Package ‘ngsReports’

May 4, 2024

Version 2.6.0
Date 2024/02/02
Title Load FastqQC reports and other NGS related files
Description This package provides methods and object classes for parsing
FastQC reports and output summaries from other NGS tools into R.
As well as parsing files, multiple plotting methods have been implemented
for visualising the parsed data.
Plots can be generated as static ggplot objects or interactive plotly
objects.

URL https://github.com/smped/ngsReports
BugReports https://github.com/smped/ngsReports/issues
License LGPL-3
Encoding UTF-8
Depends R (>= 4.2.0), BiocGenerics, ggplot2 (>= 3.5.0), patchwork (>=
1.1.1), tibble (>= 1.3.1)
Imports Biostrings, checkmate, dplyr (>= 1.1.0), forcats, ggdendro,
grDevices (>= 3.6.0), grid, jsonlite, lifecycle, lubridate,
methods, plotly (>= 4.9.4), reshape2, rlang, rmarkdown, scales,
stats, stringr, tidyR, tidyselect (>= 0.2.3), utils, zoo
LazyData true
RoxygenNote 7.3.1
Collate 'AllGenerics.R' 'validationFunctions.R' 'FastpData.R'
'FastpDataList.R' 'FastpFile.R' 'FastqcData.R'
'FastqcDataList.R' 'FastqcFile.R' 'PwfCols.R' 'S4coercion.R'
'TheoreticalGC.R' 'aaa.R' 'data.R' 'errMsg.R' 'estGcDistn.R'
'extract.R' 'fqName.R' 'fqcVersion.R' 'getColours.R' 'getGC.R'
'getModule.R' 'getSummary.R' 'helpers.R' 'importNgsLogs.R'
'importSJ.R' 'isCompressed.R' 'maxAdapterContent.R'
'ngsReports-package.R' 'overRep2Fasta.R' 'path.R'
'plotAdapterContent.R' 'plotAlignmentSummary.R'
'plotAssemblyStats.R' 'plotBaseQuals.R' 'plotDupLevels.R'
'plotFastqcPCA.R' 'plotGcContent.R' 'plotInsertSize.R'
Contents

`plotKmers.R` `plotNContent.R` `plotOverrep.R`
`plotReadTotals.R` `plotSeqContent.R` `plotSeqLengthDistn.R`
`plotSeqQuals.R` `plotSummary.R` `pwf.R` `readTotals.R`

VignetteBuilder knitr

Suggests BiocStyle, Cairo, DT, knitr, pander, readr, testthat, truncnorm

biocViews QualityControl, ReportWriting

RdMacros lifecycle

Roxygen list(markdown = TRUE)

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

git_branch RELEASE_3_19

git_last_commit 935453c

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-03

Author Stevie Pederson [aut, cre] (<https://orcid.org/0000-0001-8197-3303>),
Christopher Ward [aut],
Thu-Hien To [aut]

Maintainer Stevie Pederson <stephen.pederson.au@gmail.com>

Contents

`.addPercent` .......................................................... 4
`.emptyPlot` .......................................................... 4
`.FastpFile-class` .................................................. 5
`.FastqcFile-class` .................................................. 6
`.getXtoolName` ...................................................... 6
`.hidePWFRects` ...................................................... 7
`.makeDendro` ....................................................... 7
`.makeLabels` ......................................................... 8
`.makePwfGradient` .................................................. 9
`.makeSidebar` ....................................................... 10
.parseAdapterRemovalLogs ........................................ 10
.parseBowtieLogs .................................................. 11
.parseBuscoLogs ................................................... 11
.parseCutadaptLogs ................................................ 12
.parseDuplicationMetricsLogs .................................... 12
.parseFeatureCountsLogs ......................................... 13
.parseFlagstatLogs ................................................ 13
.parseHisat2Logs .................................................. 14
.parseMacs2CallpeakLogs ....................................... 15
.parseQuastLogs ................................................... 15
.parseRnaseqcMetricsLogs ....................................... 16
.addPercent

Add a percentage sign to the end of a string

Usage

.addPercent(x)

Arguments

x

Any vector

Value

character vector

Examples

x <- 1:10
ngsReports:::.addPercent(x)

.emptyPlot

Create an empty plot with supplied text

Description

Create an empty plot with supplied text

Usage

.emptyPlot(x)

Details

Create plot using theme_void and only with the supplied text
Value

A ggplot2 object

Examples

ngsReports:::.emptyPlot("This is an empty plot")

---

Description

The .FastpFile Object Class defines a path to the output from the standalone tool fastp. [Experimental]

Arguments

x character(1) denoting a file.path

Details

This class simply refers to a fastp output file after checking for existence and validity (i.e. the correct internal structure). Underlying files are expected to be in json format.

The helper function .FastpFile() is a simple constructor which checks validity and enables construction of other dependent classes.

Value

An object of class .FastqcFile

Slots

path Character vector of length 1 which contains a valid file path.
The .FastqcFile Object Class

Description
The .FastqcFile Object Class defines a path to the output from the standalone tool FastQC. [Stable]

Arguments
x character(1) denoting a file.path

Details
This class simply refers to a fastqc output file after checking for existence and validity (i.e. the correct internal structure). Underlying files can be zipped (*.fastqc.zip) or extracted directories.
The helper function .FastqcFile() is a simple constructor which checks validity and enables construction of other dependent classes.

Value
An object of class .FastqcFile

Slots
path Character vector of length 1 which contains a valid file path.

Examples
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)[1]

# As this is the root structure, we can only call this
# function with an individual file
ff <- ngsReports::.FastqcFile(fl)

.getToolName Identify tool name

Description
Identify tool name for log files after reading in using readLines.

Usage
.getToolName(x, possTypes)
.hidePWFRects

Arguments

x Character vector as output when readLines to a supplied log file

Details

Checks for all the required fields in the lines provided

Value

logical(1)

---

.hidePWFRects Hide PWF tooltips from line plots

Description

Hide tooltips from PWF rectangles in line plots

Usage

.hidePWFRects(x)

Arguments

x plotlyObject$x$data

Value

plotlyObject$x$data

---

.makeDendro Make the dendrogram for heatmap-style plots

Description

Set the clusters for heatmap-style interactive plots

Usage

.makeDendro(df, rowVal, colVal, value)

Arguments

df The data frame to be clustered
rowVal The rows to be clustered
colVal The value which will become column names
value The value to use for the clustering
Value

A dendrogram

Examples

```r
# Get the files included with the package
dir <- system.file("extdata", package = "ngsReports")
fileList <- list.files(dir, pattern = "fastqc.zip", full.names = TRUE)
cols <- c("Filename", "Position", "Illumina_Universal_Adapter")
ac <- getModule(fileList, "Adapter_Content")[cols]
ngsReports:::.makeDendro(df = ac,
  rowVal = "Filename",
  colVal = "Position",
  value = "Illumina_Universal_Adapter")
```

Description

Checks for the presence of labels and returns defaults

Usage

```r
.makeLabels(
  x,
  labels,
  pattern = ".(fast|fq|bam|sam|cram).*",
  col = "Filename",
  ...
)
```

Arguments

- **x**: A data.frame with a column titled "Filename"
- **labels**: Named vector of labels for plotting
- **pattern**: character Regular expression to remove from filenames
- **col**: character Column to use for generating labels
- ... Not used

Details

Takes a named vector of labels and checks for the correct fields. If no vector is supplied, returns the file names missing the specified pattern, which defaults to removing the suffixes fastq(.gz), fq(.gz), bam, sam or cram.
**.makePwfGradient**

**Value**

Named character vector

**Examples**

```r
f <- paste0(c("File1", "File2"), ".fastq")
df <- data.frame(Filename = f, stringsAsFactors = FALSE)
ngsReports:::.makeLabels(df)
```

---

**.makePwfGradient**   
*Construct a gradient using PwfCols*

**Description**

Construct a gradient using PwfCols

**Usage**

```
.makePwfGradient(
  vals,
  pwfCols,
  breaks = c(0, 5, 10, 100),
  passLow = TRUE,
  na.value = "white"
)
```

**Arguments**

- **vals**  
  The values which need to have the scale generated
- **pwfCols**  
  An object of class PwfCols
- **breaks**  
  The breaks for the PWF bins
- **passLow**  
  Is the PASS category at the low or high end of the numeric range
- **na.value**  
  The colour to plot for missing values

**Details**

This constructs a list of arguments for passing to `scale_fill_gradientn()` using the values contained in the fill aesthetic and the supplied breaks for PASS/WARN/FAIL criteria.

**Value**

Returns a ggplot list
.makeSidebar

Shortcut for making the status sidebar

Description
Shortcut for making the status sidebar

Usage

.makeSidebar(status, key, pwfCols, usePlotly = TRUE)

Arguments

status A data frame with columns 'Filename' & 'Status'
key A vector of values corresponding to the Filename column
pwfCols An object of class PwfCols
usePlotly If TRUE, output is a plotly panel

Value
if usePlotly = TRUE, a plotly object. The sidebar for an interactive plot showing PASS/WARN/FAIL status for each file. If usePlotly = FALSE the underlying ggplot object will be returned.

.parseAdapterRemovalLogs

Parse data from Adapter Removal log files

Description
Parse data from Adapter Removal log files

Usage

.parseAdapterRemovalLogs(data, which = 3)

Arguments

data List of lines read using readLines on one or more files
which which element of the log file to return. Can be 1:4, "sequences", "settings", "statistics" or "distribution"

Details
Checks for structure will have been performed
`.parseBowtieLogs`  

**Value**

```
tibble
```

---

**Description**
Parse data from Bowtie log files

**Usage**
```
`.parseBowtieLogs(data, ...)
```

**Arguments**
- `data` : List of lines read using `readLines` on one or more files
- `...` : Not used

**Details**
Checks for structure will have been performed

**Value**
```
data.frame
```

---

`.parseBuscoLogs`  

**Description**
Parse data from BUSCO log files

**Usage**
```
`.parseBuscoLogs(data, ...)
```

**Arguments**
- `data` : List of lines read using `readLines` on one or more files
- `...` : Not used

**Details**
Checks for structure will have been performed
Description
Parse data from cutadapt log files

Usage
.parseCutadaptLogs(data, which = 1)

Arguments
- data: List of lines read using readLines on one or more files
- which: which element of the log file to return. Can be summary, adapter1, adapter2, adapter3 or overview, or any integer in 1:5

Details
Checks for structure will have been performed

Value
tibble

Description
Parse data from Picard duplicationMetrics log files

Usage
.parseDuplicationMetricsLogs(data, which = 1)

Arguments
- data: List of lines read using readLines on one or more files
- which: which element of the log file to return. Can be 1:2, "metrics" or "histogram"
.parseFeatureCountsLogs

Details
Checks for structure will have been performed

Value
tibble

parsesFeatureCountsLogs

Description
Parse data from featureCounts summary files

Usage
.parseFeatureCountsLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used

Details
Checks for structure will have been performed

Value
tibble

.parseFlagstatLogs  Parse data from samtools flagstat files

Description
Parse data from samtools flagstat files

Usage
.parseFlagstatLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used

Details
Checks for structure will have been performed

Value
tibble
Arguments

data  List of lines read using readLines on one or more files
...  Not used

Details

Checks for structure will have been performed

Value

data.frame

Description

Parse data from HISAT2 log files

Usage

.parseHisat2Logs(data, ...)

