Package 'alevinQC'

March 29, 2021

Type Package

Title Generate QC Reports For Alevin Output

Version 1.6.1

Description Generate QC reports summarizing the output from an alevin run. Reports can be generated as html or pdf files, or as shiny applications.

Encoding UTF-8

Depends R (>= 4.0)

Imports rmarkdown (>= 2.5), tools, methods, ggplot2, GGally, dplyr, rjson, shiny, shinydashboard, DT, stats, utils, tximport (>= 1.17.4), cowplot, rlang

RoxygenNote 7.1.1

Suggests knitr, BiocStyle, testthat

VignetteBuilder knitr

biocViews QualityControl, SingleCell

URL https://github.com/csoneson/alevinQC

BugReports https://github.com/csoneson/alevinQC/issues

License MIT + file LICENSE

git_url https://git.bioconductor.org/packages/alevinQC

git_branch RELEASE_3_12

git_last_commit 528a726

git_last_commit_date 2021-02-02

Date/Publication 2021-03-29

Author Charlotte Soneson [aut, cre] (<https://orcid.org/0000-0003-3833-2169>), Avi Srivastava [aut]

Maintainer Charlotte Soneson <charlottesoneson@gmail.com>

R topics documented:

alevinQC-pkg 2
alevinQCReport 2
alevinQCShiny 4
checkAlevinInputFiles 4

plotAlevinBarcodeCollapse 5
plotAlevinHistogram 6
plotAlevinKneeNbrGenes 7
plotAlevinKneeRaw 7
plotAlevinQuant 8
plotAlevinQuantPairs 9
readAlevinQC 10

Index **11**

`alevinQC-pkg` *alevinQC*

Description

alevinQC

`alevinQCReport` *Generate alevin summary report*

Description

Generate a report summarizing the main aspects of an alevin quantification run. The report generation assumes that alevin has been run with the –dumpFeatures flag to generate the necessary output files.

Usage

```
alevinQCReport(
  baseDir,
  sampleId,
  outputFile,
  outputDir = "./",
  outputFormat = NULL,
  showCode = FALSE,
  forceOverwrite = FALSE,
  knitrProgress = FALSE,
  quiet = FALSE,
  ignorePandoc = FALSE,
  customCBList = list(),
  ...
)
```

Arguments

`baseDir`	Path to the output directory from the alevin run (should be the directory containing the `alevin` directory).
`sampleId`	Sample ID, will be used to set the title for the report.
`outputFile`	File name of the output report. The file name extension must be either `.html` or `.pdf`, and consistent with the value of `outputFormat`.

outputDir	Path to the output directory where the report will be generated.
outputFormat	The format of the output report. Either "html_document" or "pdf_document". The file name extension of outputFile must be consistent with this choice.
showCode	Logical, whether to display the R code in the report.
forceOverwrite	Logical, whether to force overwrite an existing report with the same name in the output directory.
knitrProgress	Logical, whether to display the progress of knitr when generating the report.
quiet	Logical, whether to show progress messages.
ignorePandoc	Logical, determines what to do if pandoc or pandoc-citeproc is missing (if Sys.which("pandoc") or Sys.which("pandoc-citeproc") returns ""). If ignorePandoc is TRUE, only a warning is given. The figures will be generated, but not the final report. If ignorePandoc is FALSE (default), the execution stops immediately.
customCBList	Named list with custom set(s) of barcodes to provide summary statistics/plots for, in addition to the whitelists generated by alevin.
...	Other arguments that will be passed to rmarkdown::render.

Details

When the function is called, a .Rmd template file will be copied into the output directory, and rmarkdown::render will be called to generate the final report. If there is already a .Rmd file with the same name in the output directory, the function will raise an error and stop, to avoid overwriting the existing file. The reason for this behaviour is that the copied template in the output directory will be deleted once the report is generated.

Value

Generates a summary report in the outputDir directory, and returns (invisibly) the name of the generated report.

Author(s)

Charlotte Soneson

Examples

```
alevinQCReport(baseDir = system.file("extdata/alevin_example_v0.14",
                                     package = "alevinQC"),
               sampleId = "example", outputFile = "alevinReport.html",
               outputDir = tempdir(), forceOverwrite = TRUE)
```

alevinQCShiny *Generate alevin summary shiny app*

Description

Generate a shiny app summarizing the main aspects of an alevin quantification run. The app generation assumes that alevin has been run with the –dumpFeatures flag to generate the necessary output files.

Usage

```
alevinQCShiny(baseDir, sampleId, customCBList = list())
```

Arguments

baseDir	Path to the output directory from the alevin run (should be the directory containing the alevin directory).
sampleId	Sample ID, will be used set the title for the app.
customCBList	Named list with custom set(s) of barcodes to provide summary statistics/plots for, in addition to the whitelists generated by alevin.

Value

A shiny app.

Author(s)

Charlotte Soneson

Examples

```
app <- alevinQCShiny(baseDir = system.file("extdata/alevin_example_v0.14",
                                           package = "alevinQC"),
                     sampleId = "example")
if (interactive()) {
    shiny::runApp(app)
}
```

checkAlevinInputFiles *Check that all required input files are available*

Description

Check that all required input files are available

Usage

```
checkAlevinInputFiles(baseDir)
```

Arguments

baseDir	Path to the output directory from the alevin run (should be the directory containing the alevin directory).

