

# Package ‘TarSeqQC’

April 15, 2020

**Type** Package

**Version** 1.16.0

**Date** 2019-03-13

**Title** TARgeted SEQuencing Experiment Quality Control

**Author** Gabriela A. Merino, Cristobal Fresno, Yanina Murua, Andrea S. Llera  
and Elmer A. Fernandez

**Maintainer** Gabriela Merino <merino.gabriela33@gmail.com>

**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 (>= 3.5.1), methods, GenomicRanges, Rsamtools (>= 1.9.2),  
ggplot2, plyr, openxlsx

**Imports** grDevices, stats, utils, S4Vectors, IRanges, BiocGenerics,  
reshape2, GenomeInfoDb, BiocParallel, Biostrings, cowplot,  
graphics, GenomicAlignments, Hmisc

**Suggests** BiocManager, RUnit

**Collate** 'TarSeqQC-package.R' 'TargetExperiment.R'  
'TargetExperimentList.R' 'checkBedFasta.R'  
'TargetExperiment-ampliPanel.R'  
'TargetExperiment-ampliPanel2.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-plotRegion.R'  
'TargetExperiment-plotFeature.R'  
'TargetExperiment-plotGeneAttrPerFeat.R'

'TargetExperiment-plotNtdPercentage.R'  
 'TargetExperiment-readFrequencies.R'  
 'TargetExperiment-myCounts.R'  
 'TargetExperiment-plotInOutFeatures.R'  
 'TargetExperiment-biasExploration.R'  
 'TargetExperiment-buildReport.R'  
 'TargetExperiment-plotAttrPerform.R'  
 'TargetExperiment-plotAttrExpl.R'  
 'TargetExperiment-plotFeatPerform.R'  
 'TargetExperiment-plotMetaDataExpl.R'  
 'TargetExperimentList-TEList.R'  
 'TargetExperimentList-initialize.R'  
 'TargetExperimentList-constructor.R'  
 'TargetExperimentList-getters.R'  
 'TargetExperimentList-setters.R' 'TargetExperimentList-show.R'  
 'TargetExperimentList-print.R'  
 'TargetExperimentList-statistics.R'  
 'TargetExperimentList-plot.R'  
 'TargetExperimentList-plotGlobalAttrExpl.R'  
 'TargetExperimentList-plotAttrExpl.R'  
 'TargetExperimentList-plotPoolPerformance.R'

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

**NeedsCompilation** no

**RoxygenNote** 5.0.1

**git\_url** <https://git.bioconductor.org/packages/TarSeqQC>

**git\_branch** RELEASE\_3\_10

**git\_last\_commit** e258522

**git\_last\_commit\_date** 2019-10-29

**Date/Publication** 2020-04-14

## R topics documented:

TarSeqQC-package . . . . .	3
ampliPanel . . . . .	4
ampliPanel2 . . . . .	4
biasExploration . . . . .	5
buildFeaturePanel . . . . .	6
buildReport . . . . .	7
checkBedFasta . . . . .	9
getBedFile . . . . .	10
initialize . . . . .	13
initialize,TargetExperimentList-method . . . . .	14
myCounts . . . . .	16
object . . . . .	17
pileupCounts . . . . .	17
plot . . . . .	19
plotAttrExpl . . . . .	20
plotAttrPerform . . . . .	22

plotFeatPerform . . . . .	23
plotFeature . . . . .	24
plotGeneAttrPerFeat . . . . .	26
plotGlobalAttrExpl . . . . .	27
plotInOutFeatures . . . . .	28
plotMetaDataExpl . . . . .	29
plotNtdPercentage . . . . .	31
plotPoolPerformance . . . . .	32
plotRegion . . . . .	33
print . . . . .	34
readFrequencies . . . . .	35
setFeature<- . . . . .	36
show . . . . .	38
summarizePanel . . . . .	39
summaryFeatureLev . . . . .	40
TargetExperiment . . . . .	41
TargetExperiment-class . . . . .	43
TargetExperimentList . . . . .	47
TargetExperimentList-class . . . . .	48
TEList . . . . .	50

<b>Index</b>	<b>51</b>
--------------	-----------

---

TarSeqQC-package	<i>TarSeqQC: Targeted Sequencing Experiment Quality Control R package</i>
------------------	---

---

## 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 occurrences.

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

---

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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Source**

see [TargetExperiment-class](#)

**See Also**

Other TargetExperiment: [TargetExperiment-class](#), [TargetExperiment](#), [ampliPanel2](#), [initialize](#), [myCounts](#)

---

ampliPanel2

*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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Source**

see [TargetExperiment-class](#)

**See Also**

Other TargetExperiment: [TargetExperiment-class](#), [TargetExperiment](#), [ampliPanel](#), [initialize](#), [myCounts](#)

---

biasExploration	<i>Plot attribute density and boxplot for each bias source quartile or category.</i>
-----------------	--

---

**Description**

biasExploration plots density and box-plot of the analyzed attribute for each bias source' quartiles per categories. It helps the identification of some bias due to high source values, for example, high gc content. This graphics could plot together using the ggplot2 geom\_violin method.

**Usage**

```
biasExploration(object, source = c("length", "gc", "pool"), dens = FALSE)
```

```
## S4 method for signature 'TargetExperiment'
biasExploration(object, source = c("length",
  "gc", "pool"), dens = FALSE)
```

**Arguments**

object	TargetExperiment class object.
source	Character 'gc', 'length', or 'pool' indicating the source bias. In the case of 'gc' and 'length', it will be categorized in four groups according to its quartiles. In the case of 'pool', its groups will be conserved.
dens	Logical indicating if density plot should be added using the geom_violin ggplot2 method.

