Package ‘RCNA’

July 21, 2025

Title Robust Copy Number Alteration Detection (RCNA)

Version 1.0

Description Detects copy number alteration events in targeted exon sequencing data for tumor samples without matched normal controls. The advantage of this method is that it can be applied to smaller sequencing panels including evaluations of exon, transcript, gene, or even user specified genetic regions of interest. Functions in the package include steps for GC-content correction, calculation of quantile based normal karyotype ranges, and calculation of feature score. Cutoffs for ``normal'' quantile and score are user-adjustable.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

Imports doParallel, methods, utils, modeest, data.table, foreach, R.utils

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, usethis

Config/testthat/edition 3

Collate 'classes.r' 'estimate_feature_score.r' 'estimate_nkr.r' 'correct_gc_bias.r' 'RCNA.r' 'data.R'

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Matt Bradley [aut, cre]

Maintainer Matt Bradley <`mbradley@emmes.com`>

Repository CRAN

Date/Publication 2024-12-03 18:40:06 UTC

Contents

correct_gc_bias 2
create_RCNA_object 4
estimate_feature_score 7
estimate_nkr 9
example_obj 12
RCNA_analysis-class 12
RCNA_object-class 13
run_RCNA 13

Index 17

`correct_gc_bias` *correct_gc_bias: Estimate and correct GC bias in coverage*

Description

This generic function is used to run to calculate and correct GC-content-based coverage bias

This function optionally estimates and then corrects the GC bias based on a GC-content factor file that is either generated or provided by the user using a sliding window approach. It creates a GC factor file and a corrected coverage file, both of which are placed in the output directory under '/gc'.

This function optionally estimates and then corrects the GC bias based on a GC-content factor file that is either generated or provided by the user using a sliding window approach. It creates a GC factor file and a corrected coverage file, both of which are placed in the output directory under '/gc'.

Usage

```
correct_gc_bias(obj, ...)

## Default S3 method:
correct_gc_bias(
  obj = NULL,
  df = NULL,
  sample.names = NULL,
  ano.file,
  out.dir = NULL,
  ncpus = 1,
  file.raw.coverage = NULL,
  file.corrected.coverage = NULL,
  file.gc.factor = NULL,
  win.size = 75,
  gc.step = 0.01,
  estimate_gc = TRUE,
  verbose = FALSE,
  ...
)
```

```
## S3 method for class 'RCNA_object'
correct_gc_bias(obj, verbose = FALSE, ...)
```

Arguments

`obj`	A RCNA_object type object - parameters will be pulled from the object instead, specifically from the ‘gcParams‘ slot.
`...`	Additional arguments (unused)
`df`	Path to the config file, or a ‘data.frame‘ object containing the valid parameters. Valid column names are ‘file.raw.coverage‘, ‘file.gc.factor‘, ‘file.corrected.coverage‘, and ‘sample.names‘. Additional columns will be ignored.
`sample.names`	Character vector of sample names. Alternatively can be specified in ‘df‘.
`ano.file`	Location of the annotation file. This file must be in CSV format and contain the following information (with column headers as specified): "feature,chromosome,start,end".
`out.dir`	Output directory for results. A subdirectory for results will be created under this + ‘/nkr/‘.
`ncpus`	Integer number of CPUs to use. Specifying more than one allows this function to be parallelized by feature.
`file.raw.coverage`	Character vector listing the raw input coverage files. Must be the same length as ‘sample.names‘. Alternatively can be specified in ‘df‘.
`file.corrected.coverage`	Character vector listing the corrected input coverage files. If not specified new names will be generated based on the raw coverage files.
`file.gc.factor`	Character vector listing the GC factor files used to correct coverage. If ‘estimate_gc=FALSE‘ then this must be provided. Otherwise it is ignored.
`win.size`	Size in base pairs of the sliding window used to estimate and correct the GC bias.
`gc.step`	Bin size for GC bias in the GC factor file. If the GC factor file is provided then the file must have corresponding bin sizes.
`estimate_gc`	Logical determining if GC content estimation should be performed. If set to ‘FALSE‘ then a factor file must be provided via ‘file.gc.factor‘ or in ‘df‘.
`verbose`	If set to TRUE will display more detail

Details

This function can be run as a stand-alone or as part of run_RCNA

The ‘df‘ argument corresponds to the ‘gcParams‘ matrix on RCNA_object. Valid column names are ‘sample.names‘, ‘file.raw.coverage‘, ‘file.corrected.coverage‘, and ‘file.gc.factor‘. The ‘file.gc.factor‘ column is not required if ‘estimate_gc=TRUE‘. Additional columns will be ignored.

