Package 'TidyMultiqc'

April 9, 2021

Type Package

Title Converts 'MultiQC' Reports into Tidy Data Frames

Version 0.1.0

Author Michael Milton

Maintainer Michael Milton <michael.r.milton@gmail.com>

Description Provides the means to convert 'multiqc_data.json' files, produced by the wonderful 'MultiQC' tool, into tidy data frames for downstream analysis in R. This analysis might involve cohort analysis, quality control visualisation, change-point detection, statistical process control, clustering, or any other type of quality analysis.

License GPL (>= 3)

Encoding UTF-8

Imports assertthat, dplyr, HistDat (>= 0.2.0), jsonlite, magrittr, purrr, rlang, stringr, tibble

Suggests tidyr, testthat (>= 3.0.0), knitr, rmarkdown, ggplot2

Config/testthat/edition 3

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-09 09:00:02 UTC

R topics documented:

`TidyMultiqc-package` *TidyMultiqc: Converting MultiQC reports into tidy data frames*

Description

This package provides the means to convert `multiqc_data.json` files, produced by the wonderful MultiQC tool, into tidy data.frames for downstream analysis in R. This analysis might involve cohort analysis, quality control visualisation, changepoint detection, statistical process control, clustering, or any other type of quality analysis.

Core API

The public API to this package

- `load_multiqc()`

Plot Extractor Functions

These functions can be used as arguments to `load_multiqc()` to specify how to extract data from MultiQC plots

- `extract_ignore_x()`
- `extract_xy()`
- `extract_histogram()`

Summary Functions

These are also passed as arguments to `load_multiqc()`. In most cases you can use normal summary statistics like `base::mean()`, but these are some other useful ones you might want.

- `summary_q30()`
- `summary_extract_df()`

`extract_histogram` *Extractor function that calculates statistics for a histogram.*

Description

Extractor function that calculates statistics for a histogram.

Usage

```
extract_histogram(data, as_hist_dat = TRUE)
```

Arguments

data	Provided internally, users don't need to provide this.
as_hist_dat	If true return an instance of the HistDat::HistDat class. Otherwise return a 1-D numeric vector. Default true, as this is strongly recommended to avoid crashing the R interpreter with large counts in the histogram.

Details

For example this might be relevant for the "Coverage histogram" plot from Qualimap. By default this returns a HistDat::HistDat instance, which is compatible with most common summary statistics (mean, quantile, etc), so your summary statistic functions can be ordinary R functions.

Value

A single HistDat::HistDat instance, or a 1-D numeric vector

See Also

Other extractors: extract_ignore_x(), extract_xy()

Examples

```
report <- load_multiqc(
  system.file("extdata", "HG00096/multiqc_data.json", package = "TidyMultiqc"),
  sections = "plots",
  plot_opts = list(
    `fastqc_per_sequence_quality_scores_plot` = list(
      extractor = extract_histogram,
      summary = list(mean = mean),
      prefix = "quality"
    )
  )
)
```

---

extract_ignore_x *Extractor function that ignores the x-axis and applies statistics over the y-values.*

Description

Extractor function that ignores the x-axis and applies statistics over the y-values.

Usage

```
extract_ignore_x(data)
```

Arguments

data	Provided internally, users don't need to provide this.

Details

For example this might be relevant for a mean per-base fastq quality score. This will let you then calculate the overall mean quality of the reads.

Value

A 1-D numeric vector of y-values from the plot

See Also

Other extractors: `extract_histogram()`, `extract_xy()`

Examples

```
report <- load_multiqc(
   system.file("extdata", "HG00096/multiqc_data.json", package = "TidyMultiqc"),
   sections = "plots",
   plot_opts = list(
     `fastqc_per_base_sequence_quality_plot` = list(
       extractor = extract_ignore_x,
       summary = list(mean = mean),
       prefix = "quality"
     )
   )
)
```

---

extract_xy *Extractor function that extracts the (x, y) pairs in the plot and puts them as columns in a data.frame, with colnames "x" and "y"*

Description

Extractor function that extracts the (x, y) pairs in the plot and puts them as columns in a data.frame, with colnames "x" and "y"

Usage

```
extract_xy(data)
```

Arguments

`data`	Provided internally, users don't need to provide this.

Details

Since this extractor returns an entire data.frame, the extractor function cannot use ordinary summary statistics like `mean`, `median` etc. If you want to do that, look into the other extractors. Instead, you will need summary functions that pull out a single value from the data.frame.

Value

A tibble with the "x" column corresponding to the x-values in the plot, and a "y" column corresponding to the y-values in the plot.

