Package 'TarSeqQC'

April 23, 2016

Type Package

Version 1.0.3

Date 2015-11-25

Title TARgeted SEQuencing Experiment Quality Control

Author Gabriela A. Merino, Cristobal Fresno and Elmer A. Fernandez

Maintainer Gabriela Merino <gmerino@bdmg.com.ar>

Description The package allows the representation of targeted experiment in R. This is based on current packages and incorporates functions to do a quality control over this kind of experiments and a fast exploration of the sequenced regions. An xlsx file is generated as output.

URL http://www.bdmg.com.ar

License GPL (>=2)

Depends R (>= 3.2.1), methods, GenomicRanges, Rsamtools (>= 1.20.4), ggplot2, plyr, openxlsx

Imports S4Vectors, IRanges, BiocGenerics, reshape2, GenomeInfoDb, BiocParallel, cowplot

Suggests RUnit

Collate 'TarSeqQC-package.R' 'TargetExperiment.R' 'TargetExperiment-ampliPanel.R' 'TargetExperiment-getters.R' 'TargetExperiment-setters.R' 'TargetExperiment-show.R' 'TargetExperiment-print.R' 'TargetExperiment-pileupCounts.R' 'TargetExperiment-buildFeaturePanel.R' 'TargetExperiment-summarizePanel.R' 'TargetExperiment-initialize.R' 'TargetExperiment-constructor.R' 'TargetExperiment-statistics.R' 'TargetExperiment-plot.R' 'TargetExperiment-ggplotColours.R' 'TargetExperiment-addStatSummSheet.R' 'TargetExperiment-buildReport.R' 'TargetExperiment-plotAttrExpl.R' 'TargetExperiment-plotFeatPerform.R'

'TargetExperiment-plotRegion.R'
'TargetExperiment-plotFeature.R'
'TargetExperiment-plotGeneAttrPerFeat.R'
'TargetExperiment-plotNtdPercentage.R'

biocViews Software, Sequencing, TargetedResequencing, QualityControl, Visualization, Coverage, Alignment, DataImport

NeedsCompilation no

R topics documented:

`TarSeqQC-package` *TarSeqQC: Targeted Sequencing Experiment Quality Control R package*

Description

The package models targeted sequencing experiment output using previous packages. This package includes the new following features:

1. Panel model:
   - Model customizable feature panels.

- Evaluation of the sequencing run performance at median or coverage level for each feature.
- Exploration of sequenced features.

2. Quality Control of the sequencing run:
   - General overview of the run performance.
   - Statistical indicators at median or coverage level.
   - Xlsx report.
3. Customizable scan bam file parameters.
4. Customizable pileup build parameters.
5. Incorporation of fasta sequence.
6. Fast exploration of read profile for particular features or genomic regions, coloring SNPs occurences.

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

`addStatSummSheet` *Build excel report of the Target Experiment.*

Description

`addStatSummSheet` adds the statistics summary sheet to the workbook that contains the Target Experiment Report.

`buildReport` builds an excel file containing some statistical results. These are computed to the selected attribute (e.g. 'coverage') along features (e.g. 'amplicon') and genes. If 'imageFile' is null, the graph generated calling the generic plot function will be used.

`ggplotColours` is a function to know what color is used when ggplot is called.

Usage

```
addStatSummSheet(object, wb, attributeThres = c(0, 1, 50, 200, 500, Inf),
    imageFile)

## S4 method for signature 'TargetExperiment'
addStatSummSheet(object, wb,
    attributeThres = c(0, 1, 50, 200, 500, Inf), imageFile)

buildReport(object, attributeThres = c(0, 1, 50, 200, 500, Inf),
    imageFile = NULL, file)

## S4 method for signature 'TargetExperiment'
buildReport(object, attributeThres = c(0, 1, 50,
```

```
    200, 500, Inf), imageFile = NULL, file = "Results.xlsx")

ggplotColours(object, n)

## S4 method for signature 'TargetExperiment'
ggplotColours(object, n)
```

Arguments

object	TargetExperiment class object.
wb	A workbook object that will contain the report.
attributeThres	Numeric indicating the intervals extreme values.
imageFile	Character indicating the name of the file that contains the plot that could be insert in the report.
file	Character indicating the name of the report.
n	amount of colors.

Value

Workbook object.

NULL.

colours

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

TargetExperiment-class

Examples

```
## Loading the TargetExperiment object
data(ampliPanel,package="TarSeqQC")
# definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)

## Building the XLSX report
imageFile<-system.file("extdata", "plot.pdf", package="TarSeqQC",
mustWork=TRUE)
buildReport(ampliPanel, attributeThres=attributeThres, imageFile=imageFile,
file="results.xlsx")
```

ampliPanel *An amplicon panel example for use the TarSeqQC R package.*

Description

A non-real dataset containing amplicon sequencing results to test the TarSeqQC package.

