(), "/log.txt")
  shiny::callModule(
    module_statistics_server,
    "moduleStatistics",
    rv = rv,
    logfilename = logfilename
  )
}

```

`module_statistics_ui` *module_statistics_ui*

Description`module_statistics_ui`**Usage**`module_statistics_ui(id)`**Arguments**`id` A character. The identifier of the shiny object**Value**

The function returns a shiny ui module.

See Also

<https://shiny.rstudio.com/articles/modules.html>

Examples

```
if (interactive()) {
  shinydashboard::tabItems(
    shinydashboard::tabItem(
      tabName = "statistics",
      module_statistics_ui(
        "moduleStatistics"
      )
    )
  )
}
```

Index

launch_app, 2

module_calibrationfile_server, 3
module_calibrationfile_ui, 4
module_correctedplots_server, 5
module_correctedplots_ui, 6
module_correctedstatistics_ui, 6
module_correctedstats_server, 7
module_experimentalfile_server, 8
module_experimentalfile_ui, 9
module_fileupload_server, 10
module_fileupload_ui, 11
module_info_server, 12
module_info_ui, 13
module_log_server, 13
module_log_ui, 14
module_modelselection_server, 15
module_modelselection_ui, 16
module_plotting_server, 17
module_plotting_ui, 18
module_results_server, 18
module_results_ui, 19
module_settings_server, 20
module_settings_ui, 21
module_statistics_server, 22
module_statistics_ui, 23