For more parameter information, see estimate_nkr.default.

Value

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

See Also

RCNA_object, RCNA_analysis, run_RCNA

Examples

```
## Run GC-bias estimation and correction on example object
# See \link{example_obj} for more information on example
example_obj@ano.file <- system.file("examples" ,"annotations-example.csv",
 package = "RCNA")
raw.cov <- system.file("examples", "coverage",
                       paste0(example_obj@sample.names, ".txt.gz"), package = "RCNA")
example_obj@gcParams$file.raw.coverage <- raw.cov
example_obj
# Create output directory
dir.create(file.path("output", "gc"), recursive = TRUE)
# Estimate and correct GC bias, append results
correct_gc_analysisObj <- correct_gc_bias(example_obj)
example_obj@commands <- c(example_obj@commands, correct_gc_analysisObj)
```

`create_RCNA_object` *RCNA_object constructor*

Description

An S4 class used to specify parameters for an analysis run

Usage

```
create_RCNA_object(
  sample.names,
  ano.file,
  ncpu = 1,
  out.dir = tempdir(),
  file.coverage = NULL,
  gcParams = NULL,
  win.size = 75,
  gc.step = 0.01,
```

```
    file.raw.coverage = NULL,
    file.corrected.coverage = NULL,
    file.gc.factor = NULL,
    estimate_gc = TRUE,
    nkrParams = NULL,
    file.nkr.coverage = NULL,
    nkr = 0.9,
    x.norm = NULL,
    norm.cov.matrix = NULL,
    scoreParams = NULL,
    file.score.coverage = NULL,
    score.cutoff = 0.5,
    low.score.cutoff = NULL,
    high.score.cutoff = NULL,
    commands = list(),
    verbose = FALSE
  )
```

Arguments

`sample.names`	Character vector containing names of subjects
`ano.file`	Character single file path detailing a feature-wise annotation file
`ncpu`	Numeric value specifying number of cores to use for analysis. Multiple cores will lead to parallel execution.
`out.dir`	Character vector containing the name of each subject's output directory
`file.coverage`	Character vector containing the path to the input coverage files for NKR and CNA score estimation.
`gcParams`	Data Frame storing all run parameters for the correct_gc_bias function. Can be specified by a file path to a CSV file, 'data.frame', or (if not specified) will be generated by other arguments.
`win.size`	Numeric value detailing the size of the sliding window used to calculate and detect correct GC-content correction.
`gc.step`	Numeric value detailing the size of each GC-content bin. If providing pre-calculated GC factor file this must match the bins in that file.
`file.raw.coverage`	Character vector containing the filename of the raw coverage files for GC-content correction. Must be used in combination with 'estimate_gc' set to TRUE.
`file.corrected.coverage`	Character vector containing the filename of the corrected coverage files.
`file.gc.factor`	Character vector containing the filename of GC factor files. Used if and only if 'estimate_gc' is set to FALSE.
`estimate_gc`	Logical that determines if GC bias factor is calculated. If set to TRUE then GC factor files will be generated for each sample. If set to FALSE then GC factor files must be supplied via 'file.gc.factor'.

`nkrParams`	Data Frame storing all run parameters for the estimate_nkr function. Can be specified by a file path to a CSV file, ‘data.frame‘, or (if not specified) will be generated by other arguments.
`file.nkr.coverage`	Character vector containing the filename of the input coverage file for NKR estimation. Defaults to ‘file coverage‘ if not specified.
`nkr`	Numeric between 0 and 1 which specifies the coverage quantile that should be considered a "normal" karyotype range for each position. Lowering this value may increase sensitivity but also Type I error.
`x.norm`	Logical vector with length equal to the length of ‘sample.names‘, denoting whether each subject has to be X-normalized. Subjects with an XX karyotype should be set to TRUE to avoid double-counting the coverage on the X chromosome. Set to FALSE if chrX coverage is already normalized.
`norm.cov.matrix`	Character containing the directory or file name of the normalized coverage matrix. Generated by `estimate_nkr` if file doesn’t exist.
`scoreParams`	Data Frame storing all run parameters for the estimate_feature_score function. Can be specified by a file path to a CSV file, ‘data.frame‘, or (if not specified) will be generated by other arguments.
`file.score.coverage`	Character vector containing the input coverage file for the scoring function. Defaults to ‘file.coverage‘ if not specified.
`score.cutoff`	Numeric between 0 and 1 which specifies the score filter on the results file. This parameter creates a symmetrical cutoff around 0, filtering all results whose absolute value is less than the specified value. Non-symmetrical cutoffs can be specified using ‘low.score.cutoff‘ and ‘high.score.cutoff‘.
`low.score.cutoff`	Numeric between 0 and 1 which specifies the lower score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`high.score.cutoff`	Numeric between 0 and 1 which specifies the upper score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`commands`	RCNA_analysis object storing commands and parameters from previous function runs on this object. For more information, see `RCNA_analysis`.
`verbose`	Show more messages and warnings. Useful for debugging.

