Package 'exomePeak2'

November 12, 2019

Type Package

Title Peak calling and quantification for MeRIP-seq data sets.

Version 1.0.0

Author Zhen Wei

Maintainer Zhen Wei <zhen.wei10@icloud.com>

Description This package can conduct peak calling and quantification for MeRIP-seq data sets while considering the biological variabilities and the GC-content bias corrections.

License GPL (>= 2)

Imports Rsamtools,GenomicAlignments,GenomicRanges,GenomicFeatures,DESeq2,ggplot2,mclust,genefilter,BSgenome,Bio

Depends SummarizedExperiment,cqn

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

`call_peaks_with_GLM` *Statistical Inference with DESeq package based on the provided reads count for exomic bins.*

Description

`call_peaks_with_GLM` conduct inference on every exome bins using negative binomial model, the significant bins will be the merged into peaks.

Usage

```
call_peaks_with_GLM(SE_bins, glm_type = c("Poisson", "NB", "DESeq2"),
  correct_GC_bg = TRUE, qtnorm = TRUE, txdb, count_cutoff = 5,
  p_cutoff = NULL, p_adj_cutoff = 0.05, log2FC_cutoff = 0,
  consistent_peak = TRUE, consistent_log2FC_cutoff = 0,
  consistent_fdr_cutoff = 0.05, alpha = 0.05, p0 = 0.8)
```

Arguments

`SE_bins`	a `SummarizedExperiment` object. The meta-data collumn should contain the design information of IP/input + treated/control.
`glm_type`	a character, which can be one of the "Poisson", "NB", and "DESeq2". This argument specify the type of generalized linear model used in peak calling; Default to be "Poisson". The DESeq2 method is only recommended for high power experiments with more than 3 biological replicates for both IP and input.

`correct_GC_bg` a `logical` value of whether to estimate the GC content linear effect on background regions; default = `TRUE`.

If `correct_GC_bg = TRUE`, it could result in a more accurate estimation of the technical effect of GC content for the RNA modifications that are highly biologically related to GC content.

`qtnorm` a `logical` of whether to perform subset quantile normalization after the GC content linear effect correction; default = `TRUE`.

Subset quantile normalization will be applied within the IP and input samples seperately to account for the inherent differences between the marginal distributions of IP and input samples.

`txdb` the txdb object that is necessary for the calculation of the merge of the peaks.

`count_cutoff` an integer value indicating the cutoff of the mean of reads count in a row, inference is only performed on the windows with the row average read count bigger than the cutoff. Default value is 5.

`p_cutoff` a numeric value of the p value cutoff used in DESeq inference.

`p_adj_cutoff` a numeric value of the adjusted p value cutoff used in DESeq2 inference; if provided, the value of `p_cutoff` will be ignored; Default = 0.05.

`log2FC_cutoff` a non negative numeric value of the log2 fold change (log2 IP/input) cutoff used in the inferene of peaks.

`consistent_peak`

a `logical` of whether the positive peaks returned should be consistent among replicates; default = `TRUE`.

`consistent_log2FC_cutoff`

a `numeric` for the modification log2 fold changes cutoff in the peak consisency calculation; default = 0.

`consistent_fdr_cutoff`

a `numeric` for the BH adjusted C-test p values cutoff in the peak consistency calculation; default = 0.05. Check `ctest`.

`alpha` a `numeric` for the binomial quantile used in the consitent peak filter; default = `0.05`.

`p0` a `numeric` for the binomial proportion parameter used in the consistent peak filter; default = `0.8`.

For a peak to be consistently methylated, the minimum number of significant enriched replicate pairs is defined as the 1 - alpha quantile of a binomial distribution with p = p0 and N = number of possible pairs between replicates.

The consistency defined in this way is equivalent to the rejection of an exact binomial test with null hypothesis of p < p0 and N = replicates number of IP * replicates number of input.

Details

`call_peaks_with_GLM` will performe exome level peak calling using DESeq2 model,

The significant bins will be merged into modification peaks.

The insignificant bins (pass the row means filtering) will also be merged into control peaks.