Arguments

data  List of lines read using readLines on one or more files
...  Not used

Details

Checks for structure will have been performed

Value

data.frame
.parseMacs2CallpeakLogs

Description
Parse data from macs2 callpeak log files

Usage
.parseMacs2CallpeakLogs(data, ...)

Arguments
- data List of lines read using readLines on one or more files
- ... Not used

Details
Checks for structure will have been performed

Value
data.frame

.parseQuastLogs

Description
Parse data from BUSCO log files

Usage
.parseQuastLogs(data, ...)

Arguments
- data List of lines read using readLines on one or more files
- ... Not used

Details
Checks for structure will have been performed

Value
data.frame
.parseRnaseqcMetricsLogs

Parse data from rnaseqc metrics files

Description
Parse data from rnaseqc metrics files

Usage
.parseRnaseqcMetricsLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used

Details
Checks for structure will have been performed

Value
data.frame

.parseSalmonLibFormatCountsLogs

Parse data from salmon lib_format_counts.json files

Description
Parse data from salmon lib_format_counts.json files

Usage
.parseSalmonLibFormatCountsLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used

Details
Checks for structure will have been performed
.parseSalmonMetaInfoLogs

Value
data.frame

Description
Parse data from salmon meta_info.json files

Usage
.parseSalmonMetaInfoLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used

Details
Checks for structure will have been performed

Value
data.frame

.parseStarLogs

Description
Parse data from STAR log files

Usage
.parseStarLogs(data, ...)

Arguments
data List of lines read using readLines on one or more files
... Not used
Details

Checks for structure will have been performed

Value

tibble

---

Description

Parse data from trimmomatic log files

Usage

`.parseTrimmomaticLogs(data, ...)`

Arguments

data List of lines read using readLines on one or more files
... not used

Details

Checks for structure will have been performed

Value

tibble

---

Description

Parse data from umi_tools dedup files

Usage

`.parseUmitoolsDedupLogs(data, ...)`
.prepHeatmap

Arguments

data List of lines read using readLines on one or more files
... Not used

Details

Checks for structure will have been performed

Value

data.frame

Description

Add dendrogram & status bar to ggplot2 heatmap

Usage

.prepHeatmap(x, status, segments, usePlotly, heat_w = 8, pwf, hv = NULL)

Arguments

x a ggplot2 heatmap produced by ngsReports
status a tibble with the columns Filename and Status
segments a dendrogram produced during clustering of samples
usePlotly logical(1)
hv character vector of fields to include in hoverinfo

Value

Either a ggplot2 object assembled using patchwork, or an interactive plotly object
.renderDendro  
*Set up dendrograms for interactive plots*

**Description**
A commonly used (hidden) function for setting up dendrograms for interactive plots. Based on code found at https://plot.ly/ggplot2/ggdendro-dendrograms/

**Usage**
```
.renderDendro(df)
```

**Arguments**
- `df` A `data.frame` as required

**Details**
Create plot using `theme_dendro`

**Value**
A plotly object

---

.splitByTab  
*Split elements of a vector into a data.frame*

**Description**
Split elements of a character vector by the tab separator

**Usage**
```
.splitByTab(x, firstRowToNames = TRUE, tab = "\t")
```

**Arguments**
- `x` A character vector
- `firstRowToNames` logical. Should the first element be used for column names
- `tab` character. The string used to represent the tab symbol

**Details**
This will split a vector into a data.frame checking that every line has the same number of separators. By default the first element will be set as the column names.
This is designed to take input from `readLines()`
.updateThemeFromDots

**Value**

A data frame

**Examples**

```r
x <- c("ColA\tColB", "Value1\tValue2")
ngsReports:::.splitByTab(x, firstRowToNames = TRUE)
ngsReports:::.splitByTab(x, firstRowToNames = FALSE)
```

### Description

Add custom theme elements from dotArgs

### Usage

```r
.updateThemeFromDots(p, ...)
```

### Arguments

- `p` ggplot object
- `...` Standard dot arguments

### Value

ggplot2 object

---

**estGcDistn**

*Estimate a GC Content Distribution From Sequences*

**Description**

Generate a GC content distribution from sequences for a given read length and fragment length

**Usage**

```r
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```r
## S4 method for signature 'ANY'
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```r
## S4 method for signature 'character'
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```r
## S4 method for signature 'DNASTringSet'
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```
Arguments

- **x**: DNAStringSet or path to a fasta file
- **n**: The number of reads to sample
- **rl**: Read Lengths to sample
- **f1**: The mean of the fragment lengths sequenced
- **fragSd**: The standard deviation of the fragment lengths being sequenced
- **bins**: The number of bins to estimate
- **...**: Not used

Details

The function takes the supplied object and returns the theoretical GC content distribution. Using a fixed read length essentially leads to a discrete distribution so the bins argument is used to define the number of bins returned. This defaults to 101 for 0 to 100% inclusive.

The returned values are obtained by interpolating the values obtained during sampling. This avoids returned distributions with gaps and jumps as would be obtained setting readLengths at values not in multiples of 100.

Based heavily on https://github.com/mikelove/fastqcTheoreticalGC

Value

A tibble with two columns: GC_Content and Freq denoting the proportion of GC and frequency of occurrence respectively

Examples

```r
faDir <- system.file("extdata", package = "ngsReports")
faFile <- list.files(faDir, pattern = "fasta", full.names = TRUE)
df <- estGcDistn(faFile, n = 200)
```

---

**FastpData-class**

The FastpData Object Class

Description

The FastpData Object Class [Experimental]

Usage

`FastpData(x)`

Arguments

- **x**: Path to a single zip archive or extracted folder for a individual fastp report.
**Details**

This object class is the main object required for generating plots and tables. Instantiation will first check for a .json file with the correct data structure, and will then parse all the data into R as a FastpData object. Fastp modules are contained as individual slots, which can be viewed using slotNames. Sub-modules are also contained within many larger modules with modules being based on the sections within a fastp html report.

Individual modules can be returned using the function `getModule()` and specifying which module/sub-module is required. See `getModule()` for more details.

**Value**

An object of class FastpData

**Slots**

- **Summary** Contains three submodules 1) Before_filtering, 2) After_filtering and 3) Filtering_result.
- All values presented in the initial table for individual fastp reports are contained in other sections of the report.
- **Adapters** Contains a tibble with all data from this module
- **Duplication** Contains a tibble with all duplication results
- **Insert_size** Contains a tibble with all insert size estimates
- **Before_filtering,After_filtering** The modules can be selected for either Read1 or Read2 paired logical(1) indicating whether the file is from paired-end sequencing
- **command** character(1) with the executed command
- **version** character(1) with the fastp version being used
- **path** Path to the Fastp report

---

**FastpDataList-class**

The FastpDataList Object Class

---

**Description**

The FastpDataList Object Class [Stable]

**Usage**

`FastpDataList(x)`

**Arguments**

- **x** Character vector of file paths specifying paths to fastp.json.gz output

**Value**

An object of class FastpDataList
Slots

... this can either be a single character vector of paths to fastp files, or several instances of .Fastp-File objects

---

Description

The FastqcData Object Class [Stable]

Usage

FastqcData(x)

Arguments

x Path to a single zip archive or extracted folder for a individual FastQC report.

Details

This object class is the main object required for generating plots and tables. Instantiation will first test for a compressed file (or extracted directory) with the correct data structure, and will then parse all the data into R as a FastqcData object. FastQC modules are contained as individual slots, which can be viewed using slotNames. Individual modules can be returned using the function getModule() and specifying which module is required. See getModule() for more details.

Value

An object of class FastqcData

Slots

Summary Summary of PASS/WARN/FAIL status for each module
Basic_Statistics The Basic_Statistics table from the top of a FastQC html report
Per_base_sequence_quality The underlying data from the Per_base_sequence_quality module
Per_sequence_quality_scores The underlying data from the Per_sequence_quality_scores module
Per_base_sequence_content The underlying data from the Per_base_sequence_content module
Per_sequence_GC_content The underlying data from the Per_sequence_GC_content module
Per_base_N_content The underlying data from the Per_base_N_content module
Sequence_Length_Distribution The underlying data from the Sequence_Length_Distribution module
Sequence_Duplication_Levels  The underlying data from the Sequence_Duplication_Levels module
Overrepresented_sequences  The underlying data from the Overrepresented_sequences module
Adapter_Content  The underlying data from the Adapter_Content module
Kmer_Content  The underlying data from the Kmer_Content module
Total_Deduplicated_Percentage  Estimate taken from the plot data for Sequence_Duplication_Levels. Only included in later versions of FastQC
version  The version of FastQC used for generation of the report (if available)
path  Path to the FastQC report

Examples

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)[1]

# Load the FASTQC data as a FastqcData object
fd <- FastqcData(fl)
fd
```

---

FastqcDataList-class  The FastqcDataList Object Class

Description

The FastqcDataList Object Class [Stable]

Usage

```
FastqcDataList(x)
```

Arguments

x  Character vector of file paths specifying paths to FastQC reports

Value

An object of class FastqcDataList

Slots

... this can either be a single character vector of paths to FASTQC files, or several instances of .FastqcFile objects
Examples

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
fdl
```

### fqcVersion, FastqcData-method

*Get the FASTQC version*

#### Description

Get the FASTQC version used to generate the initial files.

#### Usage

```r
## S4 method for signature 'FastqcData'
fqcVersion(object)

## S4 method for signature 'FastqcDataList'
fqcVersion(object)

## S4 method for signature 'ANY'
fqcVersion(object)
```

#### Arguments

- `object` An object of class `FastqcData` or `FastqcDataList`

#### Value

A character vector (FastqcData), or tibble (FastqcDataList).

#### Examples

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Get the FASTQC version
fqcVersion(fdl)
```
**Return the Underlying Fastq File Names from Fastqc/Fastp Objects**

**Description**

Return the Underlying Fastq File Names from Fastqc/Fastp Objects

**Usage**

\[fqName(object)\]

## S4 method for signature 'ANY'
fqName(object)

## S4 method for signature 'FastqcData'
fqName(object)

## S4 method for signature 'FastqcDataList'
fqName(object)

fqName(object) <- value

## S4 replacement method for signature 'FastqcData'
fqName(object) <- value

## S4 replacement method for signature 'FastqcDataList'
fqName(object) <- value

## S4 method for signature 'FastpData'
fqName(object)

## S4 method for signature 'FastpDataList'
fqName(object)

**Arguments**

- **object** An object able to extract an Fastq name from
- **value** Replacement value for fqName

**Value**

Returns the names of the Fastq files the FastQC report was generated from, without any preceding directories.
Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
fqName(fdl)

nm <- paste0(letters[seq_along(fdl)], ".fq")
fqName(fdl) <- nm
fqName(fdl)

---

**gcAvail**

*List Genomes or Transcriptomes with Theoretical GC Content*

Description

List available genomes or transcriptomes in a TheoreticalGC object

Usage

gcAvail(object, type)

## S4 method for signature 'TheoreticalGC'
gcAvail(object, type)

Arguments

object An object of class TheoreticalGC
type character indicating either Genome or Transcriptome

Details

An object of class TheoreticalGC can hold the theoretical GC content for one or more species, for either the genome or transcriptome. This function checks which species are available in the given object, for either the genome or transcriptome, as supplied to the parameter type.

Value

A tibble object

Examples

gcAvail(gcTheoretical, "Genome")
Theoretical GC content

Description

This object contains the theoretical GC content for each provided species, for both the genome and transcriptome, where available.

Usage

gcTheoretical

Format

An object of class `TheoreticalGC` of length 1.