Value

A RCNA_object class object with the specified parameters.

See Also

`RCNA_analysis`, `run_RCNA`

Examples

```
# Create an example object - see \link{example_obj} for more information.
samples <- c("ex-sample-1", "ex-sample-2", "ex-sample-3")
ex.obj <- create_RCNA_object(sample.names = samples,
             ano.file = system.file("examples" ,"annotations-example.csv", package = "RCNA"),
                    out.dir = "output",
                    file.raw.coverage = system.file("examples", "coverage",
                       paste0(samples, ".txt.gz"), package = "RCNA"),
                    norm.cov.matrix = file.path("output", "norm-cov-matrix.csv.gz"),
                    nkr = 0.9,
                    x.norm = "FALSE",
                    low.score.cutoff = -0.35,
                    high.score.cutoff = 0.35,
                    ncpu = 1)
class(ex.obj)
```

---

estimate_feature_score

estimate_feature_score: Estimate CNV score for each gene in the annotation file

---

Description

This function estimates the the CNA score for each feature in the annotation file. It creates two flat file text tables with a row for each feature, which is placed in the output directory under '/score' - one with the score filter applied and one with all score results reported.

This function estimates the the CNA score for each feature in the annotation file. It creates two flat file text tables with a row for each feature, which is placed in the output directory under '/score' - one with the score filter applied and one with all score results reported.

Usage

```
estimate_feature_score(obj, ...)

## Default S3 method:
estimate_feature_score(
  obj = NULL,
  df = NULL,
  sample.names = NULL,
  ano.file,
  out.dir = NULL,
  ncpus = 1,
  file.score.coverage = NULL,
  score.cutoff = 0.5,
  low.score.cutoff = NULL,
  high.score.cutoff = NULL,
```

```
  verbose = FALSE,
  ...
)

## S3 method for class 'RCNA_object'
estimate_feature_score(obj, verbose = FALSE, ...)
```

Arguments

`obj`	A RCNA_object type object - parameters will be pulled from the object instead, specifically from the ‘scoreParams‘ slot.
`...`	Additional arguments (unused)
`df`	Path to the config file, or a ‘data.frame‘ object containing the valid parameters. Valid column names are ‘file.score.coverage‘ and ‘sample.names‘. Additional columns will be ignored.
`sample.names`	Character vector of sample names. Alternatively can be specified in ‘df‘.
`ano.file`	Location of the annotation file. This file must be in CSV format and contain the following information (with column headers as specified): "feature,chromosome,start,end".
`out.dir`	Output directory for results. A subdirectory for results will be created under this + ‘/nkr/‘.
`ncpus`	Integer number of CPUs to use. Specifying more than one allows this function to be parallelized by feature.
`file.score.coverage`	Character vector listing the input coverage files. Must be the same length as ‘sample.names‘. Alternatively can be specified in ‘df‘.
`score.cutoff`	Numeric between 0 and 1 which specifies the score filter on the results file. This parameter creates a symmetrical cutoff around 0, filtering all results whose absolute value is less than the specified value. Non-symmetrical cutoffs can be specified using ‘low.score.cutoff‘ and ‘high.score.cutoff‘.
`low.score.cutoff`	Numeric between 0 and 1 which specifies the lower score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`high.score.cutoff`	Numeric between 0 and 1 which specifies the upper score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`verbose`	If set to TRUE will display more detail

Details

This function can be run as a stand-alone or as part of run_RCNA.

The ‘df‘ argument corresponds to the ‘scoreParams‘ matrix on RCNA_object. Valid column names are ‘sample.names‘ and ‘file.score.coverage‘. Additional columns will be ignored.

For more parameter information, see estimate_feature_score.default.