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#), [plotFeatPerform](#)

**Examples**

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

# Attribute boxplot and density plot exploration
g<-biasExploration(ampliPanel,source="gc", dens=TRUE)
# x11(type="cairo")
if(interactive()){
  g
}
```

---

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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Examples**

```
## 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)
```

---

buildReport

*TargetExperiment* auxiliar function.

---

**Description**

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.

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

**Usage**

```
buildReport(object, attributeThres = c(0, 1, 50, 200, 500, Inf),
  imageFile = NULL, file= "Results.xlsx")

## 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)

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

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)

```

### Arguments

object	TargetExperiment class object.
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.
wb	A workbook object that will contain the report.

### Value

Workbook object.  
colours.

### Note

see full example in [TargetExperiment-class](#)

### Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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")

## Loading the TargetExperimentList object
data(ampliPanel, package="TarSeqQC")
colors<-ggplotColours(ampliPanel, n=5)
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")

```



```
colors<-ggplotColours(TEList, n=5)
```

---

checkBedFasta	<i>Function to control Bed and FASTA files compatibility.</i>
---------------	---

---

### Description

checkBedFasta checks the compatibility of a Bed file and a Fasta file. The functions first will control the consistency of the Bed file in terms of duplicated positions or feature's IDs and correct definition of start-end values. Then, the method will control the consistency between the specified features and the reference file. During its execution, several testing messages will be printed.

### Usage

```
checkBedFasta.bedFile, fastaFile)
```

### Arguments

bedFile	Character indicating the bed file full path.
fastaFile	Character indicating the full path to the genome reference file.

### Value

NULL

### Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

### See Also

[TargetExperiment-class](#)

### Examples

```
##Define the bed and fasta file full paths
bedFile<-system.file("extdata", "mybed.bed", package="TarSeqQC", mustWork=TRUE)
fastaFile<-system.file("extdata", "myfasta.fa", package="TarSeqQC",
  mustWork=TRUE)
##Checking the bed-fasta consistency
checkBedFasta.bedFile, fastaFile)
```

---

`getBedFile`*Getters for TargetExperiment and TargetExperimentList objects.*

---

**Description**

Obtain information about TargetExperiment and TargetExperimentList slots, 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)

getOverlappedRegions(object, collapse = FALSE)

## S4 method for signature 'TargetExperiment'
getOverlappedRegions(object, collapse = FALSE)

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

getPanels(object)

## S4 method for signature 'TargetExperimentList'
getPanels(object)

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

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

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

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

```

### Arguments

object	TargetExperiment/TargetExperimentList 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
data.frame	regions definition for overlapped features
GRanges	feature panels with statistical information

**Note**

see full example in [TargetExperiment-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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 featurePanel 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")
## Get the regions of overlapped features
getOverlappedRegions(ampliPanel, collapse=FALSE)
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")
## Get the bedFile slot
getBedFile(TEList)
## Get the panels slot
getPanels(TEList)
## Get the Feature slot
getFeature(TEList)
## Get the attribute slot
getAttribute(TEList)
## Get the region related to a feature or a gene
getRegion(TEList, level="gene", ID="gene7", collapse=FALSE)

## Get the low counts features
getLowCtsFeatures(TEList, level="feature")

```

---

initialize

*TargetExperiment object constructor.*


---

## Description

initialize creates the TargetExperiment object architecture for the specified bed and alignment 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 'ampli-con', '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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[TargetExperiment](#), [buildFeaturePanel](#) [summarizePanel](#)

Other TargetExperiment: [TargetExperiment-class](#), [TargetExperiment](#), [ampliPanel2](#), [ampliPanel](#), [myCounts](#)

**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)
}
```

---

initialize,TargetExperimentList-method

*TargetExperimentList object constructor.*

---

**Description**

initialize creates the TargetExperimentList object containing the experiment results of several targeted sequencing experiments carried out using a unique bed file.

**Usage**

```
## S4 method for signature 'TargetExperimentList'
initialize(.Object, TEList, feature = NULL,
          attribute = "coverage")
```

**Arguments**

.Object	TargetExperimentList class.
TEList	List containing all the TargetExperiment objects corresponding to the experiments that will be compared.
feature	Character indicating the name of the feature that will be explored (e.g 'amplicon', 'exon', 'gene').
attribute	Character indicating the name of the attribute that will be explored. Should be 'coverage' or 'medianCounts'.

**Value**

TargetExperimentList object.

**Note**

see full example in [TargetExperimentList-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[TargetExperimentList](#)

Other TargetExperimentList: [TargetExperimentList-class](#), [TargetExperimentList](#), [object](#)

**Examples**

```
# Defining the set of TargetExperiment objects
data(ampliPanel1, package="TarSeqQC")
data(ampliPanel2, package="TarSeqQC")
ampliList<-list(ampliPanel1, ampliPanel2)
# Defining feature parameter
feature<-"amplicon"
# Defining attribute parameter
attribute<-"coverage"
##Calling the constructor
object<-TargetExperimentList(TEList=ampliList, attribute=attribute,
                             feature=feature)
```

---

myCounts

*A pileup matrix example for use the TarSeqQC R package.*

---

## Description

The pileup matrix obtained using pileupCounts. It is built on the non-real dataset containing amplicon sequencing results to test the TarSeqQC package.

## Format

A data.frame object

## Details

**pos** genomic positions of the explored features.

**seqnames** chromosomes of the explored features.

**seq** reference nucleotide corresponding to the genomic position.

**A,C,G,T,N** number of nucleotide read.

**=** Amount of read nucleotides matching the reference nucleotide.

**-** Amount of read deletions.

**which\_label** feature location.

**counts** Total read counts

## Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

## Source

see [TargetExperiment-class](#)

## See Also

Other TargetExperiment: [TargetExperiment-class](#), [TargetExperiment](#), [ampliPanel2](#), [ampliPanel](#), [initialize](#)



---

object	<i>A set of two amplicon panels example for use the TarSeqQC R package.</i>
--------	---

---

### Description

A non-real dataset containing amplicon sequencing results to test the TarSeqQC package, principally the use of the TargetExperimentList class.

### Format

A TargetExperimentList object

### Details

**bedFile** Bed file containing 29 amplicons and 8 genes in 2 PCR pools.

**panels** GRanges obtaining amplicon coverage for two targeted sequencing experiment performed using the same bed file

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

### Source

see [TargetExperimentList-class](#)

### See Also

Other TargetExperimentList: [TargetExperimentList-class](#), [TargetExperimentList](#), [initialize](#), [TargetExperimentList](#)

---

pileupCounts	<i>Function to obtain the pileup counts for a bam file.</i>
--------------	---

---

### Description

pileupCounts waits for 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 set before, the number of next columns could change. If 'distinguish\_nucleotide' was set to TRUE, then one column per ntd will appear containing the counts obtained for each of them. Same will occur when 'distinguish\_strands' is set to 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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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

