Package 'ampliQueso'

April 4, 2014

Type Package

Title Analysis of amplicon enrichment panels

Version 1.0.0

Date 2013-08-06

Author Alicja Szabelska <alicja.szabelska@up.poznan.pl>; Marek Wiewiorka <marek.wiewiorka@gmail.com>; Michal Okoniewski <mic

Maintainer Michal Okoniewski <michal@fgcz.ethz.ch>

Depends R (>= 2.15.0), rnaSeqMap (>= 2.17.1), knitr, rgl, ggplot2,gplots, parallel, doParallel, foreach,VariantAnnotation,genefilter,statmod,xtable

Imports edgeR, DESeq, samr

Suggests

Description The package provides tools and reports for the analysis of amplicon sequencing panels, such as AmpliSeq

License GPL-2

URL

biocViews Bioinformatics, ReportWriting, Transcription,GeneExpression, DifferentialExpression,HighThroughputSequencing, RNAseq, Visualization

Collate compareCoverages.R getCountTable.R getSNP.R auxFunctions.R compareCoveragesRegions.R runAQReport.R getCamelTests.R

R topics documented:

`camelSampleTable` *ampliQueso sample data*

Description

Sample camel/coverage measures for 2 regions.

Usage

```
data(ampliQueso)
```

Format

A table with 6 variables and 2 values.

`x` a numeric vector

`y` a numeric vector

Examples

```
data(ampliQueso)
```

`camelTest` *camelTest - calculating permutation tests for camel measures (sequential and parallel versions)*

Description

Calculating permutation tests for reads coverage (camel) measures for regions specified by a BED file. This function features a few normalization modes and camel measures.

Usage

```
camelTest(iBedFile, iCovdesc = "covdesc", iT1, iT2, iNorm = c("none", "density", "minMax"), iMeasure = c
```

Arguments

iBedFile	BED file with genomic coordinates refering to the same genome as the BAMs
iCovdesc	Covdesc-like file - BAM files are read from row names. Similar to covdesc in simpleaffy and rnaSeqMap - tab delimited table of BAM files and groupings. This is the table of experimental design.
iT1	name of the first group from the experiment (it should be consistent with the groupings specified in covdesc file)
iT2	name of the other group from the experiment (it should be consistent with the groupings specified in covdesc file)
iNorm	vector specifing the data normalisations prior to camel measures calculations
iMeasure	camel measure to be used in the non-parametric test. Camel measures are implemented in the package rnaSeqMap and described in Okoniewski et al, NAR, 2011
iSizes	TBD
iNPerm	integer number specifing the number of permutations for camel tests calcualtions
iParallel	boolean specifing whether calculations shoulb be run in parallel,true by default, The degree of parallelism is set up automagically and by default equals to the number of logical CPU cores.

Value

data frame with p-values of camel tests for all regions, measures and data normalisations specified

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

---

compareCoverages *compareCoverages - camel measure for a single comparison.*

Description

Comparing coverage profiles with the camel measures for the two groups of BAM files, for a given genomic region.

Usage

```
compareCoverages(ch, st, en, group, t1, t2, measure="DA", covdesc="covdesc")
```

Arguments

ch	Chromosome of the region of interest
st	Start of the region of interest
en	End of the region of interest
group	Name of the attribute/group in the experimental description (i.e. covdesc)
t1	First name of the attribute used for grouping
t2	Another name of the attribute used for grouping
measure	Method of comparison for the coverages in the selected genomic region
covdesc	Covdesc-like data frame - BAM files are read from row names

Value

A single camel value with the results of comparison.

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

---

compareCoveragesReg *compareCoveragesReg - calculating given camel measures for a list of regions specified by a BED file (sequential and parallel versions)*

Description

Calculating given camel measures for a list of regions specified by a BED file (sequential version)

Usage

```
compareCoveragesReg(iBedFile, iGroup, iT1, iT2, iMeasure = c("DA", "QQ", "PP", "HD1", "HD2"), iCovdesc
```

Arguments

iBedFile	BED file with same genomic coordinates as in BAMs (preferably standard hg19)
iGroup	Name of the attribute/group in the experimental description (i.e. covdesc)
iT1	name of the first group from the experiment (it should be consistent with the groupings specified in covdesc file)
iT2	name of the other group from the experiment (it should be consistent with the groupings specified in covdesc file)
iMeasure	vector specifing camel measures to be calculated
iCovdesc	Covdesc-like data frame - BAM files are read from row names. Similar to covdesc in simpleaffy and rnaSeqMap - tab delimited table of BAM files and groupings. This is the table of experimental design.
iParallel	boolean specifing whether calculations shoulb be run in parallel,true by default, The degree of parallelism is set up automagically and by default equals to the number of logical CPU cores.