Value

This function will return a list of `GRangesList` object storing peaks for both modification and control.

`consDESeq2_DM` *Calculate the index for across sample differential modification consistency in a DESeqDataSet object*

Description

Calculate the index for across sample differential modification consistency in a DESeqDataSet object

Usage

```
consDESeq2_DM(dds, consistent_log2FC_cutoff = 1, p = 0.7,
  alpha = 0.05)
```

Arguments

`dds`	a DESeqDataSet object.
`consistent_log2FC_cutoff`	a numeric for the cutoff of differential modification log2 fold changes; default = 1.
`p`	the binomial parameter p used by the binomial test; default = 0.7.
`alpha`	the significant level of the binomial test; default = 0.05.

Details

The minimum consistent number cutcoff is defined by 1-alpha quantile of a binomial distribution with probability of success = p, and number of trials = number of possible pairs between replicates. This is equivalent to the rejection of an exact binomial test with null hypothesis of $p < 0.8$ and N = number of possible pairs.

Value

a logical index for the consistently modified rows in the DESeqDataSet.

`consDESeq2_M` *Calculate the index for across sample modification consistency in a DESeqDataSet object*

Description

Calculate the index for across sample modification consistency in a DESeqDataSet object

Usage

```
consDESeq2_M(dds, consistent_log2FC_cutoff = 1,
  consistent_fdr_cutoff = 0.05, p0 = 0.8, alpha = 0.05)
```

Arguments

dds	a DESeqDataSet object.
consistent_log2FC_cutoff	a numeric for the modification log2 fold changes cutoff in the peak consisency calculation; default = 1.
consistent_fdr_cutoff	a numeric for the BH adjusted C-test p values cutoff in the peak consistency calculation; default = 0.05. Check ctest.
p0	a numeric for the binomial proportion parameter used in the consistent peak filter; default = 0.8. For a peak to be consistently methylated, the minimum number of significant enriched replicate pairs is defined as the 1 - alpha quantile of a binomial distribution with p = p0 and N = number of possible pairs between replicates. The consistency defined in this way is equivalent to the rejection of an exact binomial test with null hypothesis of p < p0 and N = replicates number of IP * replicates number of input.
alpha	a numeric for the binomial quantile used in the consitent peak filter; default = 0.05.

Details

The minimum consistent number cutcoff is defined by 1-alpha quantile of a binomial distribution with probability of success = p, and number of trials = number of possible pairs between replicates. This is equivalent to the rejection of an exact binomial test with null hypothesis of p < 0.8 and N = number of possible pairs.

Value

a logical index for the consistently modified rows in the DESeqDataSet.

---

convertTxDb *Convert the txdb object into the full transcript and whole genome types*

Description

This function can convert the txdb object into full transcript and whole genome types.

Usage

```
convertTxDb(txdb, type = c("full_tx", "whole_genome"))
```

Arguments

txdb	a TxDb object containing the regular transcript annotation.
type	the type of TxDb object of the output, can be one in c("full_tx","whole_genome").

Value

a TxDb object that will change the exon region into the full transcript and the whole genome regions.

ctest *Exact poisson test on the ratio between IP and input counts.*

Description

Exact poisson test on the ratio between IP and input counts.

Usage

```
ctest(IP_count, input_count, IP_sizeFactor, input_sizeFactor, fold = 1,
  alpha = 0.05)
```

Arguments

IP_count	a numeric vector for the counts of IP sample.
input_count	a numeric vector for the counts of input sample.
IP_sizeFactor	a numeric vector for the size factors of IP sample.
input_sizeFactor	a numeric vector for the size factors of input sample.
fold	the fold change under the null hypothesis; default = 1.
alpha	the alpha level used for determining the optimal independent filtering in filtered_R; default = 0.05.

Details

C-tests will be conducted between each entries of the IP and input vector.