```
##Obtain the pileup matrix for the first amplicon  
data(ampliPanel, package="TarSeqQC")  
bed<-getBedFile(ampliPanel)[1]  
## 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 set interval. The resulting graph can be busy and might be better off saved.

For TargetExperimentList objects, plot allows a fast and simple representation of several feature panels using a heatmap plot. Along the x-axis, the features are represented and patients/samples along the y-axis. Finally, each cell is colored according to the attribute interval.

**Usage**

```
## S3 method for class 'TargetExperiment'
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, ...)

## S3 method for class 'TargetExperimentList'
plot(x, y, attributeThres = c(0, 1, 50,
  200, 500, Inf), pool = FALSE, sampleLabs = TRUE, featureLabs = FALSE,...)
```

**Arguments**

x	TargetExperiment/TargetExperimentList 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 the inner circle.
outerRadius	Numeric. Radius of the 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 center.
chrLabels	Logical. Chromosome names must be plotted?.
pool	Logical indicating if the plots should be performed for each pool separately
sampleLabs	Logical. Sample names must be plotted?.
featureLabs	Logical. Feature names must be plotted?.
...	not used but necessary for redefining the generic function.

**Value**

a ggplot2 graph.

**Note**

see full example in [TargetExperiment-class](#)

see full example in [TargetExperimentList-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**References**

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

**See Also**

[plotFeatPerform](#)

**Examples**

```
## 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)
if(interactive()){
  g
}

## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")
# Definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)

## Plot panel overview
g<-plot(TEList, attributeThres=attributeThres, featureLabs =TRUE)
if(interactive()){
  g
}
```

---

plotAttrExpl

*Plot attribute exploration of a TargetExperiment/TargetExperimentList object.*

---

**Description**

plotAttrExpl plots density and/or box-plot of the analyzed attribute at a feature level. These graphics could be displayed together using the ggplot2 geom\_violin method. If panel's pools are present, one facet for each pool will be showed.

**Usage**

```
plotAttrExpl(object, dens = FALSE, join = FALSE, log = TRUE,
             pool = FALSE, ...)

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

## S4 method for signature 'TargetExperimentList'
plotAttrExpl(object, dens = FALSE,
             join = FALSE, log = TRUE, pool = FALSE, attributeThres = NULL)
```

**Arguments**

object	TargetExperiment/TargetExperimentList class object.
dens	Logical indicating if density plot should be included
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.
pool	Logical indicating if plots should be displayed for each pool separately
...	necessary arguments
level	Character 'feature' or 'gene' indicating at which level should be analyzed the attribute.
color	A character indicating a valid name color.
attributeThres	Numeric indicating the attribute interval extreme values. It is not a mandatory parameter but if it is specified, then the plots will be colored according to the interval in which falls the attribute median values.

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#), [plotFeatPerform](#)

**Examples**

```
## 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")
if(interactive()){
  g
}
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")

# Attribute boxplot and density plot exploration
g<-plotAttrExpl(TEList, log=FALSE, pool=FALSE)
# x11(type="cairo")
if(interactive()){
  g
}
```

---

plotAttrPerform

*Plot feature performance of a TargetExperiment object.*


---

**Description**

plotAttrPerform plots the achieved performance for the selected attribute. The resulting graph shows one bar per each attribute interval and its height is defined according to the number of features achieving attribute values within that interval.

**Usage**

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

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

**Arguments**

**object** TargetExperiment class object.  
**attributeThres** Numeric indicating the intervals extreme values.

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#)

**Examples**

```
## 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<-plotAttrPerform(ampliPanel, attributeThres=attributeThres)
if(interactive()){
  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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#)

**Examples**

```
## 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)
if(interactive()){
  g
}
```

---

plotFeature

*Plot read profiles for a particular feature.*

---

**Description**

plotFeature plots the read profiles for a selected feature. The minAAF parameter set the minimum proportion value to call an SNP and the minRD the minimum read depth. They are combined to obtain a minimum read count value at each position used to distinguish between possible SNPs and background noise. If SNPs is set as 'TRUE', colored bars will appear indicating the occurrence of possible SNPs surpassing the minAAF and minRD, at each genomic position.

**Usage**

```
plotFeature(object, featureID, SNPs = TRUE, minAAF=0.05, minRD=10, xlab = "",
  title = "", size = 0.5, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
plotFeature(object, featureID, SNPs = TRUE,
  minAAF=0.05, minRD=10, 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.
minAAF	Numeric indicating the minimum alternative allele proportion necessary to call a SNP.
minRD	Numeric indicating the minimum read depth of alternative alleles necessary to call a SNP.
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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plotRegion](#)

**Examples**

```
## 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")
if(interactive()){
  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, overlap=FALSE, level="feature")

## S4 method for signature 'TargetExperiment'
plotGeneAttrPerFeat(object, geneID, overlap=FALSE,
  level="feature")
```