Details

The object is defined with the S4 class `TheoreticalGC`. Species for which information is available can be found using the command `gcAvail(gcTheoretical)` and selecting the appropriate type. Metadata is accessible using `mData(gcTheoretical)`.

All GC content was calculated using code from https://github.com/mikelove/fastqcTheoreticalGC using BSgenome packages. This provides a default set of GC content data for common organisms generated using 100bp reads/fragments and 1e6 reads.

See Also

gcAvail

Examples

```r
## Check which genomes are included
gcAvail(gcTheoretical, "Genome")

## Check which transcriptomes are included
gcAvail(gcTheoretical, "Transcriptome")
```
getColours

Work with objects of class PwfCols

Description
Get and modify colours from objects of class PwfCols

Usage

## S4 method for signature 'PwfCols'
getColours(object)

## S4 method for signature 'PwfCols'
setColours(object, PASS, WARN, FAIL, MAX)

## S4 method for signature 'PwfCols'
setAlpha(object, alpha)

Arguments

object An object of class PwfCols
PASS The colour denoting PASS on all plots, in rgb format
WARN The colour denoting WARN on all plots, in rgb format
FAIL The colour denoting FAIL on all plots, in rgb format
MAX The colour denoting the limit of values in rgb format
alpha Numeric(1). Ranges from 0 to 1 by default, but can also be on the range 0 to 255.

Details
Use getColours to obtain the colours in an object of class PwfCols.
These can be modified using the functions setColours and setAlpha

Value
getColours will return a character vector of colours corresponding to PASS/WARN/FAIL
setColours will return an object of class PwfCols
setAlpha will return an object of class PwfCols

Examples
getColours(pwf)

# How to add transparency
pwf2 <- setAlpha(pwf, 0.1)
getColours(pwf2)
getGC

getGC

Get Theoretical GC content

Description

Get the GC content data from a TheoreticalGC object

Usage

getGC(object, name, type)

## S4 method for signature 'ANY'
getGC(object, type)

## S4 method for signature 'TheoreticalGC'
getGC(object, name, type)

Arguments

object
An object of class Theoretical GC

name
The Name of the species in 'Gspecies' format, e.g. Hsapiens

type
The type of GC content. Can only be either "Genome" or "Transcriptome"

Value

A tibble object

Examples

getGC(gcTheoretical, name = "Hsapiens", type = "Genome")

getModule,FastqcData-method

Retrieve a given module from a Fastqc* Object

Description

Retrieve a specific module from a Fastqc* object as a data.frame
Usage

## S4 method for signature 'FastqcData'
getModule(object, module)

## S4 method for signature 'FastqcDataList'
getModule(object, module)

## S4 method for signature 'ANY'
getModule(object, module)

## S4 method for signature 'FastpData'
getModule(object, module)

## S4 method for signature 'FastpDataList'
getModule(object, module)

Arguments

object Can be a FastqcData, fastqcDataList, or simply a character vector of paths

module The requested module as contained in a FastQC report. Possible values are Summary, Basic_Statistics, Per_base_sequence_quality, Per_tile_sequence_quality, Per_sequence_quality_scores, Per_base_sequence_content, Per_sequence_GC_content, Per_base_N_content, Sequence_Length_Distribution, Sequence_Duplication_Levels, Overrepresented_sequences, Adapter_Content, Kmer_Content, Total_Deduplicated_Percentage. Note that spelling and capitalisation is exactly as contained within a FastQC report, with the exception that spaces have been converted to underscores. Partial matching is implemented for this argument.

Details

This function will return a given module from a Fastqc* object as a data.frame. Note that each module will be it's own unique structure, although all will return a data.frame

Value

A single tibble containing module-level information from all FastQC reports contained in the Fastqc* object.

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Extract the Summary module, which corresponds to the PASS/WARN/FAIL flags
getModule(fdl, "Summary")
getSummary/.FastqcFile-method

The Basic_Statistics module corresponds to the table at the top of each FastQC report
getModule(fd1, "Basic_Statistics")

getSummary/.FastqcFile-method

Get the summary information from Fastqc Files

Description
Read the information from the summary.txt files in each .FastqcFile

Usage

## S4 method for signature '.FastqcFile'
getSummary(object)

## S4 method for signature 'ANY'
getSummary(object)

## S4 method for signature 'FastqcData'
getSummary(object)

## S4 method for signature 'FastqcDataList'
getSummary(object)

Arguments

object Can be a FastqcData, FastqcDataList object or a vector of paths to unparsed FastQC reports.

Details
This simply extracts the summary of PASS/WARN/FAIL status for every module as defined by the tool FastQC for each supplied file.

Value
A tibble containing the PASS/WARN/FAIL status for each module, as defined in a FastQC report.

Examples

# Get the files included with the package
cDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(cDir, pattern = "fastqc.zip", full.names = TRUE)
# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Return a tibble/tibble with the raw information
getSummary(fdl)

---

**importNgsLogs**

**Import Various NGS-related log files**

**Description**

[Maturing] Imports NGS-related log files such as those generated from stderr.

**Usage**

`importNgsLogs(x, type = "auto", which, stripPaths = TRUE)`

**Arguments**

- **x** character. Vector of filenames. All log files must be of the same type. Duplicate file paths will be silently ignored.
- **type** character. The type of file being imported. Can be one of bowtie, bowtie2, hisat2, star, flagstat, featureCounts, duplicationMetrics, cutadapt, umitoolsDedup, maccs2Callpeak, adapterRemoval, rnaseqcMetrics, quast, salmonLibFormatCounts, salmonMetaInfo or busco. Defaults to type = "auto" which will automatically detect the file type for all implemented types.
- **which** Which element of the parsed object to return. Ignored in all file types except when type is set to duplicationMetrics, cutadapt or adapterRemoval. See details for possible values. To return all elements, set this value to 'all'
- **stripPaths** logical(1). Remove paths from the Filename column

**Details**

Imports one or more log files as output by tools such as: bowtie, bowtie2, featureCounts, Hisat2, STAR, salmon picard MarkDuplicates, cutadapt, flagstat, maccs2Callpeak, Adapter Removal, trimmomatic, rnaseqcMetrics, quast or busco. autoDetect can be used to detect the log type by parsing the file.

The featureCounts log file corresponds to the `counts.out.summary`, not the main `counts.out` file. Whilst most log files return a single tibble, some are more complex with multiple modules.

adapterRemoval can return one of four modules (which = 1:4). When calling by name, the possible values are sequences, settings, statistics or distribution. Partial matching is implemented.

cutadapt can return one of five modules (which = 1:5). When calling by name the possible modules are summary, adapter1, adapter2, adapter3 or overview. Note that adapter2/3 may be missing from these files depending on the nature of your data. If cutadapt log files are obtained using report=minimal, all supplied log files must be of this format and no modules can be returned.
duplicationMetrics will return either the metrics of histogram. These can be requested by setting which as 1 or 2, or naming either module.

**Value**

A tibble. Column names are broadly similar to the text in supplied files, but have been modified for easier handling under R naming conventions.

**Examples**

```r
f <- c("bowtiePE.txt", "bowtieSE.txt")
bowtieLogs <- system.file("extdata", f, package = "ngsReports")
df <- importNgsLogs(bowtieLogs, type = "bowtie")
```

---

**Description**

Import the SJ.out.tab files produced by STAR

**Usage**

```r
importSJ(x, stripPaths = TRUE)
```

**Arguments**

- `x` vector of file paths to SJ.out.tab files
- `stripPaths` logical(1) Remove directory prefixes from the file paths in x

**Details**

Imports one or more splice-junction output files as produced by STAR. If all are located in separated directories with identical names, be sure to set the argument `stripPaths = FALSE`

All co-ordinates are 1-based, in keeping with the STAR manual

**Value**

A tibble

**Author(s)**

Stephen Pederson stephen.pederson@adelaide.edu.au
isCompressed

Check to see if a file is compressed

Description

Check to see if a file, or vector of files is compressed

Usage

isCompressed(path, type = c("zip", "gzip"), verbose = FALSE)

Arguments

- **path**: The path to one or more files
- **type**: The type of compression to check for. Currently only ZIP/GZIP files have been implemented.
- **verbose**: logical/integer Determine the level of output to show as messages

Details

Reads the first four bytes from the local file header. If the file is a .ZIP file, this should match the magic number PK\003\004.

This function assumes that the first thing in a zip archive is the .ZIP entry with the local file header signature. ZIP files containing a self-extracting archive may not exhibit this structure and will return FALSE

Value

A logical vector

Examples

# Get the files included with the package
fileDir <- system.file("extdata", package = "ngsReports")
allFiles <- list.files(fileDir, pattern = "zip$", full.names = TRUE)
isCompressed(allFiles)
maxAdapterContent  

Get the maximum Adapter Content

Description

Get the maximum Adapter Content across one or more FASTQC reports

Usage

maxAdapterContent(x, asPercent = TRUE)

Arguments

x Can be a .FastqcFile, FastqcData, FastqcDataList or path
asPercent logical. Format the values as percentages with the added % symbol

Details

This will extract the Adapter_Content module from the supplied object, and provide a tibble with the final value for each file.

Value

A tibble object containing the percent of reads with each adapter type at the final position

Examples

# Get the files included with the package
tmp <- system.file("extdata", package = "ngsReports")
fl <- list.files(tmp, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
dl <- FastqcDataList(fl)

# Get the maxAdapterContent
maxAdapterContent(dl)
### mData

**Extract Metadata for TheoreticalGC objects**

**Description**

Extract Metadata for TheoreticalGC objects

**Usage**

```r
mData(object)
```

## S4 method for signature 'TheoreticalGC'

mData(object)

**Arguments**

- `object` An object of class Theoretical GC

**Value**

A tibble object

**Examples**

```r
mData(gcTheoretical)
```

### overRep2Fasta

**Write fasta of Over-Represented sequences.**

**Description**

Output overrepresented sequences to disk in fasta format.