Value

data frame with camel measures calculated for the specified regions and samples

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

`getCountTable` *getCountTable - obtaining typical count table.*

Description

Obtaining a count table for all the amplicons described in the BED file.

Usage

```
getCountTable(covdesc="covdesc", bedFile="amplicons.bed")
```

Arguments

`covdesc`	Covdesc-like data frame - BAM files are read from row names. Similar to covdesc in simpleaffy and rnaSeqMap - tab delimited table of BAM files and groupings. This is the table of experimental design.
`bedFile`	BED file with same genomic coordinates as in BAMs (preferably standard hg19)

Value

Count table with the number of columns the same as number of samples and the number of rows the same as number of amplicons.

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

Examples

```
library(ampliQueso)
setwd(path.package("ampliQueso"))
cc <- getCountTable(covdesc=system.file("extdata","covdesc",package="ampliQueso") , bedFile=system.file("extdata
```

getSNP *getSNP - finding SNPs in all samples for regions specified by a BED file all regions if BED file not provided (sequential and parallel versions)*

Description

Finding SNPs in all samples for regions specified by a BED file (sequential version) using Samtools mpileup functionality. Please make sure that samtools executables are installed and available from command line without specifing the full path(e.g. added to the PATH environment variable)

Usage

```
getSNP(covdesc, minQual, refSeqFile, bedFile = NULL,iParallel=TRUE)
```

Arguments

covdesc	Covdesc-like data frame - BAM files are read from row names. Similar to covdesc in simpleaffy and rnaSeqMap - tab delimited table of BAM files and groupings. This is the table of experimental design.
minQual	number specifing the minimum quality of the SNPs returned by samtools used to filter out these of poor quality. Please refer to Samtools documentation for further details on the definition of this measure.
refSeqFile	Reference sequence in FASTA format used for SNPs detection.
bedFile	BED file with same genomic coordinates as in BAMs (preferably standard hg19)
iParallel	boolean specifing whether calculations shoulb be run in parallel,true by default, The degree of parallelism is set up automagically and by default equals to the number of logical CPU cores.

Value

list of data frames, for one sample each containing coordinates of the SNP with its quality

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

ndMax *ampliQueso sample data*

Description

Nucleotide distribution object for a region with the largest camel measures difference between samples.

Usage

```
data(ampliQueso)
```

Format

The format is: Formal class 'NucleotideDistr' [package "rnaSeqMap"] with 13 slots ..@ chr : chr "chr1" ..@ start : num 1.93e+08 ..@ end : num 1.93e+08 ..@ strand : num 1 ..@ type : chr(0) ..@ data :List of 8 .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:68] 0 15 16 159 164 165 167 170 175 176 ... .@ lengths : int [1:68] 11 10 1 1 2 2 1 3 1 2 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:45] 0 4 5 6 77 80 81 82 83 84 ... .@ lengths : int [1:45] 11 7 3 1 1 2 1 8 3 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:78] 0 14 15 230 234 238 239 240 243 245 ... .@ lengths : int [1:78] 11 10 1 1 2 1 1 1 3 2 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:49] 0 1 11 12 95 97 98 100 101 103 ... .@ lengths : int [1:49] 10 1 10 1 3 1 2 7 1 2 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:64] 0 8 9 10 118 121 124 125 127 131 ... .@ lengths : int [1:64] 11 5 4 2 1 2 1 1 1 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:37] 0 4 47 48 49 50 51 52 53 55 ... .@ lengths : int [1:37] 11 11 1 3 10 2 6 1 1 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:77] 0 1 26 28 252 259 262 264 269 271 ... .@ lengths : int [1:77] 10 1 10 1 1 2 1 2 2 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:71] 0 16 167 171 174 175 177 179 180 181 ... .@ lengths : int [1:71] 11 11 1 2 1 2 1 2 1 3 ... .@ elementMetadata: NULL .. ..@ metadata : list() ..@ assayData :<environment: 0x104fe810> ..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots .. .. ..@ name : chr "" .. .. ..@ lab : chr "" .. .. ..@

contact : chr "" .. .. ..@ title : chr "" .. .. ..@ abstract : chr "" .. .. ..@ url : chr "" .. .. ..@ pubMedIds : chr "" .. .. ..@ samples : list() .. .. ..@ hybridizations : list() .. .. ..@ normControls : list() .. .. ..@ preprocessing : list() .. .. ..@ other : list() .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 2 .. ..$: int [1:3] 1 0 0 .. ..$: int [1:3] 1 1 0 ..@ annotation : chr(0) ..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. .. ..@ .Data:List of 3 .. ..$: int [1:3] 3 0 1 .. ..$: int [1:3] 2 20 1 .. ..$: int [1:3] 1 3 0

Examples

```
data(ampliQueso)
```

---

ndMin *ampliQueso sample data*

Description

Nucleotide distribution object for a region with the lowest camel measures difference between samples.