Value

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

See Also

RCNA_object, RCNA_analysis, run_RCNA

Examples

```
## Estimate feature scores on example object
# See \link{example_obj} for more information on example
example_obj@ano.file <- system.file("examples" ,"annotations-example.csv", package = "RCNA")
example_obj
# Create output directories
dir.create(file.path("output", "score"), recursive = TRUE)
# Copy example GC-corrected coverage files
cov.corrected <- system.file("examples", "gc", package = "RCNA")
file.copy(from = cov.corrected, to = "output", recursive = TRUE)
# Copy example NKR results for "feature_a"
nkr.res <- system.file("examples", "nkr", package = "RCNA")
file.copy(from = nkr.res, to = "output", recursive = TRUE)
# Run score estimation for "feature_a" and append results
estimate_feature_score_analysisObj <- estimate_feature_score(example_obj)
example_obj@commands <- c(example_obj@commands, estimate_feature_score_analysisObj)
```

`estimate_nkr` *estimate_nkr: Estimate CNA "normal" karyotype ranges*

Description

This generic function is used to run normal karyotype range estimation.

This function estimates the normal karyotype range for each feature in the annotation file. It creates an .RData object for each feature, which is placed in the output directory under '/nkr'. This intermediate output is used in estimate_feature_score.

This function estimates the normal karyotype range for each feature in the annotation file. It creates an .RData object for each feature, which is placed in the output directory under '/nkr'. This intermediate output is used in estimate_feature_score.

Usage

```
estimate_nkr(obj, ...)

## Default S3 method:
estimate_nkr(
  obj = NULL,
  df = NULL,
  sample.names = NULL,
  ano.file,
  out.dir = NULL,
  ncpus = 1,
  file.ci.coverage = NULL,
  nkr = 0.9,
  x.norm = NULL,
  norm.cov.matrix = NULL,
  verbose = FALSE,
  ...
)

## S3 method for class 'RCNA_object'
estimate_nkr(obj, verbose = FALSE, ...)
```

Arguments

`obj`	A RCNA_object type object - parameters will be pulled from the object instead, specifically from the ‘nkrParams‘ slot.
`...`	Additional arguments (unused)
`df`	Path to the config file, or a ‘data.frame‘ object containing the valid parameters. Valid column names are ‘file.nkr.coverage‘, ‘x.norm‘, and ‘sample.names‘. Additional columns will be ignored.
`sample.names`	Character vector of sample names. Alternatively can be specified in ‘df‘.
`ano.file`	Location of the annotation file. This file must be in CSV format and contain the following information (with column headers as specified): "feature,chromosome,start,end".
`out.dir`	Output directory for results. A subdirectory for results will be created under this + ‘/nkr/‘.
`ncpus`	Integer number of CPUs to use. Specifying more than one allows this function to be parallelized by feature.
`file.ci.coverage`	Character vector listing the input coverage files. Must be the same length as ‘sample.names‘. Alternatively can be specified in ‘df‘.
`nkr`	Numeric between 0 and 1 which specifies the coverage quantile that should be considered a "normal" karyotype range for each position. Lowering this value may increase sensitivity but also Type I error.
`x.norm`	Whether or not to perform normalization for normal female/XX karyotype (default = FALSE). Can be specified for each sample separately via ‘df‘ column labeled ‘x.norm‘.

`norm.cov.matrix`	Character file path detailing the location of the normalized coverage matrix generated by this function. Re-using this file between runs can cut down on runtime significantly for large sample sizes. If the file doesn't exist yet it will be created at this location. If this file name ends in ".gz" then the output will be compressed using gzip.
`verbose`	If set to TRUE will display more detail

Details

This function can be run as a stand-alone or as part of run_RCNA

The 'df' argument corresponds to the 'nkrParams' matrix on RCNA_object. Valid column names are 'sample.names', 'file.ci.coverage', and 'x.norm'. Additional columns will be ignored.

For more parameter information, see estimate_nkr.default.

Value

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

A RCNA_analysis class object that describes the input parameters and output files generated by this step of the workflow.