Format

A TargetExperiment object

Details

bedFile Bed file containing 29 amplicons and 8 genes.

feature Character "amplicon" indicating that the analyzed features are amplicon sequences

attribute Character "coverage"

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Source

see TargetExperiment-class

See Also

Other TargetExperiment: TargetExperiment-class; TargetExperiment, TargetExperiment-methods; initialize, initialize,TargetExperiment-method

buildFeaturePanel *Function to build a feature panel based on specific genomic regions.*

Description

buildFeaturePanel builds panel slots of a TargetExperiment object. Input can be a bam file or a pileup matrix. If the bed file contains a high number of amplicons, the bam file as input is recommended in order to diminish memory requirements. The resulting object is a GRanges instance having panel and counts/coverage information.

Usage

```
buildFeaturePanel(object, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
buildFeaturePanel(object, BPPARAM = bpparam())
```

Arguments

object	TargetExperiment class object.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

GRanges object.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Examples

```
if (interactive()) {
  ## loading TargetExperiment object
  data(ampliPanel, package="TarSeqQC")
  ## Defining bam file, bed file and fasta file names and paths
  setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
      package="TarSeqQC", mustWork=TRUE)
  setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
      package="TarSeqQC", mustWork=TRUE)

  myFeaturePanel<-buildFeaturePanel(ampliPanel)
}
```

---

getBedFile *Getters for TargetExperiment object.*

Description

Obtain TargetExperiment's slot information, according to the given function call.

Usage

```
getBedFile(object)

## S4 method for signature 'TargetExperiment'
getBedFile(object)

getBamFile(object)
```

```
## S4 method for signature 'TargetExperiment'
getBamFile(object)

getFastaFile(object)

## S4 method for signature 'TargetExperiment'
getFastaFile(object)

getFeaturePanel(object)

## S4 method for signature 'TargetExperiment'
getFeaturePanel(object)

getGenePanel(object)

## S4 method for signature 'TargetExperiment'
getGenePanel(object)

getFeature(object)

## S4 method for signature 'TargetExperiment'
getFeature(object)

getAttribute(object)

## S4 method for signature 'TargetExperiment'
getAttribute(object)

getScanBamP(object)

## S4 method for signature 'TargetExperiment'
getScanBamP(object)

getPileupP(object)

## S4 method for signature 'TargetExperiment'
getPileupP(object)

getRegion(object, level, ID, collapse = TRUE)

## S4 method for signature 'TargetExperiment'
getRegion(object, level, ID, collapse = TRUE)

getLowCtsFeatures(object, level, threshold = 50)

## S4 method for signature 'TargetExperiment'
getLowCtsFeatures(object, level, threshold = 50)
```

Arguments

object	TargetExperiment class object.
level	Character indicating 'gene' or 'feature'. Useful to getRegion function
ID	Character indicating the feature name that getRegion should be found.
collapse	Logical. Should the region be collapsed?.
threshold	Numeric what should be the minimum attribute value?.

Value

according to the call one of the following objects can be returned

GRanges	bed file of the experiment
BamFile	reference to the BAM file
FaFile	reference to the fasta file
GRanges	feature panel with statistical information
GRanges	summarized version of the feature panel at gene level
character	name of the explored features (e.g 'amplicon', 'exon')
character	name of the analyzed attribute ('coverage' or 'medianCounts')
ScanBamParam	parameters for the scan of the BAM file
PileupParam	parameters for the pileup building
data.frame	regions or low counts features

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar> Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

TargetExperiment-class

Examples

```
## Loading the TargetExperiment object
data(ampliPanel,package="TarSeqQC")
## Get the bedFile slot
getBedFile(ampliPanel)
## Get the bamFile slot
getBamFile(ampliPanel)
## Get the fastaFile slot
getFastaFile(ampliPanel)
## Get the feateurePanel slot
```

```
getFeaturePanel(ampliPanel)
## Get the genePanel slot
getGenePanel(ampliPanel)
## Get the Feature slot
getFeature(ampliPanel)
## Get the attribute slot
getAttribute(ampliPanel)
## Get the scanBamP slot
getScanBamP(ampliPanel)
## Get the pileupP slot
getPileupP(ampliPanel)
## Get the region related to a feature or a gene
getRegion(ampliPanel, level="gene", ID="gene7", collapse=FALSE)
## Get the low counts features
getLowCtsFeatures(ampliPanel, level="feature")
```

`initialize` *TargetExperiment object constructor.*

Description

`initialize` creates the TargetExperiment object architecture for the specified bed and alingment BAM files. If 'scanBamP' and/or 'pileupP' parameters are not specified, default values of their constructors will be used.