**Usage**

```r
overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)
```

## S4 method for signature 'ANY'

overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)

## S4 method for signature 'FastqcData'

overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)

## S4 method for signature 'FastqcDataList'

overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)
Arguments

x Can be a FastqcData or FastqcDataList

path Path to export the fasta file to. Reverts to a default in the working directory if not supplied

n The number of sequences to output

labels An optional named factor of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.

noAdapters logical. Remove any sequences identified as possible adapters or primers by FastQC

... Used to pass any alternative patterns to remove from the end of filenames

Details

Fasta will contain Filename, Possible Source, Percent of total reads

Value

Exports to a fasta file, and returns the fasta information invisibly

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Export the top10 Overrepresented Sequences as a single fasta file
faOut <- file.path(tempdir(), "top10.fa")
overRep2Fasta(fdl, path = faOut)

path

Return the File Paths from an object

Description

Return the File Paths from an object
Usage

## S4 method for signature '.FastqcFile'
path(object)

## S4 method for signature 'FastqcData'
path(object)

## S4 method for signature 'FastqcDataList'
path(object)

## S4 method for signature '.FastpFile'
path(object)

## S4 method for signature 'FastpData'
path(object)

## S4 method for signature 'FastpDataList'
path(object)

Arguments

object An object of class .FastqcFile

Details

Obtains the file.path for objects of multiple classes

Value

A character vector of the file paths to the underlying FastQC reports

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
path(fdl)

plotAdapterContent Draw an Adapter Content Plot

Description

Draw an Adapter Content Plot across one or more FASTQC reports
Usage

plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  ...
)

## S4 method for signature 'ANY'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  ...
)

## S4 method for signature 'FastqcData'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols, showPwf = TRUE,
  warn = 5,
  fail = 10,
  scaleColour = NULL,
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastqcDataList'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  showPwf = TRUE,
  warn = 5,
  fail = 10,
  plotType = c("heatmap", "line"),
  adapterType = "Total",
  cluster = FALSE,
  dendoagram = FALSE,
  heat_w = 8L,
scaleFill = NULL,
scaleColour = NULL,
plotlyLegend = FALSE,
...
)

## S4 method for signature 'FastpData'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  scaleFill = NULL,
  plotlyLegend = FALSE,
  plotTheme = theme_get(),
  ...
)

## S4 method for signature 'FastpDataList'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  pwfCols,
  showPwf = FALSE,
  warn = 5,
  fail = 10,
  cluster = FALSE,
  dendrogram = FALSE,
  scaleFill = NULL,
  plotTheme = theme_get(),
  heat_w = 8L,
  ...
)

Arguments

x Can be a FastqcData, a FastqcDataList or character vector of file paths
usePlotly logical. Output as ggplot2 (default) or plotly object.
labels An optional named vector of labels for the file names. All filenames must be present in the names.
pattern regex used to trim the ends of all filenames for plotting
... Used to pass additional attributes to theme() for FastQC objects and geoms for Fastp objects
pwfCols Object of class PwfCols() containing the colours for PASS/WARN/FAIL
showPwf logical(1) Show PASS/WARN/FAIL status as would be included in a standard FastQC report
plotAdapterContent

warn, fail

The default values for warn and fail are 5 and 10 respectively (i.e. percentages)

plotlyLegend

logical(1) Show legend when choosing interactive plots. Ignored for heatmaps

plotType

can only take the values plotType = "heatmap" or plotType = "line"

adapterType

A regular expression matching the adapter(s) to be plotted. To plot all adapters summed, specify adapterType = "Total". This is the default behaviour.

cluster

logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

dendrogram

logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

heat_w

Width of the heatmap relative to other plot components

scaleFill, scaleColour

scale_fill_* and scale_colour_* objects

plotTheme

Set theme elements by passing a theme

Details

This extracts the Adapter_Content module from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

When x is a single or FastqcData object line plots will always be drawn for all adapters. Otherwise, users can select line plots or heatmaps. When plotting more than one fastqc file, any undetected adapters will not be shown.

An interactive version of the plot can be made by setting usePlotly as TRUE

Value

A standard ggplot2 object, or an interactive plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotAdapterContent(fdl)

# Also subset the reads to just the R1 files
r1 <- grepl("R1", fqName(fdl))
plotAdapterContent(fdl[r1])

# Plot just the Universal Adapter
# and change the y-axis using ggplot2::scale_y_continuous
plotAdapterContent(fdl, adapterType = "Illumina Universal", plotType = "line") +
plotAlignmentSummary

Description
Plot a summary of alignments from a set of log files

Usage

plotAlignmentSummary(
  x,
  type = c("star", "bowtie", "bowtie2", "hisat2"),
  usePlotly = FALSE,
  stripPaths = TRUE,
  asPercent = FALSE,
  ...
  fill = c("red", "yellow", "blue", rgb(0, 0.5, 1))
)

Arguments

  x          Paths to one or more alignment log files
  type       The aligner used. Can be one of star, bowtie, bowtie2 or hisat2
  usePlotly  logical. If TRUE an interactive plot will be generated.
  stripPaths logical(1). Remove paths from the Filename column
  asPercent  Show alignments as percentages, with the alternative (FALSE) being the total
             number of reads If FALSE a ggplot object will be output
  ...        Used to pass additional attributes to theme() and between methods
  fill       Colours used to fill the bars. Passed to scale_fill_manual.

Details
Loads a set of alignment log files and creates a default plot. Implemented aligners are bowtie,
bowtie2, Hisat2 and STAR.

Value
A ggplot2 object, or a plotly object
Examples

```r
f <- c("bowtie2PE.txt", "bowtie2SE.txt")
bowtie2logs <- system.file("extdata", f, package = "ngsReports")
plotAlignmentSummary(bowtie2logs, "bowtie2")
```

---

**plotAssemblyStats**  
*Plot a summary of assembly logs*

**Description**

Plot a summary of assembly stats from a set of log files

**Usage**

```r
plotAssemblyStats(
  x,
  type = c("quast", "busco"),
  usePlotly = FALSE,
  plotType = c("bar", "paracoord"),
  ...
)
```

**Arguments**

- **x**  
  Paths to one or more log files

- **type**  
  The tool used. Can be one of quast or busco

- **usePlotly**  
  logical. If TRUE an interactive plot will be generated. If FALSE a ggplot object will be output

- **plotType**  
  character. Plot type to output, one of bar or paracoord.

- **...**  
  Used to pass additional attributes to theme() and between methods

**Details**

Load a set of assembly log files and creates a default plot. Implemented tools are quast and BUSCO. quast will plot a parralel coordinate plot of some assembly statistics BUSCO will plot a stacked barplot of completeness statistics

**Value**

A ggplot2 object, or a plotly object
Examples

# get quast log filenames
quastFiles <- system.file("extdata",
c("quast1.tsv", "quast2.tsv"), package = "ngsReports")

# The default plot
plotAssemblyStats(quastFiles)

plotBaseQuals

Plot the Base Qualities for each file

Description

Plot the Base Qualities for each file as separate plots

Usage

plotBaseQuals(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'ANY'
plotBaseQuals(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'FastqcData'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
  showPwf = TRUE,
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastqcDataList'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  warn = 25,
  fail = 20,
showPwf = TRUE,
boxWidth = 0.8,
plotType = c("heatmap", "boxplot"),
plotValue = "Mean",
cluster = FALSE,
dendrogram = FALSE,
nc = 2,
heat_w = 8L,
...
}

## S4 method for signature 'FastpData'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
pwfCols,
  warn = 25,
  fail = 20,
  showPwf = FALSE,
  module = c("Before_filtering", "After_filtering"),
  reads = c("read1", "read2"),
  readsBy = c("facet", "linetype"),
  bases = c("A", "T", "C", "G", "mean"),
scaleColour = NULL,
  plotTheme = theme_get(),
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastpDataList'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
pwfCols,
  warn = 25,
  fail = 20,
  showPwf = FALSE,
  module = c("Before_filtering", "After_filtering"),
  plotType = "heatmap",
  plotValue = c("mean", "A", "T", "C", "G"),
scaleFill = NULL,
  plotTheme = theme_get(),
  cluster = FALSE,
dendrogram = FALSE,
heat_w = 8L,
...
)

Arguments

x Can be a FastqcData, FastqcDataList or character vector of file paths
usePlotly logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
labels An optional named vector of labels for the file names. All filenames must be present in the names.
pattern Regex to remove from the end of the Fastp report and Fastq file names
... Used to pass additional attributes to theme() and between methods
pwfCols Object of class PwfCols() to give colours for pass, warning, and fail values in plot
warn, fail The default values for warn and fail are 30 and 20 respectively (i.e. percentages)
boxWidth set the width of boxes when using a boxplot
showPwf Include the Pwf status colours
plotlyLegend logical(1) Show legend for interactive plots. Only called when drawing line plots
plotType character Can be either "boxplot" or "heatmap"
plotValue character Type of data to be presented. Can be any of the columns returned by the appropriate call to getModule()
cluster logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
nc numeric. The number of columns to create in the plot layout. Only used if drawing boxplots for multiple files in a FastqcDataList
heat_w Relative width of any heatmap plot components
module Select Before and After filtering when using a FastpDataList
reads Create plots for read1, read2 or all when using a FastpDataList
readsBy If paired reads are present, separate using either linetype or by facet
bases Which bases to include on the plot
scaleColour ggplot discrete colour scale, passed to lines
plotTheme theme object
scaleFill ggplot2 continuous scale. Passed to heatmap cells
plotDupLevels

Details

When acting on a FastqcDataList, this defaults to a heatmap using the mean Per_base_sequence_quality score. A set of plots which replicate those obtained through a standard FastQC html report can be obtained by setting plotType = "boxplot", which uses facet_wrap to provide the layout as a single ggplot object.

When acting on a FastqcData object, this replicates the Per_base_sequence_quality plots from FastQC with no faceting.

For large datasets, subsetting by R1 or R2 reads may be helpful.

An interactive plot can be obtained by setting usePlotly = TRUE.

Value

A standard ggplot2 object or an interactive plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot for multiple libraries is a heatmap
plotBaseQuals(fdl)

# The default plot for a single library is the standard boxplot
plotBaseQuals(fdl[[1]])

# FastpData objects have qualities by base
fp <- FastpData(system.file("extdata/fastp.json.gz", package = "ngsReports"))
plotBaseQuals(fp, plotTheme = theme(plot.title = element_text(hjust = 0.5)))

---

plotDupLevels  

Plot the combined Sequence_Duplication_Levels information

Description

Plot the Sequence_Duplication_Levels information for a set of FASTQC reports
Usage

plotDupLevels(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam)\.*", ...)  

## S4 method for signature 'ANY'
plotDupLevels(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam)\.*", ...)  

## S4 method for signature 'FastqcData'
plotDupLevels(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam)\.*", pwfCols, warn = 20, fail = 50, showPwf = TRUE, plotlyLegend = FALSE, lineCol = c("red", "blue"), lineWidth = 1, ...)  

## S4 method for signature 'FastqcDataList'
plotDupLevels(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam)\.*", pwfCols, warn = 20, fail = 50, showPwf = TRUE, plotlyLegend = FALSE, plotlyLegend = FALSE, plotType = c("heatmap", "line"), cluster = FALSE, dendrogram = FALSE, heatCol = hcl.colors(50, "inferno"), heat_w = 8, ...)  

## S4 method for signature 'FastpData'
plotDupLevels(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam)\.*",
plotDupLevels

pwfCols,
warn = 20,
fail = 50,
showPwf = FALSE,
maxLevel = 10,
lineCol = "red",
barFill = "dodgerblue4",
barCol = barFill,
plotlyLegend = FALSE,
plotTheme = theme_get(),
...
)

## S4 method for signature 'FastpDataList'
plotDupLevels(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
pwfCols,
  warn = 20,
  fail = 50,
  showPwf = FALSE,
  plotlyLegend = FALSE,
  plotType = c("bar", "heatmap"),
  barFill = "blue",
  barCol = "blue",
  cluster = FALSE,
  dendrogram = FALSE,
  scaleFill = NULL,
  plotTheme = theme_get(),
  heat_w = 8,
  maxLevel = 10,
  ...
)

Arguments

  x                  Can be a FastqcData, FastqcDataList or file path
  usePlotly         logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
  labels            An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
  pattern           regex to remove from the end of fastp & fastq file names
  ...               Used to pass additional attributes to theme() and between methods
  pwfCols           Object of class PwfCols() to give colours for pass, warning, and fail values in the plot
warn, fail

The default values for warn and fail are 20 and 50 respectively (i.e. percentages).

showPwf

logical(1) Show PWF rectangles in the background.

plotlyLegend

logical(1) Show legend for line plots when using interactive plots.

lineCol, lineWidth

Colours and width of lines drawn.

deduplication

Plot Duplication levels 'pre' or 'post' deduplication. Can only take values "pre" and "post".

plotType

Choose between "heatmap" and "line".

cluster

logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering.

dendrogram

logical. Plot will automatically be clustered if TRUE.

heatCol

Colour palette used for the heatmap.

heat_w

Relative width of the heatmap relative to other plot components.

maxLevel

The maximum duplication level to plot. Beyond this level, all values will be summed.

barFill, barCol

Colours for bars when calling geom_col().

plotTheme

theme object. Applied after a call to theme_bw().

scaleFill

Discrete scale used to fill heatmap cells.