See Also

RCNA_object, RCNA_analysis, run_RCNA

Examples

```
## Run NKR estimation on example object
# See \link{example_obj} for more information on example
example_obj@ano.file <- system.file("examples" ,"annotations-example.csv", package = "RCNA")
example_obj
# Create output directory
dir.create(file.path("output", "nkr"), recursive = TRUE)
# Copy example GC-corrected coverage files
cov.corrected <- system.file("examples", "gc", package = "RCNA")
file.copy(from = cov.corrected, to = "output", recursive = TRUE)
# Run NKR estimation, append results
estimate_nkr_analysisObj <- estimate_nkr(example_obj)
example_obj@commands <- c(example_obj@commands, estimate_nkr_analysisObj)
```

`example_obj` *Example RCNA_object*

Description

An example RCNA object used to run examples and demonstrate the structure of the custom S4 object provided in this package. This example uses a dummy feature ("feature_a") and three coverage files which were subset to the length of the dummy feature to be concise and quick to run. An annotation file has been included in the ‘inst/‘ directory along with the coverage files. This object is compiled in the ‘`create_RCNA_object`‘ function’s documentation. In order to use this example, you should make the following replacements: `example_obj@ano.file <- system.file("examples" ,"annotations-example.csv", package = "RCNA") raw.cov <- system.file("examples", "coverage", paste0(samples, ".txt.gz"), package = "RCNA") example_obj@gcParams$file.raw.coverage <- raw.cov`

Usage

```
example_obj
```

Format

An RCNA object created using create_RCNA_object(). See create_RCNA_object for more details on the slots of this object.

Details

This will set the location of the example annotation file and the example raw coverage files to the flat files included with the package.

`RCNA_analysis-class` *RCNA_analysis constructor*

Description

An S4 class used to track parameters from a specific RCNA function execution.

Slots

- `call` A character vector detailing the function call that (‘correct_gc_bias‘, ‘estimate_nkr‘, ‘estimate_feature_score‘) was performed to produce this S4 object.
- `params` A list corresponding to the parameters that were submitted with the associated function call
- `res.files` A list containing the names of the flat files that were created by the documented function call.

`RCNA_object-class` *RCNA.object definition*

Description

An S4 class used to specify parameters for an analysis run

Slots

`sample.names` Required. Character vector containing names of subjects

`ano.file` Required. Character single file path detailing a feature-wise annotation file

`out.dir` Required. Character vector containing the name of each subject?s output directory

`gcParams` Data Frame storing all run parameters for the correct_gc_bias function

`nkrParams` Data Frame storing all run parameters for the estimate_nkr function

`scoreParams` Data Frame storing all run parameters for the estimate_feature_score function

`commands` RCNA_analysis object storing commands and parameters from previous function runs on this object

See Also

`run_RCNA` `RCNA_analysis`,

`run_RCNA` *run_RCNA: Perform RCNA copy number detection workflow*

Description

‘run_RCNA‘ will execute correct_gc_bias, estimate_nkr, and estimate_feature_score in that specific order. For more information, see each of those functions’ individual documentation.

‘run_RCNA‘ will execute correct_gc_bias, estimate_nkr, and estimate_feature_score in that specific order. For more information, see each of those functions’ individual documentation, or create_RCNA_object.

‘run_RCNA‘ will execute correct_gc_bias, estimate_nkr, and estimate_feature_score in that specific order. For more information, see each of those functions’ individual documentation.

Usage

```
run_RCNA(obj, ...)

## Default S3 method:
run_RCNA(
  obj = NULL,
  sample.names,
  ano.file,
  out.dir = tempdir(),
  gcParams = NULL,
  win.size = 75,
  gc.step = 0.01,
  file.raw.coverage = NULL,
  file.corrected.coverage = NULL,
  file.gc.factor = NULL,
  estimate_gc = TRUE,
  nkrParams,
  file.nkr.coverage = NULL,
  ncpu = 1,
  nkr = 0.9,
  x.norm = NULL,
  scoreParams,
  score.cutoff = 0.5,
  low.score.cutoff = NULL,
  high.score.cutoff = NULL,
  commands = c(),
  verbose = FALSE,
  ...
)

## S3 method for class 'RCNA_object'
run_RCNA(obj, estimate_gc = TRUE, verbose = FALSE, ...)
```

Arguments

`obj`	An ‘RCNA_object‘ type created by create_RCNA_object.
`...`	Additional arguments (unused).
`sample.names`	Character vector containing names of subjects
`ano.file`	Character single file path detailing a feature-wise annotation file
`out.dir`	Character vector containing the name of each subject’s output directory
`gcParams`	Data Frame storing all run parameters for the correct_gc_bias function. Can be specified by a file path to a CSV file, ‘data.frame‘, or (if not specified) will be generated by other arguments.
`win.size`	Numeric value detailing the size of the sliding window used to calculate and detect correct GC-content correction.
`gc.step`	Numeric value detailing the size of each GC-content bin. If providing pre-calculated GC factor file this must match the bins in that file.