### Arguments

object	TargetExperiment object.
geneID	Character indicating the ID of the selected gene.
overlap	Logical indicating if the amplicons should be collapsed in overlapped regions.
level	Character indicating the level of the plot. Can be 'feature', to plot the features' attribute; 'region', to plot overlapped regions' attribute or 'both' to generate the two previous plots

### 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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

### See Also

[plotAttrExpl](#)

### Examples

```
## Loading the TargetExperiment object
data(ampliPanel, package="TarSeqQC")
## Exploring amplicon attribute values for a particular gene
# Ignoring amplicon overlapping
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))
if(interactive()){
  g
}
# Considering amplicon overlapping
g<-plotGeneAttrPerFeat(ampliPanel, geneID="gene4", overlap=TRUE, level="both")
# Adjust text size
g<-g+theme(title=element_text(size=16), axis.title=element_text(size=16),
legend.text=element_text(size=14))
if(interactive()){
  g
}

```

---

plotGlobalAttrExpl      *Plot attribute exploration of a TargetExperimentList object.*

---

### Description

plotGlobalAttrExpl displays box-plot of the analyzed achieved attribute values along all samples and at a feature level. This graphic could include density plot together the corresponding box-plot using the ggplot2 geom\_violin method.

### Usage

```

plotGlobalAttrExpl(object, attributeThres = c(0, 1, 50, 200, 500, Inf),
  dens = FALSE, log = FALSE, pool = FALSE, featureLabs = FALSE,
  medianMarg = NULL)

## S4 method for signature 'TargetExperimentList'
plotGlobalAttrExpl(object,
  attributeThres = c(0, 1, 50, 200, 500, Inf), dens = FALSE, log = FALSE,
  pool = FALSE, featureLabs = FALSE, medianMarg = NULL)

```

### Arguments

object	TargetExperimentList class object.
attributeThres	Numeric indicating the attribute interval extreme values.
dens	Logical indicating if boxplot and density function should be plotted together using the ggplot2 geom_violin method or only the boxplot (dens=FALSE) should be displayed.
log	Logical indicating if the attribute should be considered in log10 scale.
pool	Logical indicating if the plots should be performed for each pool separately
featureLabs	logical indicating if feature names should be plotted
medianMarg	numeric indicating the percentage of the median attribute value to be plotted as lines. If it is NULL no line will be displayed

### Value

ggplot2 graphics.

**Note**

see full example in [TargetExperimentList-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#)

**Examples**

```
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")

# Attribute boxplot and density plot exploration
g<-plotGlobalAttrExpl(TEList,log=FALSE)
# x11(type="cairo")
if(interactive()){
  g
}
```

---

plotInOutFeatures	<i>Function to explore read percentages in targeted regions and out targeted regions.</i>
-------------------	---

---

**Description**

plotInOutFeatures allows the graphical exploration of the data frame obtained using readFrequencies. This data frame contains information about the amount of reads mapped to the targeted regions and out of them. This information is presented in rows, one for each chromosome and in absolute and relative amounts. After its invocation, a bar plot built as a ggplot object is returned

**Usage**

```
plotInOutFeatures(object, ...)
```

## S4 method for signature 'data.frame'

```
plotInOutFeatures(object, absolute = FALSE)
```

## S4 method for signature 'TargetExperiment'

```
plotInOutFeatures(object, absolute = FALSE,
  BPPARAM = bpparam())
```

**Arguments**

object	a data frame or a TargetExperiment.
...	additional parameters according to the function call
absolute	logical indicating if absolute frequency should be used.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

**Value**

ggplot object.

**Note**

see full example in [TargetExperiment-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Examples**

```
## 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)

g<-plotInOutFeatures(ampliPanel)
```

---

plotMetaDataExpl

*Graphical exploration of a specific metadata column.*

---

**Description**

plotMetaDataExpl plots density and box-plot of a specific metadata column. If the characteristic is nonnumerical, then a frequency plot is built.

**Usage**

```
plotMetaDataExpl(object, name = c("length", "gc", "pool"), log = FALSE,
  join = TRUE, absolute = FALSE, color = "blue")

## S4 method for signature 'TargetExperiment'
plotMetaDataExpl(object, name = c("length", "gc",
  "pool"), log = FALSE, join = TRUE, absolute = FALSE, color = "blue")
```

**Arguments**

object	TargetExperiment class object.
name	a character indicating the metadata column name that should be analyzed.
log	Logical indicating if the numerical metadata column should be considered in log10 scale.
join	Logical only for numerical variables. It indicates if boxplot and density function should be plotted together using the ggplot2 geom_violin method.
absolute	Logical indicating if the frequencies of the selected categorical metadata column should be in absolute scale. If absolute is FALSE the frequencies are in relative percentages.
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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plot](#), [plotFeatPerform](#)

**Examples**

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

# Attribute boxplot and density plot exploration
g<-plotMetaDataExpl(ampliPanel, name="length")
if(interactive())
{
# x11(type="cairo")
g
}
# Explore amount of amplicons per gene
g<-plotMetaDataExpl(ampliPanel, name="gene", absolute=TRUE)
if(interactive())
{
# x11(type="cairo")
g
}
```

---

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

---

### Description

plotNtdPercentage plots the percentages of the occurrence 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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

### See Also

[plotFeature](#)

### Examples

```
## 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")
if(interactive()){
  g
}
```

---

plotPoolPerformance     *Plot pool performance of a TargetExperimentList object.*

---

### Description

plotPoolPerformance plots density and/or box-plot of the analyzed attribute achieved in each PCR pool. These graphics could be displayed together using the ggplot2 geom\_violin method.

### Usage

```
plotPoolPerformance(object, dens = FALSE, join = FALSE, log = TRUE,
  attributeThres = NULL)
```