Details

This extracts the Sequence_Duplication_Levels from the supplied object and generates a ggplot2 object, with a set of minimal defaults. For multiple reports, this defaults to a heatmap with block sizes proportional to the percentage of reads belonging to that duplication category.

If setting usePlotly = FALSE, the output of this function can be further modified using standard ggplot2 syntax. If setting usePlotly = TRUE an interactive plotly object will be produced.

Value

A standard ggplot2 or plotly object.

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Draw the default plot for a single file
plotDupLevels(fdl[[1]])
plotFastqcPCA

Draw a PCA plot for Fast QC modules

Description

Draw a PCA plot for Fast QC modules across multiple samples [Experimental]

Usage

plotFastqcPCA(
  x,
  module = "Per_sequence_GC_content",
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  sz = 4,
  groups,
  ...
)

## S4 method for signature 'ANY'
plotFastqcPCA(
  x,
  module = "Per_sequence_GC_content",
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  sz = 4,
  groups,
  ...
)

## S4 method for signature 'character'
plotFastqcPCA(
  x,
  module = "Per_sequence_GC_content",
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  sz = 4,
  groups,
  ...
)

## S4 method for signature 'FastqcDataList'
plotFastqcPCA(
  x,
module = "Per_sequence_GC_content",
usePlotly = FALSE,
labels,
pattern = ".(fast|fq|bam).*",
sz = 4,
groups,
pc = c(1, 2),
...
)

Arguments

x Can be a FastqcDataList or character vector of file paths
module character vector containing the desired FastQC module (eg. c("Per_base_sequence_quality", "Per_base_sequence_content"))
usePlotly logical. Output as ggplot2 (default) or plotly object.
labels An optional named vector of labels for the file names. All file names must be present in the names of the vector.
pattern Regex to remove from the end of any filenames
sz The size of the text labels
groups Optional factor of the same length as x. If provided, the plot will be coloured using this factor as the defined groups. Ellipses will also be added to the final plot.
... Used to pass additional attributes to theme() and between methods
pc The two components to be plotted

Details

This carries out PCA on a single FastQC module and plots the output using either ggplot or plotly. Current modules for PCA are Per_base_sequence_quality, Per_sequence_quality_scores, Per_sequence_GC_content, Per_base_sequence_content, and Sequence_Length_Distribution.

If a factor is provided in the groups argument, this will be applied to the plotting colours and ellipses will be drawn using these groups. Only the labels will be plotted using geom_text()

Value

A standard ggplot2 object, or an interactive plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
grp <- as.factor(gsub(".+\(\[12\]\).*", "\"1\", fqName(fdl)))
plotGcContent

Plot the Per Sequence GC Content

Description

Plot the Per Sequence GC Content for a set of FASTQC files

Usage

plotGcContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).\*", ...)  

## S4 method for signature 'ANY'
plotGcContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).\*", ...)  

## S4 method for signature 'FastqcData'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCobject,
  plotlyLegend = FALSE,
  Fastafile,
  n = 1e+06,
  counts = FALSE,
  scaleColour = NULL,
  lineCols = c("red3", "black"),
  linetype = 1,
  linewidth = 0.5,
  ...
)

## S4 method for signature 'FastqcDataList'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  Fastafile,
  n = 1e+06,
species = "Hsapiens",
GCObject,
Fastafile,
n = 1e+06,
plotType = c("heatmap", "line", "cdf"),
cluster = FALSE,
dendrogram = FALSE,
heat_w = 8,
pwfCols,
showPwf = TRUE,
scaleFill = NULL,
scaleColour = NULL,
plotlyLegend = FALSE,
lineCols = RColorBrewer::brewer.pal(12, "Paired"),
linetype = 1,
linewidth = 0.5,

## S4 method for signature 'FastpData'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".*\.(fast|fq|bam).\*",
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCobject,
  Fastafile,
  n = 1e+06,
  plotType = "bar",
  scaleFill = NULL,
  plotlyLegend = FALSE,
  plotTheme = theme_get(),
  ...
)

## S4 method for signature 'FastpDataList'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".*\.(fast|fq|bam).\*",
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCobject,
plotGeContent

Fastafie,  
n = 1e+06,  
plotType = "bar",  
scaleFill = NULL,  
plotTheme = theme_get(),  
plotlyLegend = FALSE,  
...  
)

Arguments

x  
Can be a FastqcData, FastqcDataList or character vector of file paths

usePlotly  
logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly

labels  
An optional named vector of labels for the file names.

pattern  
Pattern to remove from the end of filenames

...  
Used to pass various potting parameters to themes and geoms.

theoreticalGC  
logical default is FALSE to give the true GC content, set to TRUE to normalize values of GC_Content by the theoretical values using gcTheoretical(). Species must be specified. For Fastqc* objects, the entire distributions will be used, whereas for the Fastp* objects, only the expected mean value is shown as a horizontal line.

gcType  
character Select type of data to normalize GC content against. Accepts either "Genome" (default) or "Transcriptome".

species  
character if gcTheory is TRUE it must be accompanied by a species. Species currently supported can be obtained using mData(gcTheoretical)

GCObject  
an object of class GCTheoretical. Defaults to the gcTheoretical object supplied with the package

plotlyLegend  
logical(1) Show legend on interactive line plots

Fastafie  
a fasta file contains DNA sequences to generate theoretical GC content

n  
number of simulated reads to generate theoretical GC content from Fastafie

counts  
logical. Plot the counts from each file if counts = TRUE, otherwise frequencies will be plotted. Ignored if calling the function on a FastqcDataList.

scaleColour  
ggplot2 scale for line colours

lineCols, linetype, linewidth  
Line colour type and width for observed and theoretical GC lines

plotType  
Takes values "line", "heatmap" or "cdf"

cluster  
logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

dendrogram  
logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

heat_w  
Relative width of any heatmap plot components
plotInsertSize

Description

Plot the insert size distribution from one of Fastp reports

pwfCols Object of class PwfCols() to give colours for pass, warning, and fail values in plot
showPwf logical(1) Show Pwf Status on the plot
scaleFill ggplot2 scale for filling heatmap cells or bars
plotTheme theme object

Details

Makes plots for GC_Content. When applied to a single FastqcData object a simple line plot will be drawn, with Theoretical GC content overlaid if desired.

When applied to multiple FastQC reports, the density at each GC content bin can be shown as a heatmap by setting theoreticalGC = FALSE. By default the difference in observed and expected theoretical GC is shown. Species and genome/transcriptome should also be set if utilising the theoretical GC content.

As an alternative to a heatmap, a series of overlaid distributions can be shown by setting plotType = "line".

Can produce a static ggplot2 object or an interactive plotly object.

Value

A ggplot2 or plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot for a FastqcDataList
plotGcContent(fdl)

# Plot a single FastqcData object
plotGcContent(fdl[[1]])
plotInsertSize

Usage

plotInsertSize(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'FastpData'
plotInsertSize(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  plotType = c("histogram", "cumulative"),
  counts = FALSE,
  plotTheme = theme_get(),
  expand.x = 0.01,
  expand.y = c(0, 0.05),
  ...
)

## S4 method for signature 'FastpDataList'
plotInsertSize(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  plotType = c("heatmap", "line", "cumulative"),
  plotTheme = theme_get(),
  scaleFill = NULL,
  scaleColour = NULL,
  cluster = FALSE,
  dendrogram = FALSE,
  heat_w = 8,
  ...
)

Arguments

  x          A FastpData or FastpDataList object
  usePlotly  logical. Generate an interactive plot using plotly
  labels     An optional named vector of labels for the file names. All file names must be
              present in the names of the vector.
  pattern     Regex to remove from the end of any filenames
  ...         Passed to geom* functions during plotting
  plotType    Determine the plot type. Options vary with the input structure
  counts      logical(1) Plot read counts, or percentages (default)
  plotTheme   a theme object
  expand.x, expand.y  Axis expansions
scaleFill: Continuous scale used to fill heatmap cells. Defaults to the "inferno" palette.
scaleColour: Discrete scale for adding line colours.
cluster: logical default FALSE. If set to TRUE, data will be clustered using hierarchical clustering.
dendrogram: logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE the dendrogram will be displayed.
heat_w: Width of the heatmap relative to other plot components.

Details
Takes a Fastp os a set of Fastp reports and plot insert size distributions. Plots can be drawn as cumulative totals or the default histograms for a single report, and as boxplots or heatmaps for a set of reports.

Value
A ggplot or plotly object.

Examples
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastp.json.gz", full.names = TRUE)
fp <- FastpData(fl)
plotInsertSize(
  fp, counts = TRUE, fill = "steelblue4",
  plotTheme = theme(plot.title = element_text(hjust = 0.5))
)
plotInsertSize(fp, plotType = "cumulative")

plotKmers

Description
Plot Overrepresented Kmers

Usage
plotKmers(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'ANY'
plotKmers(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'FastqcData'
plotKmers(
Arguments

x Can be a FastqcData, FastqcDataList or file paths
usePlotly logical. Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly.

labels An optional named vector of labels for the file names. All filenames must be present in the names.

pattern regex to drop from the end of filenames

... Used to pass parameters to theme for FastqcData objects and to geoms for Fast-pData objects

n numeric. The number of Kmers to show.

linewidth Passed to geom_line()

plotlyLegend Show legend for interactive plots

pal The colour palette. If the vector supplied is less than n, grDevices::colorRampPalette() will be used

cluster logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

dendrogram logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

pwfCols Object of class PwfCols() to give colours for pass, warning, and fail values in the plot

showPwf Show the PASS/WARN/FAIL status

scaleFill, scaleColour ggplot2 scales to be used for colour palettes

heatCol Colour palette used for the heatmap. Default is inferno from the viridis set of palettes

heat_w Relative width of any heatmap plot components

module The module to obtain data from when using a FastpData object

reads Either read1 or read2. Only used when using a FastpData object

readsBy Strategy for visualising both read1 and read2. Can be set to show each set of reads by facet, or within the same plot taking the mean of the enrichment above mean, or the difference in the enrichment above mean

trans Function for transforming the count/mean ratio. Set as NULL to use the ratio without transformation

plotTheme theme object

**Details**

As the Kmer Content module present in FastQC reports is relatively uninformative, and omitted by default in later versions of FastQC, these are rudimentary plots.

Plots for FastqcData objects replicate those contained in a FastQC report, whilst the heatmap generated from FastqcDataList objects simply show the location and abundance of over-represented Kmers.