Usage

```
## S4 method for signature 'TargetExperiment'
initialize(.Object, bedFile, bamFile, fastaFile,
    scanBamP = NULL, pileupP = NULL, feature = NULL, attribute = NULL,
    BPPARAM = bpparam())
```

Arguments

`.Object`	TargetExperiment class.
`bedFile`	Character indicating the bed file full path.
`bamFile`	Character indicating the alignment and index bam files full paths.
`fastaFile`	Character indicating the full path to the genome reference and index files.
`scanBamP`	ScanBamParam indicating the parameters for read the BAM file.
`pileupP`	PileupParam indicating the parameters for pileup building.
`feature`	Character indicating the name of the feature that will be explored (e.g 'amplicon', 'exon').
`attribute`	Character indicating the name of the attribute that will be explored. Should be 'coverage' or 'medianCounts'.
`BPPARAM`	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

TargetExperiment object.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

TargetExperiment, buildFeaturePanel summarizePanel

Other TargetExperiment: TargetExperiment-class; TargetExperiment, TargetExperiment-methods; ampliPanel

Examples

```
## Defining bam file, bed file and fasta file names and paths
if (interactive()){
    bamFile<-system.file("extdata", "mybam.bam", package="TarSeqQC",
        mustWork=TRUE)
    bedFile<-system.file("extdata", "mybed.bed", package="TarSeqQC",
        mustWork=TRUE)
    fastaFile<-system.file("extdata", "myfasta.fa", package="TarSeqQC",
        mustWork=TRUE)

    ## Creating a TargetExperiment object

    ## Defining feature parameter
    feature<-"amplicon"
    ## Defining attribute parameter
    attribute<-"coverage"
    ##Calling the constructor
    ampliPanel<-TargetExperiment(bedFile, bamFile, fastaFile,
        attribute=attribute, feature=feature)
}
```

---

pileupCounts *Function to obtain the pileup counts for a bam file.*

Description

pileupCounts waits a TargetExperiment object containing the bed file information in order to obtain pileup counts only for the specified genomic regions. The resulting object is a data.frame instance, in which each row represents one position of the specified features across the bed file. The first three columns called 'pos', 'seqnames' and 'which_label,' represent the position in the seqnames (e.g. pos=10183795 and seqnames=chr3) and the associated feature. According to the 'pileupP' parameters setted before, the number of next columns could change. If 'distinguish_nucleotide' was set as TRUE, then one column per ntd will appear containing the counts obtained for each of them. Same will occur when 'distinguish_strands' is set as TRUE. The last column, called 'counts', contains the total counts obtained for the corresponding position.

Usage

```
pileupCounts(bed, bamFile, fastaFile, scanBamP = NULL, pileupP = NULL,
    BPPARAM = bpparam())
```

Arguments

bed	a Granges object containing the bed file information.
bamFile	Character indicating the alignment and index bam files full path.
fastaFile	Character indicating the full path to the genome reference and index files.
scanBamP	ScanBamParam indicating the parameters the BAM file read.
pileupP	PileupParam indicating the parameters for the pileup build.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

data.frame object.

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

References

1. Morgan M, Pages H, Obenchain V and Hayden N. Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import. R package version 1.20.1

See Also

Rsamtools-pileup

Examples

```
##Defining the bed GRanges
data(ampliPanel, package="TarSeqQC")
bed<-getBedFile(ampliPanel)
## Defining bam file and fasta file names and paths
bamFile<-system.file("extdata", "mybam.bam",
    package="TarSeqQC", mustWork=TRUE)
fastaFile<-system.file("extdata", "myfasta.fa",
    package="TarSeqQC", mustWork=TRUE)
## extracting the pileup matrix
myCounts<-pileupCounts(bed, bamFile, fastaFile)
head(myCounts)
```

`plot` *Plot TargetExperiment object overview.*

Description

`plot` allows a fast and simple representation of one feature panel using a polar histogram plot. Histogram bar reflects the percentage of features that have shown the analyzed attribute in a user setted interval. The resulting graph can be busy and might be better off saved.