```
## S4 method for signature 'TargetExperimentList'
plotPoolPerformance(object, dens = FALSE,
  join = FALSE, log = TRUE, attributeThres = NULL)
```

### Arguments

object	TargetExperimentList class object.
dens	Logical indicating if density plot should be included
join	Logical indicating if boxplot and density function should be plotted together using the ggplot2 geom_violin method. For it uses, dens should be TRUE.
log	Logical indicating if the attribute should be considered in log10 scale.
attributeThres	Numeric indicating the attribute interval extreme values. It is not a mandatory parameter but if it is specified, then the plots will be colored according to the interval in which falls the attribute median values.

### Value

ggplot2 graphics.

### Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>



**Examples**

```
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")
# Attribute boxplot and density plot exploration
g<-plotPoolPerformance(TEList,log=FALSE)
# x11(type="cairo")
if(interactive()){
  g
}
```

---

plotRegion

*Plot read profiles for a particular genomic region.*


---

**Description**

plotRegion plots the read profiles for a selected region. The minAAF parameter set the minimum proportion value to call an SNP and the minRD the minimum read depth. They are combined to obtain a minimum read count value at each position used to distinguish between possible SNPs and background noise. If SNPs is set as 'TRUE', colored bars will appear indicating the occurrence of possible SNPs surpassing the minAAF and minRD, at each genomic position.

**Usage**

```
plotRegion(object, region, seqname, SNPs = TRUE, minAAF=0.05, minRD=10,
           xlab = "", title = "", size = 0.5, BPPARAM = bpparam())

## S4 method for signature 'TargetExperiment'
plotRegion(object, region, seqname, SNPs = TRUE,
           minAAF=0.05, minRD=10, 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.
minAAF	Numeric indicating the minimum alternative allele proportion necessary to call a SNP.
minRD	Numeric indicating the minimum read depth of alternative alleles necessary to call a SNP.
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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[plotFeature](#)

**Examples**

```
## 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")
if(interactive()){
  g
}
```

---

print

*Print a TargetExperiment/TargetExperimentList object.*

---

**Description**

Generic print method for TargetExperiment and TargetExperimentList classes and descendants.

**Usage**

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

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

**Arguments**

x TargetExperiment/TargetExperimentList class object.  
 ... Included for generic print compatibility.

**Value**

console output of the object.

**Note**

see full example in [TargetExperiment-class](#)  
 see full example in [TargetExperimentList-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Examples**

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

---

readFrequencies	<i>Function to explore read frequencies in targeted regions and out targeted regions.</i>
-----------------	---

---

**Description**

readFrequencies builds a data frame containing the read frequencies falling in targeted regions and out of those, separated by chromosome.

**Usage**

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

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

**Arguments**

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

**Value**

data.frame object.

**Note**

see full example in [TargetExperiment-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Examples**

```
## 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)

myReadPercentages<-readFrequencies(ampliPanel)
```

---

setFeature<-

*Setters for the TargetExperiment slots*

---

**Description**

Set TargetExperiment slots, according to the given function call.

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

### Arguments

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

### Value

a TargetExperiment object

### Note

see full example in [TargetExperiment-class](#)  
see full example in [TargetExperimentList-class](#)

### Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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()
}
## loading TargetExperimentList object
data(TEList, package="TarSeqQC")
## Set feature slot value
setFeature(TEList)<-"amplicon"
```

---

show	<i>Show method for the TargetExperiment and TargetExperimentList classes.</i>
------	---

---

**Description**

show a TargetExperiment/TargetExperimentList object

**Usage**

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

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

**Arguments**

object            TargetExperiment/TargetExperimentList class object

**Details**

Generic show method for TargetExperiment and TargetExperimentList classes output visualization.

**Value**

console output of the object

**Note**

see full example in [TargetExperiment-class](#)

see full example in [TargetExperimentList-class](#)

**Author(s)**

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**Examples**

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

---

summarizePanel	<i>Function to summarize a featurePanel slot at a gene level.</i>
----------------	---

---

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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 and TargetExperimentList summary methods.*

---

**Description**

Explore the TargetExperiment and TargetExperimentList'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),
  pool = FALSE)

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

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

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

**Arguments**

object            TargetExperiment/TargetExperimentList class object.  
 ...              required by summary.



attributeThres numeric indicating the intervals extreme values required by summaryIntervals.  
 pool logical indicating if the summary should be performed for each pool separately

### 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](#)

see full example in [TargetExperimentList-class](#)

### Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.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)
## Loading the TargetExperimentList object
data(TEList, package="TarSeqQC")
# Object summary
summary(TEList)
# Defining the attribute interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)
# Doing a frequency table for the attribute intervals
summaryIntervals(TEList, attributeThres=attributeThres)
```

---

TargetExperiment      *TargetExperiment constructor*

---

### Description

TargetExperiment creates a TargetExperiment object with the architecture specified by the bed and alignment 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 set 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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[TargetExperiment-class1](#)

Other TargetExperiment: [TargetExperiment-class](#), [ampliPanel2](#), [ampliPanel](#), [initialize](#), [myCounts](#)

**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"
##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** set the respective TargetExperiment slots

**show** generic output of the object

**print** generic output of the object

**summary** print statistics summary for the set 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

**readFrequencies** calculate frequencies of reads fall in and out of targeted regions

**plotInOutFeatures** plot frequencies of reads fall in and out of targeted regions

**biasExploration** plot attribute distributions along groups of bias sources

**plotMetaDataExpl** plot density and box plots or frequency bar plot of metadata columns

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

## See Also

Rsamtools

Other TargetExperiment: [TargetExperiment](#), [ampliPanel2](#), [ampliPanel](#), [initialize](#), [myCounts](#)