**Value**

A standard ggplot2 object or an interactive plotly object
Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
plotKmers(fdl[[1]])

# Use a FastpData object
fl <- system.file("extdata", "fastp.json.gz", package = "ngsReports")
fp <- FastpData(fl)
plotKmers(fp, size = 2)
plotKmers(
  fp, reads = "read1", size = 2, trans = NULL,
  scaleFill = scale_fill_gradient(low = "white", high = "black")
)

plotNContent

---

## Description

Draw an N Content Plot across one or more FastQC reports

## Usage

plotNContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

#### S4 method for signature 'ANY'

plotNContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

#### S4 method for signature 'FastqcData'

plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  warn = 5,
  fail = 20,
  showPwf = TRUE,
  ...,
  lineCol = "red"
)

#### S4 method for signature 'FastqcDataList'
plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  warn = 5,
  fail = 20,
  showPwf = TRUE,
  cluster = FALSE,
  dendrogram = FALSE,
  heat_w = 8,
  scaleFill = NULL,
  ...
)

## S4 method for signature 'FastpData'
plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  module = c("Before_filtering", "After_filtering"),
  moduleBy = c("facet", "colour", "linetype"),
  reads = c("read1", "read2"),
  readsBy = c("facet", "colour", "linetype"),
  scaleColour = NULL,
  scaleLine = NULL,
  plotTheme = theme_get(),
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastpDataList'
plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  module = c("Before_filtering", "After_filtering"),
  reads = c("read1", "read2"),
  scaleFill = NULL,
  plotTheme = theme_get(),
  cluster = FALSE,
  dendrogram = FALSE,
  heat_w = 8,
  ...
)
**Arguments**

- **x**: Can be a `FastqcData`, `FastqcDataList` or file paths.
- **usePlotly**: logical. Output as ggplot2 (default) or plotly object.
- **labels**: An optional named vector of labels for the file names. All filenames must be present in the names.
- **pattern**: Regex used to trim the end of filenames.
- **...**: Used to pass additional attributes to theme() for `FastqcData` objects and to geom* calls for FastpData-based objects.
- **pWfCols**: Object of class `PwfCols()` containing the colours for PASS/WARN/FAIL.
- **warn, fail**: The default values for warn and fail are 5 and 10 respectively (i.e. percentages).
- **showPwf**: logical(1) Show the PASS/WARN/FAIL status.
- **lineCol**: Line colours.
- **cluster**: logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering.
- **dendrogram**: logical redundant if `cluster` is FALSE if both `cluster` and `dendrogram` are specified as TRUE then the dendrogram will be displayed.
- **heat_w**: Relative width of any heatmap plot components.
- **scaleFill, scaleColour, scaleLine**: ggplot2 scale objects.
- **module**: Used for Fastp* structures to show results before or after filtering.
- **moduleBy, readsBy**: How to show each module or set of reads on the plot.
- **reads**: Show plots for read1, read2 or both.
- **plotTheme**: theme object.
- **plotlyLegend**: logical(1) Show legend on interactive plots.

**Details**

This extracts the N_Content from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

When `x` is a single `FastqcData` object line plots will always be drawn for all Ns. Otherwise, users can select line plots or heatmaps.

**Value**

A standard ggplot2 object, or an interactive plotly object.
Examples

```r
## Using a Fastp Data object
fl <- system.file("extdata/fastp.json.gz", package = "ngsReports")
fp <- FastpData(fl)
plotNContent(fp)
plotNContent(fp, pattern = "_.001.+",
moduleBy = "colour", scaleColour = scale_colour_brewer(palette = "Set1"),
plotTheme = theme(
    legend.position = 'inside', legend.position.inside = c(0.99, 0.99),
    legend.justification = c(1, 1), plot.title = element_text(hjust = 0.5))
)
)
```

---

**plotOverrep**  
*Plot a summary of Over-represented Sequences*

---

**Description**

Plot a summary of Over-represented Sequences for a set of FASTQC reports

**Usage**

```r
plotOverrep(
    x,
    usePlotly = FALSE,
    labels,
    pattern = ".(fast|fq|bam).*",
    pwfCols,
    ...
)
```

```r
## S4 method for signature 'ANY'
plotOverrep(
    x,
    usePlotly = FALSE,
    labels,
    pattern = ".(fast|fq|bam).*",
    pwfCols,
    ...
)
```

```r
## S4 method for signature 'character'
plotOverrep(
    x,
    usePlotly = FALSE,
    ...)
plotOverrep

labels,
pattern = ".(fast|fq|bam)\.*",
pwfCols,
...
)

## S4 method for signature 'FastqcData'
plotOverrep(
x,
usePlotly = FALSE,
labels,
pattern = "\.(fast|fq|bam)\.*",
pwfCols,
n = 10,
expand.x = c(0, 0, 0.05, 0),
expand.y = c(0, 0.6, 0, 0.6),
plotlyLegend = FALSE,
...
)

## S4 method for signature 'FastqcDataList'
plotOverrep(
x,
usePlotly = FALSE,
labels,
pattern = "\.(fast|fq|bam)\.*",
pwfCols,
showPwf = TRUE,
cluster = FALSE,
dendrogram = FALSE,
scalefill = NULL,
paletteName = "Set1",
panel_w = 8,
expand.x = c(0, 0, 0.05, 0),
expand.y = rep(0, 4),
...
)

Arguments

x Can be a FastqcData, FastqcDataList or file paths
usePlotly logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
labels An optional named factor of labels for the file names. All filenames must be present in the names.
pattern Regex to remove from the end of any filenames
pwfCols Object of class PwfCols() containing the colours for PASS/WARN/FAIL
plotOverrep

... Used to pass additional attributes to theme() and between methods

- **n** The number of sequences to plot from an individual file

- **expand.x, expand.y** Output from expansion() or numeric vectors of length 4. Passed to scale_*_continuous()

- **plotlyLegend** Show legend on interactive plots

- **showPwf** Show PASS/WARN/FAIL status on the plot

- **cluster** logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

- **dendrogram** logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

- **scaleFill** ggplot scale object

- **paletteName** Name of the palette for colouring the possible sources of the overrepresented sequences. Must be a palette name from RColorBrewer. Ignored if specifying the scaleFill separately

- **panel_w** Width of main panel on output

**Details**

Percentages are obtained by simply summing those within a report. Any possible double counting by FastQC is ignored for the purposes of a simple approximation.

Plots generated from a FastqcData object will show the top n sequences grouped by their predicted source & coloured by whether the individual sequence would cause a WARN/FAIL.

Plots generated from a FastqcDataList group sequences by predicted source and summarise as a percentage of the total reads.

**Value**

A standard ggplot2 object

**Examples**

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# A brief summary across all FastQC reports
plotOverrep(fdl)
```
plotReadTotals

Draw a barplot of read totals

Description

Draw a barplot of read totals

Usage

plotReadTotals(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).\*", ...)

## S4 method for signature 'ANY'
plotReadTotals(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).\*", ...)

## S4 method for signature 'FastqcDataList'
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  duplicated = TRUE,
  bars = c("stacked", "adjacent"),
  vertBars = TRUE,
  divBy = 1,
  barCols = c("red", "blue"),
  expand.y = c(0, 0.02),
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastpDataList'
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
  adjPaired = TRUE,
  divBy = 1e+06,
  scaleFill = NULL,
  labMin = 0.05,
  status = TRUE,
  labelVJ = 0.5,
  labelFill = "white",
  plotTheme = theme_get(),
  vertBars = FALSE,
  plotlyLegend = FALSE,
  expand.y = c(0, 0.05),
  ...)
Arguments

x Can be a FastqData, FastqcDataList or file paths

usePlotly logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly

labels An optional named vector of labels for the file names. All filenames must be present in the names.

pattern Regex used to trim the end of filenames

... Used to pass additional attributes to theme()

duplicated logical(1). Include deduplicated read total estimates to plot charts

bars If duplicated = TRUE, show unique and deduplicated reads as "stacked" or "adjacent".

vertBars logical(1) Show bars as vertical or horizontal

divBy Scale read totals by this value. The default shows the y-axis in millions for FastpDataList objects, but does not scale FastQC objects, for the sake of backwards compatability

barCols Colours for duplicated and unique reads.

expand.y Passed to ggplot2::expansion for the axis showing totals

plotlyLegend logical(1) Show legend on interactive plots

adjPaired Scale read totals by 0.5 when paired

scaleFill ScaleDiscrete function to be applied to the plot

labMin Only show labels for filtering categories higher than this values as a proportion of reads. Set to any number > 1 to turn off labels

status logical(1) Include read status in the plot

labelVJ Relative vertical position to labels within each bar.

labelFill Passed to geom_label

plotTheme theme to be added to the plot

Details

Draw a barplot of read totals using the standard ggplot2 syntax. The raw data from readTotals() can otherwise be used to manually create a plot.

Duplication levels are based on the value shown on FASTQC reports at the top of the DeDuplicated-Totals plot, which is known to be inaccurate. As it still gives a good guide as to sequence diversity it is included as the default. This can be turned off by setting duplicated = FALSE.

For FastpDataList objects, duplication statistics are not part of the default module containing ReadTotals. However, the status of reads and the reason for being retained or filtered is, and as such these are shown instead of duplication statistics.
plotSeqContent

Value

Returns a ggplot or plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Plot the Read Totals showing estimated duplicates
plotReadTotals(fdl)

# Plot the Read Totals without estimated duplicates
plotReadTotals(fdl, duplicated = FALSE)

plotSeqContent

Plot the per base content as a heatmap

Description

Plot the Per Base content for a set of FASTQC files.

Usage

plotSeqContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'ANY'
plotSeqContent(x, usePlotly = FALSE, labels, pattern = ".(fast|fq|bam).*", ...)

## S4 method for signature 'FastqcData'
plotSeqContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  bases = c("A", "T", "C", "G"),
  scaleColour = NULL,
  plotTheme = theme_get(),
  plotlyLegend = FALSE,
  expand.x = 0.02,
  expand.y = c(0, 0.05),
  ...
)
## S4 method for signature 'FastqcDataList'
plotSeqContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).*",
  pwfCols,
  showPwf = TRUE,
  plotType = c("heatmap", "line", "residuals"),
  scaleColour = NULL,
  plotTheme = theme_get(),
  cluster = FALSE,
  dendrogram = FALSE,
  heat_w = 8,
  plotlyLegend = FALSE,
  nc = 2,
  ...
)

## S4 method for signature 'FastpData'
plotSeqContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).*",
  module = c("Before_filtering", "After_filtering"),
  reads = c("read1", "read2"),
  readsBy = c("facet", "linetype"),
  moduleBy = c("facet", "linetype"),
  bases = c("A", "T", "C", "G", "N", "GC"),
  scaleColour = NULL,
  scaleLine = NULL,
  plotlyLegend = FALSE,
  plotTheme = theme_get(),
  expand.x = 0.02,
  expand.y = c(0, 0.05),
  ...
)

## S4 method for signature 'FastpDataList'
plotSeqContent(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).*",
  module = c("Before_filtering", "After_filtering"),
  moduleBy = c("facet", "linetype"),
  reads = c("read1", "read2"),
  ...)
readsBy = c("facet", "linetype"),
bases = c("A", "T", "C", "G", "N", "GC"),
showPwf = FALSE,
pwfCols,
warn = 10,
fail = 20,
plotType = c("heatmap", "line", "residuals"),
plotlyLegend = FALSE,
scaleColour = NULL,
scaleLine = NULL,
plotTheme = theme_get(),
cluster = FALSE,
dendrogram = FALSE,
heat_w = 8,
expand.x = c(0.01),
expand.y = c(0, 0.05),
nc = 2,
...
)

Arguments

x
usePlotly logical. Generate an interactive plot using plotly
labels An optional named vector of labels for the file names. All file names must be present in the names of the vector.
pattern Regex to remove from the end of any filenames
... Used to pass additional attributes to plotting geoms
bases Which bases to draw on the plot. Also becomes the default plotting order by setting these as factor levels
scaleColour Discrete colour scale as a ggplot ScaleDiscrete object If not provided, will default to scale_colour_manual
plotTheme theme object to be applied. Note that all plots will have theme_bw theme applied by default, as well as any additional themes supplied here
plotlyLegend logical(1) Show legends for interactive plots. Ignored for heatmaps
expand.x, expand.y Passed to expansion in the x- and y-axis scales respectively
pwfCols Object of class PwfCols() to give colours for pass, warning, and fail values in plot
showPwf Show PASS/WARN/FAIL categories as would be defined in a FastQC report
plotType character. Type of plot to generate. Must be "line", "heatmap" or "residuals"
cluster logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
**plotSeqContent**

- `heat_w`: Relative width of any heatmap plot components
- `nc`: Specify the number of columns if plotting a FastqcDataList as line plots. Passed to `facet_wrap`.
- `module`: Fastp Module to show. Can only be Before/After_filtering
- `reads`: Which set of reads to show
- `readsBy, moduleBy`: When plotting both R1 & R2 and both modules, separate by either linetype or linetype
- `scaleLine`: Discrete scale_linetype object. Only relevant if plotting values by linetype
- `warn, fail`: Default values for WARN and FAIL based on FastQC reports. Only applied to heatmaps for FastpDataList objects

**Details**

Per base sequence content (%A, %T, %G, %C), is shown as four overlaid heatmap colours when plotting from multiple reports. The individual line plots are able to be generated by setting `plotType = "line"`, and the layout is determined by `facet_wrap` from ggplot2.