See Also

Other extractors: `extract_histogram()`, `extract_ignore_x()`

Examples

```
report <- load_multiqc(
  system.file("extdata", "wgs/multiqc_data.json", package = "TidyMultiqc"),
  sections = "plots",
  plot_opts = list(
    qualimap_genome_fraction = list(
      extractor = extract_xy,
      summary = list(
        `%Q30` = purrr::partial(summary_extract_df, row_select = x == 30)
      )
    )
  )
)
```

---

load_multiqc *Loads one or more MultiQCs report into a data frame*

Description

Loads one or more MultiQCs report into a data frame

Usage

```
load_multiqc(
  paths,
  plot_opts = list(),
  find_metadata = function(...) {     list() },
  sections = "general"
)
```

Arguments

`paths`	A vector of filepaths to multiqc_data.json files
`plot_opts`	A named list mapping the internal MultiQC plot name, e.g. "fastqc_per_sequence_quality_scores_plot" to a list of options for that plot. The list can have the following keys:

extractor Mandatory for scatter/line plots, ignored for bar graphs. A function which converts the raw plot JSON into a some kind of data, usually a vector. Often you will want to use a built-in `extract_x` functions provided by this package

summary A named list of functions that each map the output from the extractor function (usually a 1-D vector) to a scalar, to "summarise" it. For example, you might want to use the `base::mean()` function to summarise the plot. See also the `summary_x` functions in this package.

prefix Optional. A new name for this plot. MultiQC sometimes has some unwieldy names for its plot, so this lets you rename them

`find_metadata` A function that will be called with a sample name and the parsed JSON and returns a named list of metadata fields for the sample

`sections` Vector of the sections to include in the output: 'plots' in the list means parse plot data, 'general' means parse the general stats section, and 'raw' means parse the raw data section. This defaults to 'general', which tends to contain the most useful statistics

Value

A tibble (data.frame subclass) with QC data and metadata as columns, and samples as rows

Examples

```
load_multiqc(
  system.file("extdata", "wgs/multiqc_data.json", package = "TidyMultiqc"),
  sections = c("plots", "general", "raw"),
  plot_opts = list(
    fastqc_per_sequence_quality_scores_plot = list(
      summary = list(`%q30` = summary_q30),
      extractor = extract_histogram,
      prefix = "quality"
    )
  )
)
```

---

parse_plot_features *Returns a list of summary statistics for a plotly plot, provided as a list e.g. from jsonlite.*

Description

Returns a list of summary statistics for a plotly plot, provided as a list e.g. from jsonlite.

Usage

```
parse_plot_features(
  plot_data,
  prefix,
  extractor = extract_ignore_x,
  summary = list(mean = mean)
)
```

Arguments

plot_data	A list containing the names plot_type, datasets and config.
prefix	The prefix for this plot type in the final data frame
extractor	A function which converts the raw plot JSON into a vector
summary	A function that maps a vector to a scalar

Details

This is an internal function that may be of some use to those who want to extract data from plotly JSON, outside of the context of MultiQC. If you are trying to extract data from a MultiQC report, please use the normal load_multiqc() function instead. Please also refer to load_multiqc() for more information on these arguments, as they are identical to the elements of the plot_opts list.

Value

A list of samples, each containing a list of plots, each containing a list of summary stats

Examples

```
parse_plot_features(
  plot_data=jsonlite::read_json(
    system.file(
      "extdata", "wgs/multiqc_data.json", package = "TidyMultiqc"
    )
  )$report_plot_data$snpeff_effects,
  prefix='effects'
)
```

---

summary_extract_df *Summary function that only works with the extract_xy() extractor. Extracts a single point from the x,y data.frame by first selecting a row and then returning the y value for that row*

Description

Summary function that only works with the extract_xy() extractor. Extracts a single point from the x,y data.frame by first selecting a row and then returning the y value for that row

Usage

```
summary_extract_df(df, row_select, col = "y")
```

Arguments

df	A data.frame with x and y columns. This is provided automatically by the package and users don't need to provide this.
row_select	An expression that will be pass through to dplyr::filter(). This is a quoted argument so you can refer to the variables x and y
col	A column name, either "x" or "y"

Value

The value in a single cell of the data.frame

`summary_q30` *Summary statistic for finding the %Q30 of a dataset of quality scores This is the proportion of total reads in a dataset that have a quality score of 30 or above.*

Description

Summary statistic for finding the %Q30 of a dataset of quality scores This is the proportion of total reads in a dataset that have a quality score of 30 or above.

Usage

```
summary_q30(vec)
```

Arguments

`vec`	Either a HistDat::HistDat or a 1-D numeric vector

Value

The %Q30 of the dataset, as a numeric of length 1

Index