

# Package ‘BiasCorrector’

September 22, 2020

**Title** A GUI to Correct Measurement Bias in DNA Methylation Analyses

**Version** 0.1.2

**Date** 2020-09-22

**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

**LazyData** true

**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](mailto:lorenz.kapsner@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-09-22 15:20:21 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 . . . . .	7
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

<b>Index</b>	<b>24</b>
--------------	-----------

---

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

---

## Description

Launch BiasCorrector

## Usage

```

launch_app(
  port = 3838,
  plotdir = "plots",
  csvdir = "csv",
  logfile = "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: 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_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: <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_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: 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_fileupload_server,  
    "moduleFileUpload",  
    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, 7  
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