Individual line plots are also generated when plotting from a single FastqcData object.

If setting `usePlotly = TRUE` for a large number of reports, the plot can be slow to render. An alternative may be to produce a plot of residuals for each base, produced by taking the position-specific mean for each base.

**Value**

A ggplot2 object or an interactive plotly object

**Examples**

```r
# Get the files included with the package
gctag <- system.file("extdata", package = "ngsReports")
fl <- list.files(gctag, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotSeqContent(fdl)

fp <- FastpData(system.file("extdata/fastp.json.gz", package = "ngsReports"))
plotSeqContent(fp)
plotSeqContent(fp, moduleBy = "linetype", bases = c("A", "C", "G", "T"))
```
plotSeqLengthDistn

Plot the Sequence Length Distribution

Description

Plot the Sequence Length Distribution across one or more FASTQC reports

Usage

plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).\*",
  ...)

## S4 method for signature 'ANY'
plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).\*",
  ...)

## S4 method for signature 'character'
plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).\*",
  ...)

## S4 method for signature 'FastqcData'
plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  pattern = "\.(fast|fq|bam).\*",
  counts = TRUE,
  plotType = c("line", "cdf"),
  expand.x = c(0, 0.2, 0, 0.2),
  plotlyLegend = FALSE,
  colour = "red",
  ...
### plotSeqLengthDistrn

```
## S4 method for signature 'FastqcDataList'
plotSeqLengthDistrn(
    x,
    usePlotly = FALSE,
    labels,
    pattern = ".\.(fast|fq|bam).\*",
    counts = FALSE,
    plotType = c("heatmap", "line", "cdf"),
    cluster = FALSE,
    dendrogram = FALSE,
    heat_w = 8,
    pwfCols,
    showPwf = TRUE,
    scaleFill = NULL,
    scaleColour = NULL,
    heatCol = hcl.colors(50, "inferno"),
    plotlyLegend = FALSE,
    ...)
```

#### Arguments

- **x**: Can be a FastqcData, FastqcDataList or file paths
- **usePlotly**: logical. Output as ggplot2 or plotly object.
- **labels**: An optional named vector of labels for the file names. All filenames must be present in the names.
- **pattern**: Regex to remove from the end of any filenames
- **counts**: logical. Should distributions be shown as counts or frequencies (percentages)
- **plotType**: character. Can only take the values plotType = "heatmap" plotType = "line" or plotType = "cdf"
- **expand.x**: Output from expansion() or numeric vector of length 4. Passed to scale_x_discrete
- **plotlyLegend**: logical(1) Show legend for interactive line plots
- **colour**: Line colour
- **cluster**: logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
- **dendrogram**: logical redundant if cluster and usePlotly are FALSE. If both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
- **heat_w**: Relative width of any heatmap plot components
- **pwfCols**: Object of class PwfCols() to give colours for pass, warning, and fail values in plot
- **showPwf**: logical(1) Show PASS/WARN/FAIL status
plotSeqQuals

scaleFill, scaleColour
Optional ggplot scale objects

heatCol
The colour scheme for the heatmap

Details
This extracts the Sequence Length Distribution from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

A cdf plot can also be generated to provide guidance for minimum read length in some NGS workflows, by setting plotType = "cdf". If all libraries have reads of identical lengths, these plots may be less informative.

An alternative interactive plot is available by setting the argument usePlotly = TRUE.

Value
A standard ggplot2 object, or an interactive plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Plot as a frequency plot using lines
plotSeqLengthDistn(fdl)

# Or plot the cdf
plotSeqLengthDistn(fdl, plotType = "cdf")

plotSeqQuals

Plot the Per Sequence Quality Scores

Description
Plot the Per Sequence Quality Scores for a set of FASTQC reports

Usage

plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).\*",
)
plotSeqQuals

pwfCols,
...
)

## S4 method for signature 'ANY'
plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  ...
)

## S4 method for signature 'character'
plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  ...
)

## S4 method for signature 'FastqcData'
plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
  showPwf = TRUE,
  counts = FALSE,
  alpha = 0.1,
  warn = 30,
  fail = 20,
  colour = "red",
  plotlyLegend = FALSE,
  ...
)

## S4 method for signature 'FastqcDataList'
plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pattern = ".(fast|fq|bam).*",
  pwfCols,
counts = FALSE,
alpha = 0.1,
warn = 30,
fail = 20,
showPwf = TRUE,
plotType = c("heatmap", "line"),
dendrogram = FALSE,
cluster = FALSE,
scaleFill = NULL,
heatCols = hcl.colors(100, "inferno"),
heat_w = 8,
scaleColour = NULL,
plotlyLegend = FALSE,
...)

Arguments

x Can be a FastqcData, FastqcDataList or path

usePlotly logical. Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly

labels An optional named vector of labels for the file names. All file names must be present in the names of the vector.

pattern Regex to remove from the end of any filenames

pwfCols Object of class PwfCols() containing the colours for PASS/WARN/FAIL

... Used to pass various potting parameters to theme. Can also be used to set size and colour for box outlines.

showPwf logical(1) Show PASS/WARN/FAIL status

counts logical. Plot the counts from each file if counts = TRUE, otherwise the frequencies will be plotted

alpha set alpha for line graph bounds

warn, fail The default values for warn and fail are 5 and 10 respectively (i.e. percentages)

colour Colour for single line plots

plotlyLegend logical(1) Show legend for interactive line plots

plotType character. Can only take the values plotType = "heatmap" or plotType = "line"

dendrogram logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

cluster logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

scaleFill, scaleColour ggplot2 scales

heatCols Colour palette for the heatmap

heat_w Relative width of any heatmap plot components
plotSummary

Details

Plots the distribution of average sequence quality scores across the set of files. Values can be plotted
either as counts (counts = TRUE) or as frequencies (counts = FALSE).

Any faceting or scale adjustment can be performed after generation of the initial plot, using the
standard methods of ggplot2 as desired.

Value

A standard ggplot2 object, or an interactive plotly object

Examples

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotSeqQuals(fdl)

# Also subset the reads to just the R1 files
r1 <- grepl("R1", fqName(fdl))
plotSeqQuals(fdl[r1])

plotSummary

**Plot the PASS/WARN/FAIL information**

Description

Extract the PASS/WARN/FAIL summaries and plot them

Usage

plotSummary(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,
  ...
)

## S4 method for signature 'ANY'
plotSummary("
### plotSummary

```r
plotSummary(x,
    usePlotly = FALSE,
    labels,
    pwfCols,
    cluster = FALSE,
    dendrogram = FALSE,
    ...
)
```

#### Arguments

- **x**: Can be a `FastqcData`, `FastqcDataList` or character vector of file paths.
- **usePlotly**: logical. Generate an interactive plot using plotly.
- **labels**: An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
- **pwfCols**: Object of class `PwfCols()` containing the colours for PASS/WARN/FAIL.
- **cluster**: logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering.
- **dendrogram**: logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
- **gridlineWidth**: Passed to `geom_hline` and `geom_vline` to determine width and colour of gridlines.
- **gridlineCol**: Passed to `geom_hline` and `geom_vline` to determine width and colour of gridlines.
Details

This uses the standard ggplot2 syntax to create a three colour plot. The output of this function can be further modified using the standard ggplot2 methods if required.

Value

A ggplot2 object (usePlotly = FALSE) or an interactive plotly object (usePlotly = TRUE)

Examples

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Check the overall PASS/WARN/FAIL status
plotSummary(fdl)
```

---

**pwf**

Colours for PASS/WARN/FAIL

Description

Default colours for PASS/WARN/FAIL

Usage

`pwf`

Format

An object of class `PwfCols` of length 1.

Details

`pwf` is an object of class `PwfCols` supplied with the package and used as the default colouring. Colours correspond approximately to PASS, WARN and FAIL from the FASTQC reports, with the additional colour (MAX) included to indicate an extreme FAIL. In order, these colours in the default vector are green (`rgb(0, 0.8, 0)`), yellow (`rgb(0.9, 0.9, 0.2)`), red (`rgb(0.8, 0.2, 0.2)`) and white (`rgb(1, 1, 1)`)

Examples

```r
# Make a pie chart showing the default colours
pie(rep(1,4), labels = names(pwf), col = getColours(pwf))
```
PwfCols-class

The PwfCols class and associated methods

Description

Define the PwfCols class and associated methods

Details

This is an object of with four colours in components named PASS, WARN, FAIL and MAX. Used to indicate these categories as defined on the standard plots from fastqc.

Value

An S4 object of class PwfCols

Slots

PASS A vector of length 1, defining the colour for PASS in rgb format. Defaults to rgb(0, 0.8, 0)

WARN A vector of length 1, defining the colour for WARN in rgb format. Defaults to rgb(0.9, 0.9, 0.2)

FAIL A vector of length 1, defining the colour for FAIL in rgb format. Defaults to rgb(0.8, 0.2, 0.2)

MAX A vector of length 1, defining the colour for an extreme FAIL or NA in rgb format. Defaults to rgb(1, 1, 1)

readTotals

Get the read totals

Description

Get the read totals from one or more FASTQC reports

Usage

readTotals(x)

Arguments

x Can be a FastqcData, FastqcDataList, FastpData, FastpDataList or file paths

Value

A tibble with the columns Filename and Total_Sequences
# Examples

```r
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Print the read totals
readTotals(fdl)
```

## summariseOverrep

### Summarise Overrepresented Sequences

Summarise the Overrepresented sequences found in one or more QC files

### Usage

```r
summariseOverrep(x, ...)