**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")
if(interactive()){
  x11(type="cairo");g
}
# explore amplicon length distribution
g<-plotMetaDataExpl(ampliPanel, "length", log=FALSE, join=FALSE, color="blueviolet")
if(interactive()){
  g
}
# explore gene's relative frequencies
g<-plotMetaDataExpl(ampliPanel, "gene", abs=FALSE)
if(interactive()){
  g
}
## Deep exploration and Quality Control
myfrequencies<-readFrequencies(ampliPanel)

```

```

g<-plotInOutFeatures(readFrequencies(ampliPanel))
if(interactive()){
g
}
# definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)
# plot panel overview
g<-plot(ampliPanel, attributeThres, chrLabels =TRUE)
if(interactive()){
x11(type="cairo");g
}
# plot panel overview in a feature performance plot
g<-plotFeatPerform(ampliPanel, attributeThres, complete=TRUE, log=FALSE,
featureLabs=TRUE, sepChr=TRUE, legend=TRUE)
if(interactive()){
g
}
# explore possible attribute bias
g<-biasExploration(ampliPanel, source="gc", dens=TRUE)
if(interactive()){
x11(type="cairo");g
}
## Controlling low counts features
# Do a frequency table for the attribute intervals
summaryIntervals(ampliPanel, attributeThres)
#plotting attribute intervals
g<-plotAttrPerform(ampliPanel)
if(interactive()){
g
}
# 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))
if(interactive()){
g
}
##Obtain the pileup matrix for the first amplicon
bed<-getBedFile(ampliPanel)[1]
## extracting the pileup matrix
myCounts<-pileupCounts(bed, bamFile, fastaFile)
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)
if(interactive()){
x11(type="cairo");g
}
# exploring the read count profile for a particular amplicon
g<-plotFeature(ampliPanel, featureID="AMPL20")

```

```

if(interactive()){
x11(type="cairo");g
}
# exploring the nucleotide percentages compositions of the read counts for a
# particular amplicon
g<-plotNtdPercentage(ampliPanel,featureID="AMPL20")
if(interactive()){
g
}
## Building the XLSX report
imageFile<-system.file("extdata", "plot.pdf", package="TarSeqQC",
mustWork=TRUE)
buildReport(ampliPanel, attributeThres, imageFile ,file="Results.xlsx")

```

---

TargetExperimentList    *TargetExperimentList constructor*

---

## Description

TargetExperimentList creates a TargetExperimentList object containing a set of targeted sequencing experiment results, all those, carried out using the same bed file. Feature parameter specifies what represent each panel element (bed file row). Attribute parameter indicates which attribute would be analyzed, 'coverage' or 'medianCounts' and should be specified in order to indicate which coverage or medianCounts should be conserved.

## Usage

```
TargetExperimentList(TEList, feature = NULL, attribute = "coverage")
```

## Arguments

TELlist	List containing all the TargetExperiment objects corresponding to the experiments that will be compared.
feature	Character indicating the name of the feature that will be explored (e.g 'amplicon', 'transcript', 'gene').
attribute	Character indicating the name of the attribute that will be explored. Should be 'coverage' or 'medianCounts'.

## Value

TargetExperimentList object.

## Note

see full example in [TargetExperimentList-class](#)

## Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

[TargetExperimentList-class1](#)

Other TargetExperimentList: [TargetExperimentList-class](#), [initialize](#), [TargetExperimentList-method](#), [object](#)

**Examples**

```
# Defining the set of TargetExperiment objects
data(ampliPanel, package="TarSeqQC")
data(ampliPanel2, package="TarSeqQC")
ampliList<-list(ampliPanel, ampliPanel2)
# Defining feature parameter
feature<-"amplicon"
# Defining attribute parameter
attribute<-"coverage"
##Calling the constructor
object<-TargetExperimentList(TEList=ampliList, attribute=attribute,
feature=feature)
```

---

TargetExperimentList-class

*TargetExperimentList S4 class implementation in R*

---

**Description**

This S4 class represents a collection of Targeted Sequencing Experiments in R. All these experiments are characterized by a 'bed file' containing the specification of the explored 'features', as a 'feature panel'. These features could be amplicons, exons, transcripts, among others. In general each feature is associated to one gene but a gene could be related to many features. This class allows the representation and quality control of a set of Targeted Sequencing Experiment made over the same or different subjects but using always the same bed file'.

**Slots**

`bedFile` GRanges object that models the bed file.

`panels` GRanges object containing the feature/gene panels.

`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', 'gene'.

**Features**

1. Model sets of targeted sequencing experiments in R.
2. Evaluate the performance of the targeted sequencing technique across several experiments using coverage/read counts information.
3. Detect in early stage sequencing or library preparation errors.
4. Report quality control results.



## Functions

TargetExperimentList S4 class includes the following functions:

**initialize** constructor of TargetExperimentList to generate the feature panel starting from at least two TargetExperiment objects

**getBedFile, getPanels, getAttribute, getFeature** return the respective TargetExperimentList slots

**setFeature** set the respective TargetExperimentList slot

**show** generic output of the object

**print** generic output of the object

**summary** print statistics summary for the set attribute

**plot** plot a summarized view of the attribute values achieved by each feature in each sample

**plotGlobalAttrExpl** plot the attribute distribution for each feature

**plotAttrExpl** plot the attribute distribution in each panel

**plotpoolPerformance** plot the attribute distribution in each or pool

## Author(s)

Gabriela A. Merino <gmerino@bdmg.com.ar>, Cristobal Fresno <cfresno@bdmg.com.ar>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

## See Also

Rsamtools

Other TargetExperimentList: [TargetExperimentList](#), [initialize](#), [TargetExperimentList-method](#), [object](#)

## Examples