Usage

```
data(ampliQueso)
```

Format

The format is: Formal class 'NucleotideDistr' [package "rnaSeqMap"] with 13 slots ..@ chr : chr "chr2" ..@ start : num 68615536 ..@ end : num 68620354 ..@ strand : num 1 ..@ type : chr(0) ..@ data :List of 8 .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:78] 0 1 47 48 49 54 500 503 504 505 ... .@ lengths : int [1:78] 12 1 1 9 1 1 1 3 2 6 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:87] 0 2 61 786 787 788 790 793 795 796 ... .@ lengths : int [1:87] 12 1 12 1 1 1 1 1 2 3 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:72] 0 1 48 49 50 52 506 508 511 513 ... .@ lengths : int [1:72] 12 1 9 1 1 1 2 2 1 3 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:74] 0 60 62 63 65 738 741 743 746 748 ... .@ lengths : int [1:74] 13 1 7 3 1 1 2 2 4 2 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:61] 0 20 23 513 516 517 518 519 520 521 ... .@ lengths : int [1:61] 13 11 1 1 3 1 1 1 3 2 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:61] 0 19 21 437 438 440 441 442 444 445 ... .@ lengths : int [1:61] 13 11 1 1 2 1 1 3 7 5 ... .@ elementMetadata: NULL

.. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:64] 0 1 36 37 466 467 468 469 470 472 ... .@ lengths : int [1:64] 12 1 2 10 1 1 1 1 2 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() .. ..$:Formal class 'Rle' [package "IRanges"] with 4 slots .. ..@ values : int [1:84] 0 1 51 52 53 54 725 726 728 729 ... .@ lengths : int [1:84] 12 1 5 3 3 1 1 3 3 1 ... .@ elementMetadata: NULL .. ..@ metadata : list() ..@ assayData :<environment: 0x104fc118> ..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots .. .. ..@ name : chr "" .. .. ..@ lab : chr "" .. .. ..@ contact : chr "" .. .. ..@ title : chr "" .. .. ..@ abstract : chr "" .. .. ..@ url : chr "" .. .. ..@ pubMedIds : chr "" .. .. ..@ samples : list() .. .. ..@ hybridizations : list() .. .. ..@ normControls : list() .. .. ..@ preprocessing : list() .. .. ..@ other : list() .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 2 .. ..$: int [1:3] 1 0 0 .. ..$: int [1:3] 1 1 0 ..@ annotation : chr(0) ..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 0 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. .. ..@ .Data:List of 3 .. ..$: int [1:3] 3 0 1 .. ..$: int [1:3] 2 20 1 .. ..$: int [1:3] 1 3 0

Examples

```
data(ampliQueso)
```

`runAQReport` *runAQReport - runing the AmpliQueso report*

Description

This function creates AmpliQueso report in PDF format using either Article or Beamer LaTeX document classes. Report is created in current working directory. Please make sure that you have Beamer LaTeX package installed prior to running this report if you choose this output report type.

Usage

```
runAQReport(iCovdesc, iBedFile, iRefSeqFile, iGroup, iT1, iT2, iTopN = 5, iMinQual, iReportFormat = "pd
```

Arguments

iCovdesc	Covdesc-like data frame - BAM files are read from row names. Similar to covdesc in simpleaffy and rnaSeqMap - tab delimited table of BAM files and groupings. This is the table of experimental design.
iBedFile	BED file with same genomic coordinates as in BAMs (preferably standard hg19)
iRefSeqFile	Reference sequence in FASTA format used for SNPs detection. If set to NULL (by default) will skip SNP report.
iGroup	Name of the attribute/group in the experimental description (i.e. covdesc)
iT1	name of the first group from the experiment (it should be consistent with the groupings specified in covdesc file)
iT2	name of the other group from the experiment (it should be consistent with the groupings specified in covdesc file)
iTopN	number specifing top n expressed genes to be reported
iMinQual	number specifing the minimum quality of the SNPs returned by samtools used to filter out these of poor quality. Please refer to Samtools documentation for further details on the definition of this measure.
iReportFormat	Report output format. Currently PDF only.
iReportType	LaTeX document class to be used for report generation.
iReportPath	Output path for the report. It is created in the current working directory by default.
iVerbose	Boolean specifing whether R code should be inlcude in the report, if set to FALSE only figures and tables will be included.
iVCF	TBD
iParallel	Boolean specifing whether R code should be run in parallel – it is turned on by default..

Value

No vaue returned.

Author(s)

Alicja Szabelska, Marek Wiewiorka, Michal Okoniewski

Examples

```
#runAQReport(iCovdesc="covdesc",iBedFile="test.BED",iRefSeqFile=NULL,iGroup="group",iT1="sick",iT2="healthy",i
```

Index