Usage

```
## S4 method for signature 'TargetExperiment,ANY'
plot(x, y, attributeThres = c(0, 1, 50, 200,
    500, Inf), binSize = 1, spaceGene = 0.2, spaceChr = 1.2,
    innerRadius = 0.3, outerRadius = 1, guides = c(20, 40, 60, 80),
    alphaStart = -0.3, circleProportion = 0.95, direction = "inwards",
    chrLabels = FALSE)
```

Arguments

`x`	TargetExperiment class object.
`y`	not used but necessary for redefining the generic function.
`attributeThres`	Numeric indicating the interval extreme values.
`binSize`	Numeric indicating bin width. Should probably be left as 1, as other parameters are relative to it.
`spaceGene`	Numeric. Space between bins.
`spaceChr`	Numeric. Space between chromosomes.
`innerRadius`	Numeric. Radius of inner circle.
`outerRadius`	Numeric. Radius of outer circle.
`guides`	A vector with percentages to use for the white guide lines.
`alphaStart`	Numeric offset from 12 o'clock in radians.

circleProportion
: Numeric proportion of the circle to cover.

direction
: Character indicating if the increasing count goes from or to the centre.

chrLabels
: Logical. Chromosome names must be plotted?.

Value

a ggplot2 graph.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar> Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

References

http://www.r-bloggers.com/polar-histogram-pretty-and-useful/

See Also

plotFeatPerform

Examples

```
if(interactive()){
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")
# Definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)

## Plot panel overview
g<-plot(ampliPanel, attributeThres, chrLabels =TRUE)
g
}
```

---

plotAttrExpl *Plot attribute exploration of a TargetExperiment object.*

Description

plotAttrExpl plots density and box-plot of the analyzed attribute at a feature or gene level. This graphics could plot together using the ggplot2 geom_violin method.

Usage

```
plotAttrExpl(object, level = "feature", join = TRUE, log = TRUE,
    color = "blue")

## S4 method for signature 'TargetExperiment'
plotAttrExpl(object, level = "feature",
    join = TRUE, log = TRUE, color = "blue")
```

Arguments

object	TargetExperiment class object.
level	Character 'feature' or 'gene' indicating at which level should be analyzed the attribute.
join	Logical indicating if boxplot and density function should be plotted together using the ggplot2 geom_violin method.
log	Logical indicating if the attribute should be considered in log10 scale.
color	A character indicating a valid name color.

Value

ggplot2 graphics.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plot, plotFeatPerform

Examples

```
if(interactive()){
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")

# Attribute boxplot and density plot exploration
g<-plotAttrExpl(ampliPanel,level="feature",join=TRUE, log=FALSE, color="blue")
# x11(type="cairo")
g
}
```

plotFeatPerform *Plot feature performance of a TargetExperiment object.*

Description

plotFeatPerform plots the achieved performance for each feature/gene. The resulting graph shows one bar per each feature/gene with height according to its attribute value. If complete is set as TRUE, two bar plots (feature and gene level) will be stored in the resulting ggplot object.

Usage

```
plotFeatPerform(object, attributeThres = c(0, 1, 50, 200, 500, Inf),
    complete = TRUE, log = TRUE, featureLabs = FALSE, sepChr = FALSE,
    legend = TRUE)

## S4 method for signature 'TargetExperiment'
plotFeatPerform(object, attributeThres = c(0, 1,
    50, 200, 500, Inf), complete = TRUE, log = TRUE, featureLabs = FALSE,
    sepChr = FALSE, legend = TRUE)
```

Arguments

object	TargetExperiment class object.
attributeThres	Numeric indicating the intervals extreme values.
complete	Logical indicating if the gene and feature level exploration should be plotted.
log	Logical indicating if the attribute should be considered in log10 scale.
featureLabs	Logical indicating if feature labels should be plotted.
sepChr	Logical indicating if the plot should show chromosome divisions.
legend	Logical indicating if legend should be plotted.

Value

ggplot2 graphics

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plot

Examples

```
if(interactive()){
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")

# Definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)

# Plot panel overview in a feature performance plot
g<-plotFeatPerform(ampliPanel, attributeThres=attributeThres, log=FALSE,
featureLabs=TRUE, sepChr=TRUE, legend=TRUE)
g
}
```

`plotFeature` *Plot read profiles for a particular feature.*

Description

`plotFeature` plots the achieved performance for each feature/gene. The resulting graph shows one bar per each feature/gene with heights according to its attribute value. If complete is set as TRUE two bar plots (feature and gene level) will be stored in the resulting ggplot object.