`file.raw.coverage`	Character vector containing the filename of the raw coverage files for GC-content correction. Must be used in combination with ‘estimate_gc‘ set to TRUE.
`file.corrected.coverage`	Character vector containing the filename of the corrected coverage files.
`file.gc.factor`	Character vector containing the filename of GC factor files. Used if and only if ‘estimate_gc‘ is set to FALSE.
`estimate_gc`	A logical which determines if GC estimation should be performed. For more information, see correct_gc_bias.
`nkrParams`	Data Frame storing all run parameters for the estimate_nkr function. Can be specified by a file path to a CSV file, ‘data.frame‘, or (if not specified) will be generated by other arguments.
`file.nkr.coverage`	Character vector containing the filename of the input coverage file for NKR estimation. Defaults to ‘file coverage‘ if not specified.
`ncpu`	Numeric value specifying number of cores to use for analysis. Multiple cores will lead to parallel execution.
`nkr`	Numeric between 0 and 1 which specifies the coverage quantile that should be considered a "normal" karyotype range for each position. Lowering this value may increase sensitivity but also Type I error.
`x.norm`	Logical vector with length equal to the length of ‘sample.names‘, denoting whether each subject has to be X-normalized. Subjects with an XX karyotype should be set to TRUE to avoid double-counting the coverage on the X chromosome. Set to FALSE if chrX coverage is already normalized.
`scoreParams`	Data Frame storing all run parameters for the estimate_feature_score function. Can be specified by a file path to a CSV file, ‘data.frame‘, or (if not specified) will be generated by other arguments.
`score.cutoff`	Numeric between 0 and 1 which specifies the score filter on the results file. This parameter creates a symmetrical cutoff around 0, filtering all results whose absolute value is less than the specified value. Non-symmetrical cutoffs can be specified using ‘low.score.cutoff‘ and ‘high.score.cutoff‘.
`low.score.cutoff`	Numeric between 0 and 1 which specifies the lower score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`high.score.cutoff`	Numeric between 0 and 1 which specifies the upper score cutoff. Defaults to ‘score.cutoff‘ if not specified.
`commands`	RCNA_analysis object storing commands and parameters from previous function runs on this object. For more information, see `RCNA_analysis`.
`verbose`	If set to TRUE will display more detailed error messages.

Value

A RCNA_object class object that was used during the workflow, with RCNA_analysis objects in the ‘commands‘ slot that describes the run parameters and results of each step in the workflow.

A RCNA_object class object that was used during the workflow, with RCNA_analysis objects in the ‘commands‘ slot that describes the run parameters and results of each step in the workflow. For more details on outputs, see estimate_nkr, correct_gc_bias, and estimate_feature_score.

A RCNA_object class object that was used during the workflow, with RCNA_analysis objects in the ‘commands‘ slot that describes the run parameters and results of each step in the workflow. For more details on outputs, see estimate_nkr, correct_gc_bias, and estimate_feature_score.

See Also

RCNA_object, RCNA_analysis, correct_gc_bias, run_RCNA, estimate_feature_score

Examples

```
## Run RCNA workflow on example object
# See ?example_obj for more information on example
example_obj@ano.file <- system.file("examples" ,"annotations-example.csv", package = "RCNA")
raw.cov <- system.file("examples", "coverage",
                       paste0(example_obj@sample.names, ".txt.gz"), package = "RCNA")
example_obj@gcParams$file.raw.coverage <- raw.cov
example_obj
# Run RCNA workflow
result_obj <- run_RCNA(example_obj)
```

Index

∗ **datasets**
    example_obj, 12

correct_gc_bias, 2, *13*, *15, 16*
create_RCNA_object, 4, *12–14*

estimate_feature_score, 7, *9*, *13*, *16*
estimate_feature_score.default, *8*
estimate_nkr, *6*, 9, *13*, *16*
estimate_nkr.default, *3*, *11*
example_obj, 12

RCNA_analysis, *4*, *6*, *9*, *11*, *13*, *15, 16*
RCNA_analysis *(*RCNA_analysis-class*)*, 12
RCNA_analysis-class, 12
RCNA_object, *3, 4*, *6*, *8, 9*, *11*, *15, 16*
RCNA_object *(*RCNA_object-class*)*, 13
RCNA_object-class, 13
run_RCNA, *3, 4*, *6*, *8, 9*, *11*, *13*, 13, *16*