The following statistical model is assumed on the data:

IP ~ Poisson(mean_IP*IP_sizeFactor)

input ~ Poisson(mean_input*input_sizeFactor)

The one-sided statistical test (C-test) is conducted with the following hypothesis pair:

$$H_0 : mean(IP)/mean(input) <= fold$$

$$H_1 : mean(IP)/mean(input) > fold$$

The exact p-values will be generated using binomial test, check poisson.test for more details.

The p-values are adjusted with "BH" method with an independent filtering slected by function filtered_R in package genefilter;

the filter statistics used is the normalized read abundence.

Value

A list include adjusted p values (with method fdr) and the corresponding log2 fold changes.

`decision_deseq` *Calculate the decision table for a DESeq result.*

Description

`Decision_dsresult` is an internal function used to summary the cut-off and the number of positive results used for DESeq result..

Usage

```
decision_deseq(res, log2FC_cut, p_cut, padj_cut, min_mod)
```

Arguments

`res`	A `DESeqResults` or similar object that contains the result statistics for either modification or differential modification.
`log2FC_cut`	The log2 fold change cutoff of the inference result, default setting is 0.
`p_cut`	A numeric value between 0 to 1, indicating the p value cut off of the Wald test defined by DESeq, it will be neglected if `padj_cut` is not NULL.
`padj_cut`	A numeric value between 0 to 1, indicating the fdr cut off of the Wald test defined by DESeq.
`min_mod`	Minimum number of features returned, when this is smaller than the cut-off results, additional features are called by the order of p values.

Value

A `data.frame` object indicating the column and cut-off value used for desicion, also it includes the number of positive sites in both directions based on the decision.

`decision_deseq2` *Calculate the decision table for a DESeq2 result.*

Description

`Decision_dsresult` is an internal function used to summary the cut-off and the number of positive results used for DESeq2 result..

Usage

```
decision_deseq2(Inf_RES, log2FC_cut = 0, P_cut = 0.05,
  Padj_cut = NULL, Min_mod = 1000, Exp_dir = c("hyper", "hypo",
  "both"))
```

Arguments

Inf_RES	A DESeq2Results or similar object that contains the result statistics for either methylation or differential methylation.
log2FC_cut	The log2 fold change cutoff of the inference result, default setting is 0.
P_cut	A numeric value between 0 to 1, indicating the p value cut off of the Wald test defined by DESeq2 (or defined by QNB), it will be neglected if Padj_cut is not NULL.
Padj_cut	A numeric value between 0 to 1, indicating the fdr cut off of the Wald test defined by DESeq2 (or defined by QNB).
Min_mod	Minimum number of features returned, when this is smaller than the cut-off results, additional features are called by the order of p values.
Exp_dir	This parameter is filled when making decisions on differential methylation, it could be "hyper", "hypo", and "both".

Value

A data.frame object indicating the column and cut-off value used for desicion, also it includes the number of positive sites in both directions based on the decision.

DESeq2Results-methods *Method DESeq2Results*

Description

Method DESeq2Results

Method DESeq2Results<-

Accessor to the slot DESeq2Results in class SummarizedExomePeak.

Accessor to the slot DESeq2Results in class SummarizedExomePeak.

Usage

```
DESeq2Results(x1)

DESeq2Results(x2) <- value

## S4 method for signature 'SummarizedExomePeak'
DESeq2Results(x1)

## S4 replacement method for signature 'SummarizedExomePeak'
DESeq2Results(x2) <- value
```

Arguments

x1	A data.frame object.
x2	A SummarizedExomePeak object.
value	A data.frame object.

disj_background *Find the background of the user provided mod_gration.*

Description

Find the background of the user provided mod_gration.

Usage

```
disj_background(mod_gr, txdb, background_bins = NULL,
  background_types = c("Gaussian_mixture", "m6Aseq_prior", "manual",
  "all"), control_width = 50, rename_mod = FALSE)
```

Arguments

mod_gr	A GRanges object of user provided mod_gration (names are neccessary for the index of the splitting).
txdb	A TxDb object that define the transcript mod_gration.
background_bins	A Granges object for background binds.
background_types	A logical value, TRUE if the region of 5'UTR and long exons of the transcripts should be dropped in control region; Default TRUE.
control_width	A integer for the minimum width of the control region returned; default 50.
rename_mod	Whether to rename the returned modification sites, default = FALSE.