Usage

```
plotFeature(object, featureID, SNPs = TRUE, xlab = "", title = "",
    size = 0.5, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
plotFeature(object, featureID, SNPs = TRUE,
    xlab = "", title = featureID, size = 0.5, BPPARAM = bpparam())
```

Arguments

`object`	TargetExperiment object.
`featureID`	Character indicating the ID of the feature.
`SNPs`	Logical flag indicating if SNPs should be plotted.
`xlab`	Character containing the axis x label.
`title`	Character containing the plot title.
`size`	Numeric indicating the size of line plots.
`BPPARAM`	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

ggplot2 graphics.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plotRegion

Examples

```
if(interactive()){
## loading TargetExperiment object
data(ampliPanel, package="TarSeqQC")
## Defining bam file, bed file and fasta file names and paths
setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
    package="TarSeqQC", mustWork=TRUE)
setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
    package="TarSeqQC", mustWork=TRUE)

# Exploring the read count profile for a particular amplicon
g<-plotFeature(ampliPanel, featureID="AMPL20")
g
}
```

---

plotGeneAttrPerFeat *Plot the attribute value for all the features of a selected gene.*

Description

plotGeneAttrPerFeat plots the achieved performance for each feature for a particular gene. The resulting graph shows one bar per each gene feature with heights according to its attribute value.

Usage

```
plotGeneAttrPerFeat(object, geneID)

## S4 method for signature 'TargetExperiment'
plotGeneAttrPerFeat(object, geneID)
```

Arguments

object	TargetExperiment object.
geneID	Character indicating the ID of the selected gene.

Value

ggplot2 graphics.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plotAttrExpl

Examples

```
if(interactive()){
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")

# Exploring amplicon attribute values for a particular gene
g<-plotGeneAttrPerFeat(ampliPanel, geneID="gene4")
# Adjust text size
g<-g+theme(title=element_text(size=16), axis.title=element_text(size=16),
legend.text=element_text(size=14))
g
}
```

---

plotNtdPercentage *Plot nucleotide read percentages for a particular feature.*

Description

plotNtdPercentage plots the percentages of the ocurrence of each nucleotide in each position for a selected feature.

Usage

```
plotNtdPercentage(object, featureID, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
plotNtdPercentage(object, featureID,
    BPPARAM = bpparam())
```

Arguments

object	a TargetExperiment object.
featureID	a character indicating the feature ID.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation. returned by the function.

Value

ggplot2 graphics

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plotFeature

Examples

```
if(interactive()){
## loading TargetExperiment object
data(ampliPanel, package="TarSeqQC")
## Defining bam file, bed file and fasta file names and paths
setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
    package="TarSeqQC", mustWork=TRUE)
setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
    package="TarSeqQC", mustWork=TRUE)
# Exploring the nucleotide percentages compositions of the read counts for a
# particular amplicon
g<-plotNtdPercentage(ampliPanel,featureID="AMPL20")
g
}
```

---

plotRegion *Plot read profiles for a particular genomic region.*

Description

plotRegion plots the read profiles for a selected region. If SNPs is set as 'TRUE', colored bars will appear indicating the ocurrence of SNPs at each genomic position.

Usage

```
plotRegion(object, region, seqname, SNPs = TRUE, xlab = "", title = "",
    size = 0.5, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
plotRegion(object, region, seqname, SNPs = TRUE,
    xlab = "", title = "", size = 0.5, BPPARAM = bpparam())
```

Arguments

object	TargetExperiment object.
region	Numeric of length two indicating the selected genomic region.
seqname	Character indicating the chromosome of the genomic region.
SNPs	Logical flag indicating if SNPs should be plotted.
xlab	Character containing the axis x label.
title	Character containing the plot title.
size	Numeric indicating the size of line plots.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

ggplot2 graphics.

include TargetExperiment-FeatPerform.R

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

plotFeature

Examples

```
if(interactive()){
## loading TargetExperiment object
data(ampliPanel, package="TarSeqQC")
## Defining bam file, bed file and fasta file names and paths
setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
    package="TarSeqQC", mustWork=TRUE)
setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
```

```
    package="TarSeqQC", mustWork=TRUE)

# getting and exploring a sequenced region of a particular gene
getRegion(ampliPanel, level="gene", ID="gene7", collapse=FALSE)
# plot a particular genomic region
g<-plotRegion(ampliPanel,region=c(4500,6800), seqname="chr10", SNPs=TRUE,
xlab="", title="gene7 amplicons",size=0.5)
# x11(type="cairo")
g
}
```

---

print *Print a TargetExperiment object.*

Description

Generic `print` method for TargetExperiment class and descendants.

Usage

```
## S4 method for signature 'TargetExperiment'
print(x, ...)
```

Arguments

`x`	TargetExperiment class object.
`...`	Included for generic print compatibility.

Value

console output of the object.