```
# Defining the set of TargetExperiment objects
data(ampliPanel1, package="TarSeqQC")
data(ampliPanel2, package="TarSeqQC")
ampliList<-list(ampliPanel1, ampliPanel2)
# Defining feature parameter
feature<-"amplicon"
# Defining attribute parameter
attribute<-"coverage"
##Calling the constructor
object<-TargetExperimentList(TEList=ampliList, attribute=attribute,
                             feature=feature)
setFeature(object)<-"amplicon"
## load the example dataset
data(TEList, package="TarSeqQC")
## Early exploration
# show/print
TELList
# summary
summary(TELList)
## Controlling low counts features
# Definition of the interval extreme values
attributeThres<-c(0,1,50,200,500, Inf)
# Do a frequency table for the attribute intervals
```

```

summaryIntervals(TEList, attributeThres)
# getting low counts features at gene level
getLowCtsFeatures(TEList, level="gene", threshold=50)
# exploring panel performance along several samples
g<-plot(TEList, attributeThres=attributeThres, featureLabs =TRUE)
if(interactive()){
  g
}
g<-plotGlobalAttrExpl(TEList,log=FALSE)
# x11(type="cairo")
if(interactive()){
  g
}
g<-plotPoolPerformance(TEList,log=FALSE)
if(interactive()){
  g
}

```

---

TEList

*A set of two amplicon panels example for use the TarSeqQC R package.*


---

### Description

A non-real dataset containing amplicon sequencing results to test the TarSeqQC package, principally the use of the TargetExperimentList class.

### Format

A TargetExperimentList object

### Details

**bedFile** Bed file containing 29 amplicons and 8 genes.

**panels** GRanges obtaining amplicon coverage for two targeted sequencing experiment performed using the same bed file

**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>, Yanina Murua <ymurua@leloir.org.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

### Source

see [TargetExperimentList-class](#)

### See Also

Other TargetExperimentList: [TargetExperimentList-class](#), [TargetExperimentList](#), [initialize](#), [TargetExperimentList](#)

# Index

# (summaryFeatureLev), 40

addStatSummSheet (buildReport), 7

addStatSummSheet, TargetExperiment-method  
(buildReport), 7

addStatSummSheet-methods (buildReport),  
7

ampliPanel, 4, 5, 14, 16, 42, 44

ampliPanel2, 4, 4, 14, 16, 42, 44

biasExploration, 5

biasExploration, TargetExperiment-method  
(biasExploration), 5

biasExploration-methods  
(biasExploration), 5

buildFeaturePanel, 6, 14, 40

buildFeaturePanel, TargetExperiment-method  
(buildFeaturePanel), 6

buildFeaturePanel-methods  
(buildFeaturePanel), 6

buildReport, 7

buildReport, TargetExperiment-method  
(buildReport), 7

buildReport-methods (buildReport), 7

checkBedFasta, 9

checkBedFasta-methods (checkBedFasta), 9

getAttribute (getBedFile), 10

getAttribute, TargetExperiment-method  
(getBedFile), 10

getAttribute, TargetExperimentList-method  
(getBedFile), 10

getAttribute-methods (getBedFile), 10

getBamFile (getBedFile), 10

getBamFile, TargetExperiment-method  
(getBedFile), 10

getBamFile-methods (getBedFile), 10

getBedFile, 10

getBedFile, TargetExperiment-method  
(getBedFile), 10

getBedFile, TargetExperimentList-method  
(getBedFile), 10

getBedFile-methods (getBedFile), 10

getFastaFile (getBedFile), 10

getFastaFile, TargetExperiment-method  
(getBedFile), 10

getFastaFile-methods (getBedFile), 10

getFeature (getBedFile), 10

getFeature, TargetExperiment-method  
(getBedFile), 10

getFeature, TargetExperimentList-method  
(getBedFile), 10

getFeature-methods (getBedFile), 10

getFeaturePanel (getBedFile), 10

getFeaturePanel, TargetExperiment-method  
(getBedFile), 10

getFeaturePanel-methods (getBedFile), 10

getGenePanel (getBedFile), 10

getGenePanel, TargetExperiment-method  
(getBedFile), 10

getGenePanel-methods (getBedFile), 10

getLowCtsFeatures (getBedFile), 10

getLowCtsFeatures, TargetExperiment-method  
(getBedFile), 10

getLowCtsFeatures, TargetExperimentList-method  
(getBedFile), 10

getLowCtsFeatures-methods (getBedFile),  
10

getOverlappedRegions (getBedFile), 10

getOverlappedRegions, TargetExperiment-method  
(getBedFile), 10

getPanels (getBedFile), 10

getPanels, TargetExperimentList-method  
(getBedFile), 10

getPileupP (getBedFile), 10

getPileupP, TargetExperiment-method  
(getBedFile), 10

getPileupP-methods (getBedFile), 10

getRegion (getBedFile), 10

getRegion, TargetExperiment-method  
(getBedFile), 10

getRegion, TargetExperimentList-method  
(getBedFile), 10

getRegion-methods (getBedFile), 10

getScanBamP (getBedFile), 10

getScanBamP, TargetExperiment-method

- (getBedFile), 10
- getScanBamP-methods (getBedFile), 10
- ggplotColours (buildReport), 7
- ggplotColours, TargetExperiment-method (buildReport), 7
- ggplotColours, TargetExperimentList-method (buildReport), 7
- ggplotColours-methods (buildReport), 7
- initialize, 4, 5, 13, 16, 42, 44
- initialize, TargetExperiment-method (initialize), 13
- initialize, TargetExperimentList-method, 14