Value

A GRangesList object. The first portion is the exons regions that is not overlapped with annoation.

If the resulting ranges have less number and width compared with what defined in cut_off_num, the exon regions of txdb will be returned as the background.

The second portion is the restructed user provided mod_gration with gene id mod_grated.

---

estimateSeqDepth-methods *Method estimateSeqDepth*

Description

estimateSeqDepth estimate sequencing depth size factors for each MeRIP-seq samples. Under default setting, the sequencing depth are estimated by the robust estimator defined in package DESeq. i.e. the median of the ratios to the row geometric means.

Usage

```
estimateSeqDepth(sep, from = c("Background", "Modification", "All"), ...)

## S4 method for signature 'SummarizedExomePeak'
estimateSeqDepth(sep,
  from = c("Background", "Modification", "All"), ...)
```

Arguments

sep	a SummarizedExomePeak object.
from	a character specify the subset of features for sequencing depth estimation, can be one of c("Background", "Modification", "All").
	Background The sequencing depths are estimated from the background control regions. This method could make the IP/input LFC estimates become closer to the true modification proportion.
	Modification The sequencing depths are estimated from the modification peaks/sites.
	All The sequencing depths are estimated from both the background and the modification regions.
	Under default setting, the sequencing depth factors are estimated from the background Background regions.
...	inherited from estimateSizeFactorsForMatrix.

Details

The function takes the input of a SummarizedExomePeak object, and it estimates the sequencing depth size factors by the columns of its assay.

Value

This function will return a SummarizedExomePeak object containing newly estimated sequencing depth size factors.

See Also

normalizeGC

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Estimate the sequencing depth size factors
sep <- estimateSeqDepth(sep)
```

---

exomePeak2 *Peak Calling and Peak Statistics Quantification on MeRIP-seq Dataset.*

Description

exomePeak2 conducts peak calling and peak statistics calculation from **BAM** files of a MeRIP-seq experiment. The function integrates the following steps of a standard MeRIP-seq data analysis pipeline.

1. Check and index the BAM files with scanMeripBAM.
2. Call modification peaks on exons with exomePeakCalling.
3. Calculate offset factors of GC content biases with normalizeGC.
4. Calculate (differential) modification statistics with the generalized linear model (GLM) using glmM or glmDM
5. Export the peaks/sites statistics with user defined format by exportResults.

See the help pages of the corresponding functions for the complete documentation.

Usage