Note

see full example in `TargetExperiment-class`

Author(s)

Gabriela Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Examples

```
## Loading the TargetExperiment object
data(ampliPanel,package="TarSeqQC")
print(ampliPanel)
```

setFeature<- *Setters for the TargetExperiment slots*

Description

Set TargetExperiment slots, according to the given function call.

Usage

```
setFeature(object) <- value

## S4 replacement method for signature 'TargetExperiment,character'
setFeature(object) <- value

setAttribute(object) <- value

## S4 replacement method for signature 'TargetExperiment,character'
setAttribute(object) <- value

setScanBamP(object) <- value

## S4 replacement method for signature 'TargetExperiment,ScanBamParam'
setScanBamP(object) <- value

setPileupP(object) <- value

## S4 replacement method for signature 'TargetExperiment,PileupParam'
setPileupP(object) <- value

setFeaturePanel(object) <- value

## S4 replacement method for signature 'TargetExperiment,GRanges'
setFeaturePanel(object) <- value

setGenePanel(object) <- value

## S4 replacement method for signature 'TargetExperiment,GRanges'
setGenePanel(object) <- value

setBedFile(object) <- value

## S4 replacement method for signature 'TargetExperiment,character'
setBedFile(object) <- value

setBamFile(object) <- value

## S4 replacement method for signature 'TargetExperiment,character'
```

```
setBamFile(object) <- value

setFastaFile(object) <- value

## S4 replacement method for signature 'TargetExperiment,character'
setFastaFile(object) <- value
```

Arguments

object	TargetExperiment class object.
value	value to set the slot.

Value

a TargetExperiment object

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar> Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Examples

```
## loading TargetExperiment object
if (interactive()){
    data(ampliPanel, package="TarSeqQC")
    ## Defining bam file, bed file and fasta file names and paths
    setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
        package="TarSeqQC", mustWork=TRUE)
    setBedFile(ampliPanel)<-system.file("extdata", "mybed.bed",
        package="TarSeqQC", mustWork=TRUE)
    setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
        package="TarSeqQC", mustWork=TRUE)
    ## Set feature slot value
    setFeature(ampliPanel)<-"amplicon"
    ## Set attribute slot value
    setAttribute(ampliPanel)<-"coverage"
    ## Set scanBamP slot value
    setScanBamP(ampliPanel)<-ScanBamParam()
    ## Set pileupP slot value
    setPileupP(ampliPanel)<-PileupParam()
    ## Set featurePanel slot value
    setFeaturePanel(ampliPanel)<-buildFeaturePanel(ampliPanel)
    ## Set genePanel slot value
    setGenePanel(ampliPanel)<-summarizePanel(ampliPanel)
    ## Set bedFile slot value
    setBedFile(ampliPanel)<-system.file("extdata", "mybed.bed",
```

```
    package="TarSeqQC", mustWork=TRUE)
  ## Set bamFile slot value
  setBamFile(ampliPanel)<-system.file("extdata", "mybam.bam",
    package="TarSeqQC", mustWork=TRUE)
  ## Set fastaFile slot value
  setFastaFile(ampliPanel)<-system.file("extdata", "myfasta.fa",
    package="TarSeqQC", mustWork=TRUE)
}
```

---

show *Show method for the TargetExperiment class.*

Description

show a TargetExperiment object

Usage

```
## S4 method for signature 'TargetExperiment'
show(object)
```

Arguments

object	TargetExperiment class object

Details

Generic show method for TargetExperiment class output visualization.

Value

console output of the object

Note

see full example in TargetExperiment-class

Author(s)

Gabriela Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Examples

```
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")
show(ampliPanel)
```

summarizePanel *Function to summarize a featurePanel slot at a gene level.*

Description

summarizePanel helps the initialization of a TargetExperiment object. Is useful to summarize the featurePanel slot at a gene level, building the genePanel slot.

Usage

```
summarizePanel(object, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
summarizePanel(object, BPPARAM = bpparam())
```

Arguments

object	TargetExperiment class object.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

TargetExperiment object

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

TargetExperiment,buildFeaturePanel

Examples

```
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")

mySummarizedPanel<-summarizePanel(ampliPanel)
```

summaryFeatureLev *TargetExperiment summary.*

Description

Explore the TargetExperiment's attribute values at feature and/or gene level.

Usage

```
summaryFeatureLev(object)

## S4 method for signature 'TargetExperiment'
summaryFeatureLev(object)

summaryGeneLev(object)

## S4 method for signature 'TargetExperiment'
summaryGeneLev(object)

## S4 method for signature 'TargetExperiment'
summary(object, ...)

summaryIntervals(object, attributeThres = c(0, 1, 50, 200, 500, Inf))

## S4 method for signature 'TargetExperiment'
summaryIntervals(object, attributeThres = c(0, 1,
    50, 200, 500, Inf))
```

Arguments

object	TargetExperiment class object.
...	required by summary.
attributeThres	numeric indicating the intervals extreme values required by summaryIntervals.