## S4 method for signature 'FastpData'
summariseOverrep(x, step = c("Before", "After"), min_count = 0, ...)

## S4 method for signature 'FastpDataList'
summariseOverrep(x, min_count = 0, vals = c("Count", "Percentage"),
                  fn = c("mean", "sum", "max"),
                  pattern = ".*_",
                  by = c("reads", "sequence"),
                  ...)
```

## Description

Summarise the Overrepresented sequences found in one or more QC files.
summariseOverrep

## S4 method for signature 'FastqcData'
summariseOverrep(
  x,
  min_count = 0,
  vals = c("Count", "Percentage"),
  fn = c("mean", "sum", "max"),
  pattern = ".*",
  by = "Filename",
  ...
)

Arguments

x An object of a suitable class

... Not used

step Can be 'Before', 'After' or both to obtain data from the Before_filtering or After_filtering modules

min_count Filter sequences with counts less than this value, both before and after filtering

vals Values to use for creating summaries across multiple files. For FastpDataList objects these can be "count" and/or "rate", whilst for FastqcDataList objects these values can be "Count" and/or "Percentage"

fn Functions to use when summarising values across multiple files

by character vector of columns to summarise by. See dplyr::summarise

pattern Regular expression to filter the Possible_Source column by

Details

This function prepares a useful summary of all over-represented sequences as reported by either fastp or FastQC

Value

A tibble

Tibble columns will vary between Fastp*, FastqcDataList and FastqcData objects. Calling this function on list-type objects will attempt to summarise the presence each over-represented sequence across all files.

In particular, FastqcData objects will provide the requested summary statistics across all sequences within a file

Examples

## For operations on a FastpData object
f <- system.file("extdata/fastp.json.gz", package = "ngsReports")
fp <- FastpData(f)
summariseOverrep(fp, min_count = 100)

## Applying the function to a FastqcDataList
TheoreticalGC-class

Description

Contains Theoretical GC content for a selection of species

Details

Estimates are able to be retained for genomic and transcriptomic sequences. Values are stored as frequencies.

Value

An object of class TheoreticalGC

Slots

Genome A data.frame containing theoretical GC content for genomic sequences
Transcriptome A data.frame containing theoretical GC content for transcriptomic sequences
mData A data.frame containing metadata about all species in the object

Examples

```r
## How to form an object using your own fasta file
faDir <- system.file("extdata", package = "ngsReports")
faFile <- list.files(faDir, pattern = "fasta", full.names = TRUE)
gen_df <- estGcDistn(faFile, n = 200)
gen_df <- dplyr::rename(gen_df, Athaliana = Freq)
mData_df <-
data.frame(Name = "Athaliana", Genome = TRUE, Transcriptome = FALSE)
tr_df <- data.frame()
myGC <- new("TheoreticalGC", Genome = gen_df, Transcriptome = tr_df, mData = mData_df)
```
writeHtmlReport

Write an HTML Summary Report

Description

Compiles an HTML report using a supplied template

Usage

```r
writeHtmlReport(
  fastqcDir,
  template,
  outDir,
  usePlotly = TRUE,
  species = "Hsapiens",
  gcType = c("Genome", "Transcriptome"),
  nOver = 30,
  targetsDF,
  overwrite = FALSE,
  quiet = TRUE
)
```

Arguments

- **fastqcDir**: A directory containing zipped, or extracted FastQC reports
- **template**: The template file which will be copied into `fastqcDir`
- **outDir**: The directory to write the compiled document to
- **usePlotly**: Generate interactive plots?
- **species**: Species/closely related species of sequenced samples
- **gcType**: Is the data "Transcriptomic" or "Genomic" in nature?
- **nOver**: The maximum number of Overrepresented Sequences to show
- **targetsDF**: A data.frame with at least two columns named `Filename` and `Label`. The file-names should match the original fastq files, and the labels should be simply alternative labels for these files for convenience.
- **overwrite**: logical. Overwrite any previous copies of the template file in the destination directory
- **quiet**: logical. Show or hide markdown output in the Console.

Details

This will take a user supplied template, or the file supplied with the package and create an HTML summary of all standard FASTQC plots for all files in the supplied directory.
Value

Silently returns TRUE and will output a compiled HTML file from the supplied Rmarkdown template file

Examples

```r
## Not run:
packageDir <- system.file("extdata", package = "ngsReports")
fileList <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)
# Copy these files to tempdir() to avoid overwriting
# any files in the package directory
file.copy(fileList, tempdir(), overwrite = TRUE)
writeHtmlReport(fastqcDir = tempdir())

## End(Not run)
```

Description

Extract elements from FastqcDataList Object

Usage

```r
## S4 method for signature 'FastqcDataList,numeric,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,character,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,logical,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,ANY,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastpDataList,numeric,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastpDataList,character,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastpDataList,logical,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastpDataList,ANY,missing'
x[i, j, ..., drop = TRUE]
```
## S4 method for signature 'FastqcDataList,ANY,missing'

\[ \[i, j, \ldots, \text{drop} = \text{TRUE}] \]

### Arguments

- **x**: A `FastqcDataList` or `FastpDataList`
- **i**: character, logical or integer vector
- **j**: not used
- **...**: not used
- **drop**: not used

### Details

Extract elements in a consistent manner with R conventions

### Value

Will return a subset of the original object following the standard rules for subsetting objects

### Examples

```r
# Get the files included with the package
dir <- system.file("extdata", package = "ngsReports")
files <- list.files(dir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
data <- FastqcDataList(files)

# Subsetting using the standard methods
data[[1]]
data[[[1]]]
```
Index

* datasets
  gcTheoretical, 29
  pwf, 82

* internal
  .FastpFile-class, 5
  .FastqcFile-class, 6
  .addPercent, 4
  .emptyPlot, 4
  .getToolName, 6
  .hidePWFRects, 7
  .makeDendro, 7
  .makeLabels, 8
  .makePwfGradient, 9
  .makeSidebar, 10
  .parseAdapterRemovalLogs, 10
  .parseBowtieLogs, 11
  .parseBuscoLogs, 11
  .parseCutadaptLogs, 12
  .parseDuplicationMetricsLogs, 12
  .parseFeatureCountsLogs, 13
  .parseFlagstatLogs, 13
  .parseFlagstatLogs, 13
  .parseHisat2Logs, 14
  .parseMacs2CallpeakLogs, 15
  .parseQuastLogs, 15
  .parseSalmonLibFormatCountsLogs, 16
  .parseSalmonMetaInfoLogs, 17
  .parseStarLogs, 17
  .parseTrimmomaticLogs, 18
  .parseUmitoolsDedupLogs, 18
  .prepHeatmap, 19
  .renderDendro, 20
  .splitByTab, 20
  .updateThemeFromDots, 21
INDEX

[,FastqcDataList,numeric,missing-method, 88
[,FastqcDataList,character,missing-method, 88
[,FastqcDataList,numeric,missing-method, 88
[,FastqcDataList,logical,missing-method, 88
[,FastqcDataList,numeric,missing-method, 88
dplyr::summarise, 85
estGcDistn, 21
estGcDistn,ANY-method (estGcDistn), 21
estGcDistn,character-method (estGcDistn), 21
estGcDistn,DNAStringSet-method (estGcDistn), 21
expansion, 73
facet_wrap, 74
FastpData (FastpData-class), 22
FastpData-class, 22
FastpDataList (FastpDataList-class), 23
FastpDataList-class, 23
FastqcData (FastqcData-class), 24
FastqcData-class, 24
FastqcDataList (FastqcDataList-class), 25
FastqcDataList-class, 25
fqCversion
(fqcVersion,FastqcData-method), 26
fqCversion,ANY-method (fqCversion,FastqcData-method), 26
fqCversion,FastqcData-method, 26
fqCversion,FastqcDataList-method (fqCversion,FastqcData-method), 26
fqName, 27
fqName,ANY-method (fqName), 27
fqName,FastpData-method (fqName), 27
fqName,FastpDataList-method (fqName), 27
fqName,FastqcData-method (fqName), 27
fqName,FastqcDataList-method (fqName), 27
fqName<- (fqName), 27
fqName<= (fqName), 27
fqName-,FastqcData-method (fqName), 27
gcAvail, 28
gcAvail,TheoreticalGC-method (gcAvail), 28
gcTheoretical, 29
gcTheoretical(), 57
gem_label, 70
getColours, 30
getColours,PwfCols-method (getColours), 30
getGC, 31
getGC,ANY-method (getGC), 31
getGC,TheoreticalGC-method (getGC), 31
getModule
(getModule,FastqcData-method), 31
getModule(), 23, 24
getModule,ANY-method (getModule,FastqcData-method), 31
getModule,FastpData-method (getModule,FastqcData-method), 31
getModule,FastpDataList-method (getModule,FastqcData-method), 31
getModule,FastqcData-method, 31
getModule,FastqcDataList-method (getModule,FastqcData-method), 31
getSummary
(getSummary,.FastqcFile-method), 33
getSummary,.FastqcFile-method, 33
getSummary,ANY-method (getSummary,.FastqcFile-method), 33
getSummary,FastqcData-method (getSummary,.FastqcFile-method), 33
getSummary,FastqcDataList-method (getSummary,.FastqcFile-method), 33
ggplot2::expansion, 70
importNgsLogs, 34
plotNContent, FastqcDataList-method (plotNContent), 63
plotOverrep, 66
plotOverrep, ANY-method (plotOverrep), 66
plotOverrep, character-method (plotOverrep), 66
plotOverrep, FastqcData-method (plotOverrep), 66
plotOverrep, FastqcDataList-method (plotOverrep), 66
plotOverrep, FASTSEQ-method (plotOverrep), 66
plotReadTotals, 69
plotReadTotals, ANY-method (plotReadTotals), 69
plotReadTotals, FastpDataList-method (plotReadTotals), 69
plotSeqContent, 71
plotSeqContent, ANY-method (plotSeqContent), 71
plotSeqContent, FastpData-method (plotSeqContent), 71
plotSeqContent, FastpDataList-method (plotSeqContent), 71
plotSeqContent, FastqcData-method (plotSeqContent), 71
plotSeqContent, FastqcDataList-method (plotSeqContent), 71
plotSeqLengthDistn, 75
plotSeqLengthDistn, ANY-method (plotSeqLengthDistn), 75
plotSeqLengthDistn, character-method (plotSeqLengthDistn), 75
plotSeqLengthDistn, FastqcData-method (plotSeqLengthDistn), 75
plotSeqLengthDistn, FastqcDataList-method (plotSeqLengthDistn), 75
plotSeqQuals, 77
plotSeqQuals, ANY-method (plotSeqQuals), 77
plotSeqQuals, character-method (plotSeqQuals), 77
plotSeqQuals, FastqcData-method (plotSeqQuals), 77
plotSeqQuals, FastqcDataList-method (plotSeqQuals), 77
plotSummary, 80
plotSummary, ANY-method (plotSummary), 80
plotSummary, character-method (plotSummary), 80
plotSummary, FastqcDataList-method (plotSummary), 80
pwf, 82
PwfCols (PwfCols-class), 83
PwfCols(). 42, 48, 51, 58, 62, 65, 67, 73, 76, 79, 81
PwfCols-class, 83
readTotals, 83
readTotals(). , 70
scale_colour_manual, 73
setAlpha (getColours), 30
setAlpha, PwfCols-method (getColours), 30
setColours (getColours), 30
setColours, PwfCols-method (getColours), 30
summariseOverrep, 84
summariseOverrep, FastpData-method (summariseOverrep), 84
summariseOverrep, FastpDataList-method (summariseOverrep), 84
summariseOverrep, FastqcData-method (summariseOverrep), 84
summariseOverrep, FastqcDataList-method (summariseOverrep), 84
theme, 43, 48, 52, 58, 59, 62, 65, 70, 73
theme_bw, 73
TheoreticalGC-class, 86
writeHtmlReport, 87