```
exomePeak2(bam_ip = NULL, bam_input = NULL, bam_treated_ip = NULL,
  bam_treated_input = NULL, txdb = NULL, bsgenome = NULL,
  genome = NA, gff_dir = NULL, mod_annot = NULL,
  paired_end = FALSE, library_type = c("unstranded", "1st_strand",
  "2nd_strand"), fragment_length = 100, binding_length = 25,
  step_length = binding_length, peak_width = fragment_length/2,
  pc_count_cutoff = 5, bg_count_cutoff = 50, p_cutoff = 1e-04,
  p_adj_cutoff = NULL, log2FC_cutoff = 1, consistent_peak = FALSE,
  consistent_log2FC_cutoff = 1, consistent_fdr_cutoff = 0.05,
  alpha = 0.05, p0 = 0.8, parallel = FALSE, background = c("all",
  "Gaussian_mixture", "m6Aseq_prior", "manual"),
  manual_background = NULL, correct_GC_bg = FALSE, qtnorm = FALSE,
  glm_type = c("DESeq2", "Poisson", "NB"), LFC_shrinkage = c("apeglm",
  "ashr", "Gaussian", "none"), export_results = TRUE,
  export_format = c("CSV", "BED", "RDS"), table_style = c("bed",
  "granges"), save_plot_GC = TRUE, save_plot_analysis = FALSE,
  save_plot_name = "", save_dir = "exomePeak2_output",
  peak_calling_mode = c("exon", "full_tx", "whole_genome"))
```

Arguments

bam_ip	a character vector for the BAM file directories of the (control) IP samples.
bam_input	a character vector for the BAM file directories of the (control) input samples.
bam_treated_ip	a character vector for the BAM file directories of the treated IP samples.
bam_treated_input	a character vector for the BAM file directories of the treated input samples. If the bam files do not contain treatment group, user should only fill the arguments of BAM_ip and BAM_input.
txdb	a TxDb object for the transcript annotation. If the TxDb object is not available, it could be a character string of the UCSC genome name which is acceptable by makeTxDbFromUCSC. For example: "hg19".
bsgenome	a BSgenome object for the genome sequence information. If the BSgenome object is not available, it could be a character string of the UCSC genome name which is acceptable by getBSgenome. For example: "hg19".

genome — a character string of the UCSC genome name which is acceptable by getBSgenome or/and makeTxDbFromUCSC. For example: "hg19".

By default, the argument = NA, it should be provided when the BSgenome or/and the TxDb object are not available.

gff_dir — optional, a character which specifies the directory toward a gene annotation GFF/GTF file, it is applied when the TxDb object is not available, default = NULL.

mod_annot — a GRanges object for user provided single based RNA modification annotation.

If user provides the single based RNA modification annotation, exomePeak2 will perform reads count and (differential) modification quantification on the provided annotation.

The single base annotation will be flanked by length = floor(fragment_length - binding_length/2) to account for the fragment length of the sequencing library.

paired_end — a logical of whether the data comes from the Paired-End Library, TRUE if the data is Paired-End sequencing; default FALSE.

library_type — a character specifying the protocal type of the RNA-seq library, can be one in c("unstranded", "1st_strand", "2nd_strand"); default = "unstranded".

unstranded The randomly primed RNA-seq library type, i.e. both the strands generated during the first and the second strand sythesis are sequenced; example: Standard Illumina.

1st_strand The first strand-specific RNA-seq library, only the strand generated during the first strand sythesis is sequenced; examples: dUTP, NSR, NNSR.

2nd_strand The second strand-specific RNA-seq library, only the strand generated during the second strand sythesis is sequenced; examples: Ligation, Standard SOLiD.

fragment_length — a positive integer number for the expected fragment length in nucleotides; default = 100.

binding_length — a positive integer number for the expected binding length of the anti-modification antibody in IP samples; default = 25.

step_length — a positive integer number for the shift distances of the sliding window; default = binding_length.

peak_width — a numeric value for the minimum width of the merged peaks; default = fragment_length .

pc_count_cutoff — a numeric value for the cutoff on average window's reads count in peak calling; default = 5.

bg_count_cutoff — a numeric value for the cutoff on average window's reads count in background identification; default = 50.

p_cutoff — a numeric value for the cutoff on p values in peak calling; default = NULL.

p_adj_cutoff — a numeric value for the cutoff on Benjamini Hochberg adjusted p values in peak calling; default = 0.05.

log2FC_cutoff — a numeric value for the cutoff on log2 IP over input fold changes in peak calling; default = 1.

consistent_peak — a logical of whether the positive peaks returned should be consistent among all the replicates; default = FALSE.

consistent_log2FC_cutoff
: a numeric for the modification log2 fold changes cutoff in the peak consisency calculation; default = 1.

consistent_fdr_cutoff
: a numeric for the BH adjusted C-test p values cutoff in the peak consistency calculation; default = 0.05. Check ctest.

alpha
: a numeric for the binomial quantile used in the consitent peak filter; default = 0.05.

p0
: a numeric for the binomial proportion parameter used in the consistent peak filter; default = 0.8.