Value

according to the call one of the following objects can be returned

data.frame	statistics of the analyzed attribute
data.frame	Frequency table of the feature occurrence in the selected intervals

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Examples

```
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")

# Summary at feature level
summaryFeatureLev(ampliPanel)
# Summary at gene level
summaryGeneLev(ampliPanel)
# Defining the attribute interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)
# Doing a frequency table for the attribute intervals
summaryIntervals(ampliPanel, attributeThres=attributeThres)
```

TargetExperiment *TargetExperiment constructor*

Description

TargetExperiment creates a TargetExperiment object with the architecture specified by the bed and alingment BAM files. If 'scanBamP' and/or 'pileupP' parameters are not specified, default values of their constructors will be used. attribute and feature parameters can be setted after constructor calling.

Usage

```
TargetExperiment(bedFile, bamFile, fastaFile, scanBamP = NULL,
    pileupP = NULL, feature = NULL, attribute = NULL, BPPARAM = bpparam())
```

Arguments

bedFile	Character indicating the bed file full path.
bamFile	Character indicating the alignment and index bam files full path.
fastaFile	Character indicating the full path to the genome reference and index files.
scanBamP	ScanBamParam indicating the parameters the BAM file read.
pileupP	PileupParam indicating the parameters for the pileup build.
feature	Character indicating the name of the feature that will be explored (e.g 'amplicon', 'exon').
attribute	Character indicating the name of the attribute that will be explored. Should be 'coverage' or 'medianCounts'.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Value

TargetExperiment object.

Note

see full example in TargetExperiment-class

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

TargetExperiment-class1

Other TargetExperiment: TargetExperiment-class; ampliPanel; initialize, initialize,TargetExperiment-method

Examples

```
if (interactive()) {
## Defining bam file, bed file and fasta file names and paths
bamFile<-system.file("extdata", "mybam.bam", package="TarSeqQC",
mustWork=TRUE)
bedFile<-system.file("extdata", "mybed.bed", package="TarSeqQC",
mustWork=TRUE)
fastaFile<-system.file("extdata", "myfasta.fa", package="TarSeqQC",
mustWork=TRUE)

## Creating a TargetExperiment object

# Defining feature parameter
feature<-"amplicon"
# Defining attribute parameter
attribute<-"coverage"
##Calling the constructor
object<-TargetExperiment(bedFile, bamFile, fastaFile, attribute=attribute,
feature=feature)
}
```

TargetExperiment-class

TargetExperiment S4 class implementation in R

Description

This S4 class represents a Targeted Sequencing Experiment in R. Targeted Sequencing Experiments are characterized by a 'bed file' that contains the specification of the explored 'features' as a 'panel'. This features could be amplicons, exons, transcripts, among others. In general each feature is associated to one gene. A gene could be related to many features. This class allows the representation and quality control of a Targeted Sequencing Experiment.

Slots

`scanBamP` ScanBamParam containing the information to scan the BAM file.

`pileupP` PileupParam containing the information to build the pileup.

`bedFile` GRanges object that models the bed file.

`bamFile` BamFile object that is a reference to the BAM file.

`fastaFile` FaFile object that is a reference to the reference sequence.

`featurePanel` GRanges object that models the feature panel and related statistics.

`genePanel` GRanges object that models the analyzed panel and related statistics at a gene level.

`attribute` character indicates which attribute 'coverage' or 'medianCounts' will be used to the analysis.

`feature` character indicates the name of the analyzed features. E.g 'amplicon', 'exon', 'transcript'.

Features

1. Model Targeted Sequencing Experiments in R.
2. Obtain coverage and read counts per sequenced feature.
3. Evaluate the performance of a targeted sequencing experiment using coverage/read counts information.
4. Detect in early stage sequencing or library preparation errors.
5. Explore read profiles for particular features or genomic regions.
6. Explore any kind of experiment in which 'feature' definition is possible for several genes. E.g RNA-seq experiments in which transcripts could be the 'features'.
7. Report quality control results.