- myCounts, 4, 5, 14, 16, 42, 44
- Object (summaryFeatureLev), 40
- object, 15, 17, 48, 49
- pileupCounts, 17
- pileupCounts-methods (pileupCounts), 17
- plot, 6, 19, 21, 23, 24, 28, 30
- plot, TargetExperiment (plot), 19
- plot, TargetExperimentList (plot), 19
- plot.TargetExperiment (plot), 19
- plot.TargetExperimentList (plot), 19
- plotAttrExpl, 20, 26
- plotAttrExpl, TargetExperiment-method (plotAttrExpl), 20
- plotAttrExpl, TargetExperimentList-method (plotAttrExpl), 20
- plotAttrExpl-methods (plotAttrExpl), 20
- plotAttrPerform, 22
- plotAttrPerform, TargetExperiment-method (plotAttrPerform), 22
- plotAttrPerform-methods (plotAttrPerform), 22
- plotFeatPerform, 6, 20, 21, 23, 30
- plotFeatPerform, TargetExperiment-method (plotFeatPerform), 23
- plotFeatPerform-methods (plotFeatPerform), 23
- plotFeature, 24, 31, 34
- plotFeature, TargetExperiment-method (plotFeature), 24
- plotFeature-methods (plotFeature), 24
- plotGeneAttrPerFeat, 26
- plotGeneAttrPerFeat, TargetExperiment-method (plotGeneAttrPerFeat), 26
- plotGeneAttrPerFeat-methods (plotGeneAttrPerFeat), 26
- plotGlobalAttrExpl, 27
- plotGlobalAttrExpl, TargetExperimentList-method (plotGlobalAttrExpl), 27
- plotGlobalAttrExpl-methods (plotGlobalAttrExpl), 27
- plotInOutFeatures, 28
- plotInOutFeatures, data.frame-method (plotInOutFeatures), 28
- plotInOutFeatures, TargetExperiment-method (plotInOutFeatures), 28
- plotMetaDataExpl, 29
- plotMetaDataExpl, TargetExperiment-method (plotMetaDataExpl), 29
- plotMetaDataExpl-methods (plotMetaDataExpl), 29
- plotNtdPercentage, 31
- plotNtdPercentage, TargetExperiment-method (plotNtdPercentage), 31
- plotNtdPercentage-methods (plotNtdPercentage), 31
- plotPoolPerformance, 32
- plotPoolPerformance, TargetExperimentList-method (plotPoolPerformance), 32
- plotPoolPerformance-methods (plotPoolPerformance), 32
- plotRegion, 25, 33
- plotRegion, TargetExperiment-method (plotRegion), 33
- plotRegion-methods (plotRegion), 33
- print, 34
- print, TargetExperiment-method (print), 34
- print, TargetExperimentList-method (print), 34
- readFrequencies, 35
- readFrequencies, TargetExperiment-method (readFrequencies), 35
- readFrequencies-methods (readFrequencies), 35
- setAttribute<- (setFeature<-), 36
- setAttribute<-, TargetExperiment, character-method (setFeature<-), 36
- setAttribute<--methods (setFeature<-), 36
- setBamFile<- (setFeature<-), 36
- setBamFile<-, TargetExperiment, character-method (setFeature<-), 36
- setBedFile<- (setFeature<-), 36
- setBedFile<-, TargetExperiment, character-method (setFeature<-), 36
- setBedFile<--methods (setFeature<-), 36
- setFastaFile<- (setFeature<-), 36

- setFastaFile<- , TargetExperiment, character-method (summaryFeatureLev), 40  
(setFeature<-), 36 summaryGeneLev-methods
- setFastaFile<--methods (setFeature<-), (summaryFeatureLev), 40
- setFeature<- , 36 summaryIntervals (summaryFeatureLev), 40
- setFeature<- , TargetExperiment, character-method (summaryFeatureLev), 40  
(setFeature<-), 36 summaryIntervals, TargetExperiment-method
- setFeature<- , TargetExperimentList, character-method (summaryFeatureLev), 40  
(setFeature<-), 36 summaryIntervals-methods
- setFeature<--methods (setFeature<-), 36 (summaryFeatureLev), 40
- setFeaturePanel<- (setFeature<-), 36
- setFeaturePanel<- , TargetExperiment, GRanges-method TargetExperiment, 4, 5, 14, 16, 40, 41, 44  
(setFeature<-), 36 TargetExperiment-class, 43
- setFeaturePanel<--methods TargetExperiment-methods  
(setFeature<-), 36 (TargetExperiment), 41
- setGenePanel<- (setFeature<-), 36 TargetExperimentList, 15, 17, 47, 49, 50
- setGenePanel<- , TargetExperiment, GRanges-method TargetExperimentList-class, 48  
(setFeature<-), 36 TargetExperimentList-method
- setGenePanel<--methods (setFeature<-), (initialize, TargetExperimentList-method),  
36 14
- setPileupP<- (setFeature<-), 36 TargetExperimentList-methods  
(setFeature<-), 36 (TargetExperimentList), 47
- setPileupP<- , TargetExperiment, PileupParam-method TargetExperimentList-methods  
(setFeature<-), 36 TarSeqQC-package, 3  
TEList, 50
- setPileupP<--methods (setFeature<-), 36
- setScanBamP<- (setFeature<-), 36
- setScanBamP<- , TargetExperiment, ScanBamParam-method  
(setFeature<-), 36
- setScanBamP<--methods (setFeature<-), 36
- setsetBamFile<- (setFeature<-), 36
- setsetBamFile<--methods (setFeature<-),  
36
- show, 38
- show, TargetExperiment-method (show), 38
- show, TargetExperimentList-method  
(show), 38
- summarizePanel, 14, 39
- summarizePanel, TargetExperiment-method  
(summarizePanel), 39
- summarizePanel-methods  
(summarizePanel), 39
- summary (summaryFeatureLev), 40
- summary, TargetExperiment-method  
(summaryFeatureLev), 40
- summary, TargetExperimentList-method  
(summaryFeatureLev), 40
- summaryFeatureLev, 40
- summaryFeatureLev, TargetExperiment-method  
(summaryFeatureLev), 40
- summaryFeatureLev-methods  
(summaryFeatureLev), 40
- summaryGeneLev (summaryFeatureLev), 40
- summaryGeneLev, TargetExperiment-method