Value

Returns nothing, raises an error if any of the required files are missing.

Author(s)

Charlotte Soneson

Examples

```
checkAlevinInputFiles(system.file("extdata/alevin_example_v0.14",
                                  package = "alevinQC"))
```

---

plotAlevinBarcodeCollapse

Summary plot of cell barcode collapsing

Description

Plot the original frequency of each cell barcode in the original whitelist against the frequency after collapsing similar cell barcodes.

Usage

```
plotAlevinBarcodeCollapse(cbTable)
```

Arguments

cbTable	data.frame (such as the cbTable returned by readAlevinQC) with barcode frequencies before and after collapsing.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinBarcodeCollapse(alevin$cbTable)
```

`plotAlevinHistogram` *Histogram of selected summary statistic*

Description

Histogram of selected summary statistic

Usage

```
plotAlevinHistogram(
  cbTable,
  plotVar = "dedupRate",
  axisLabel = plotVar,
  colName = "inFinalWhiteList",
  cbName = "final whitelist"
)
```

Arguments

`cbTable`	`data.frame` (such as the `cbTable` returned by `readAlevinQC`) containing the desired summary statistic in a column.
`plotVar`	Character scalar giving the name of a numeric column of `cbTable` to plot.
`axisLabel`	Character scalar giving the label of the selected statistic (will be displayed as the axis label in the plot).
`colName`	Character scalar giving the name of a logical column of `cbTable` to use for filling the bars in the histogram.
`cbName`	Character scalar giving the name of the set of barcodes defined by `colName`, used for labelling the plot legend.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinHistogram(alevin$cbTable, plotVar = "dedupRate",
                    axisLabel = "Deduplication rate",
                    colName = "inFinalWhiteList",
                    cbName = "final whitelist")
```

plotAlevinKneeNbrGenes *Knee plot of the number of detected genes per cell*

Description

Plot the number of detected genes per cell in decreasing order. Only cells contained in the original whitelist are considered.

Usage

```
plotAlevinKneeNbrGenes(cbTable)
```

Arguments

cbTable	data.frame (such as the cbTable returned by readAlevinQC) with the number of detected genes per cell.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinKneeNbrGenes(alevin$cbTable)
```

plotAlevinKneeRaw *Knee plot of raw cell barcode frequencies*

Description

Plot the raw cell barcode frequencies in decreasing order, and indicate a predetermined breakpoint (indicating barcodes included in the original whitelist) using color as well as a label.

Usage

```
plotAlevinKneeRaw(cbTable)
```

Arguments

cbTable	data.frame with raw barcode frequencies (such as the cbTable returned by readAlevinQC).

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinKneeRaw(alevin$cbTable)
```

`plotAlevinQuant` *Panel of plots with quantification summary statistics*

Description

Panel of plots with quantification summary statistics

Usage

```
plotAlevinQuant(
  cbTable,
  colName = "inFinalWhiteList",
  cbName = "final whitelist"
)
```

Arguments

`cbTable`	`data.frame` (such as the `cbTable` returned by `readAlevinQC`) with collapsed barcode frequencies, the total UMI count and the number of detected genes for each cell.
`colName`	Character scalar giving the name of a logical column of `cbTable` to use for coloring the points.
`cbName`	Character scalar giving the name of the set of barcodes defined by `colName`, used for labelling the plot legend.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinQuant(alevin$cbTable, colName = "inFinalWhiteList",
                cbName = "final whitelist")
```

`plotAlevinQuantPairs` *Pairs plot with quantification summary statistics*

Description

Pairs plot with quantification summary statistics

Usage

```
plotAlevinQuantPairs(cbTable, colName = "inFinalWhiteList")
```

Arguments

`cbTable`	`data.frame` (such as the `cbTable` returned by `readAlevinQC`) with collapsed barcode frequencies, the total UMI count and the number of detected genes for each cell.
`colName`	Character scalar giving the name of a logical column of `cbTable` to use for coloring the points.

Value

A ggmatrix object

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                                   package = "alevinQC"))
plotAlevinQuantPairs(alevin$cbTable, colName = "inFinalWhiteList")
```

`readAlevinQC` *Read alevin data required to generate summary report*

Description

Read all alevin output files required to generate the summary report or shiny app.

Usage

```
readAlevinQC(baseDir, customCBList = list())
```

Arguments

`baseDir`	Path to the output directory from the alevin run (should be the directory containing the `alevin` directory).
`customCBList`	Named list with custom set(s) of barcodes to provide summary statistics/plots for, in addition to the whitelists generated by alevin.

Value

A list collecting all necessary information for generating the summary report/shiny app.

Author(s)

Charlotte Soneson

Examples

```
alevin <- readAlevinQC(system.file("extdata/alevin_example_v0.14",
                       package = "alevinQC"))
```

Index

alevinQC-pkg, 2
alevinQCReport, 2
alevinQCShiny, 4

checkAlevinInputFiles, 4

plotAlevinBarcodeCollapse, 5
plotAlevinHistogram, 6
plotAlevinKneeNbrGenes, 7
plotAlevinKneeRaw, 7
plotAlevinQuant, 8
plotAlevinQuantPairs, 9

readAlevinQC, 10