Functions

TargetExperiment S4 class includes the following functions:

pileupCounts calculate pileup statistics for the BAM file

buildFeaturePanel build and model a feature panel as a GRanges object and compute read statistics

summarizePanel summarize the feature panel to a gene panel and compute read statistics

initialize constructor of TargetExperiment to generate the feature and gene panels starting from an alignment BAM file and the bed file

getBedFile, getBamFile, getFeaturePanel, getGenePanel, getAttribute, getFeature, getScanBamP, getPileupP return the respective TargetExperiment slot

setAttribute,setFeature, setScanBamP, setPileupP return the respective TargetExperiment slots

show generic output of the object

print generic output of the object

summary print statistics summary for the setted attribute

freqTable build a frequency table of the attribute occurrence in user configured intervals

plot plot a summarized view of the feature panel performance

plotAttrExpl plot the density and distribution of the attribute

plotFeatPerform plot the sequencing performance for each feature and/or gene

plotFeature plot the reads profile for a particular feature

plotGeneAttrPerFeat plot the explored attribute for each feature of a particular gene

plotNtdPercentages plot nucleotide percentages for each position of a particular feature

plotRegion plot the reads profile for a particular genomic region

addStatSummSheet internal function to add the first sheet of xlsx reports

buildReport build the experiment report as an xlsx file

Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar> examples:

Defining bam file, bed file and fasta file names and paths bamFile<-system.file("extdata", "mybam.bam", package="TarSeqQC", mustWork=TRUE) bedFile<-system.file("extdata", "mybed.bed", package="TarSeqQC", mustWork=TRUE) fastaFile<-system.file("extdata", "myfasta.fa", package="TarSeqQC", mustWork=TRUE)

Creating a TargetExperiment object

Defining feature parameter feature<-"amplicon" # Defining attribute parameter attribute<-"coverage" ampliPanel<-TargetExperiment(bedFile, bamFile, fastaFile, attribute=attribute, feature=feature)

Alternative object creation # Creating the TargetExperiment object ampliPanel<-TargetExperiment(bedFile, bamFile, fastaFile) # Set feature slot value setFeature(ampliPanel)<-"amplicon" # Set attribute slot value setAttribute(ampliPanel)<-"coverage" # Set pileupP slot value in order to set the maximum depth at 1000 setPileupP(ampliPanel)<-PileupParam(max_depth=1000) # Set the featurePanel slot but now using the new pileupP definition setFeaturePanel(ampliPanel)<-buildFeaturePanel(ampliPanel) ## Early exploration # show/print ampliPanel # summary summary(ampliPanel) # summary at feature level summaryFeatureLev(ampliPanel) # summary at gene level summaryGeneLev(ampliPanel) # attribute boxplot and density plot exploration g<-plotAttrExpl(ampliPanel,level="feature",join=TRUE, log=FALSE, color="blue") # x11(type="cairo") g ## Deep exploration and Quality Control # definition of the interval extreme values attributeThres<-c(0,1,50,200,500, Inf) # plot panel overview g<-plot(ampliPanel, attributeThres, chrLabels =TRUE) g # plot panel overview in a feature performance plot g<-plotFeatPerform(ampliPanel, attributeThres, complete=TRUE, log=FALSE, featureLabs=TRUE, sepChr=TRUE, legend=TRUE) g

Controlling low counts features # Do a frequency table for the attribute intervals summaryIntervals(ampliPanel, attributeThres) # getting low counts features at gene level getLowCtsFeatures(ampliPanel, level="gene", threshold=50) # getting low counts features at feature level getLowCtsFeatures(ampliPanel, level="feature", threshold=50) # exploring amplicon attribute values for a particular gene g<-plotGeneAttrPerFeat(ampliPanel, geneID="gene4") # adjust text size g<-g+theme(title=element_text(size=16), axis.title=element_text(size=16), legend.text=element_text(size=14)) g

extracting the pileup matrix myCounts<-pileupCounts(ampliPanel) head(myCounts)

getting and exploring a sequenced region of a particular gene getRegion(ampliPanel, level="gene", ID="gene7", collapse=FALSE) # plot a particular genomic region g<-plotRegion(ampliPanel,region=c(4500,6800),

seqname="chr10", SNPs=TRUE, xlab="", title="gene7 amplicons",size=0.5) # x11(type="cairo") g # exploring the read count profile for a particular amplicon g<-plotFeature(ampliPanel, featureID="AMPL20") # x11(type="cairo") g # exploring the nucleotide percentages compositions of the read counts for a # particular amplicon g<-plotNtdPercentage(ampliPanel,featureID="AMPL20") g ## Building the XLSX report imageFile<-system.file("extdata", "plot.pdf", package="TarSeqQC", mustWork=TRUE) buildReport(ampliPanel, attributeThres, imageFile ,file="Results.xlsx")

See Also

Rsamtools

Other TargetExperiment: `TargetExperiment`, `TargetExperiment-methods`; `ampliPanel`; `initialize`, `initialize,TargetExperiment-method`

Index