  For a peak to be consistently methylated, the minimum number of significant enriched replicate pairs is defined as the 1 - alpha quantile of a binomial distribution with p = p0 and N = number of possible pairs between replicates.

  The consistency defined in this way is equivalent to the rejection of an exact binomial test with null hypothesis of p < p0 and N = replicates number of IP * replicates number of input.

parallel
: a logical value indicating whether to use parallel computation, it will require > 16GB memory if parallel = TRUE; default = FALSE.

background
: a character specifies the method for the background finding, i.e. to identify the windows without modification signal. It could be one of "Gaussian_mixture", "m6Aseq_prior", "manual", and "all"; default = "all".

  In order to accurately account for the technical variations, it is often important to estimate the sequencing depth and GC content linear effects on windows without modification signals.

  The following methods are supported in ExomePeak2 to differentiate the no modification background windows from the modification containig windows.

  all Use all windows as the background. This choise assumes no differences in the effects of technical features between the background and the modification regions.

  Gaussian_mixture The background is identified by Multivariate Gaussian Mixture Model (MGMM) with 2 mixing components on the vectors containing methylation level estimates and GC content, the background regions are predicted by the Bayes Classifier on the learned GMM.

  m6Aseq_prior The background is identified by the prior knowledge of m6A topology, the windows that are not overlapped with long exons (exon length >= 400bp) and 5'UTR are treated as the background windows.
  This type of background should not be used if the MeRIP-seq data is not targetting on m6A methylation.

  manual The background regions are defined by the user manually at the argument manual_background.

manual_background
: a GRanges object for the user provided unmodified background; default = NULL.

correct_GC_bg
: a logical value of whether to estimate the GC content linear effect on background regions; default = FALSE.

  If = TRUE, it could lead to a more accurate estimation of GC content bias for the RNA modifications that are highly biologically related to GC content.

qtnorm
: a logical of whether to perform subset quantile normalization after the GC content linear effect correction; default = FALSE.

  If qtnorm = TRUE, subset quantile normalization will be applied within the IP and input samples seperately to account for the inherent differences between the marginal distributions of IP and input samples.

glm_type — a character speciefies the type of Generalized Linear Model (GLM) fitted for the purpose of statistical inference during peak calling, which can be one of the c("DESeq2", "NB", "Poisson").

DESeq2 Fit the GLM defined in the function DESeq, which is the NB GLM with regulated estimation of the overdispersion parameters.

NB Fit the ordinary Negative Binomial (NB) GLM.

Poisson Fit the Poisson GLM.

By default, the DESeq2 GLMs are fitted on the data set with > 1 biological replicates for both the IP and input samples, the Poisson GLM will be fitted otherwise.

LFC_shrinkage — a character for the method of emperical bayes shrinkage on log2FC, could be one of c("apeglm", "ashr", "Gaussian", "none"); Default = "apeglm".

see lfcShrink for more details; if "none" is selected, only the MLE will be returned.

export_results — a logical of whether to save the results on disk; default = TRUE.

export_format — a character vector for the format(s) of the result being exported, could be the subset of c("CSV","BED","RDS"); Default = c("CSV","BED","RDS").

table_style — a character for the style of the table being exported, could be one of c("bed", "granges"); Default = "bed".

save_plot_GC — a logical of whether to generate the plots for GC content bias assessment; default = TRUE.

save_plot_analysis — a logical of whether to generate the plots for genomic analysis on modification sites; default = FALSE.

save_plot_name — a character for the name of the plots being saved; Default = "Plot".

save_dir — a character for the name of the directory being saved; Default = "exomePeak2_output".

peak_calling_mode — a character specifies the scope of peak calling on genome, can be one of c("exon", "full_transcript", "whole_genome"); Default = "exon".

exon generate sliding windows on exon regions.

full_transcript generate sliding windows on the full transcripts (include both introns and exons).

whole_genome generate sliding windows on the whole genome (include introns, exons, and the intergenic regions).

P.S. The full transcript mode and the whole genome mode demand big memory size (> 4GB) for large genomes.

Details

exomePeak2 call RNA modification peaks and calculate peak statistics from **BAM** files of a MeRIP-seq experiment.

The transcript annotation (from either the TxDb object or the **GFF** file) should be provided to perform analysis on exons.

The BSgenome object is also required to perform the GC content bias adjustment. If the bsgenome and the genome arguments are not provided (= NULL), the downstream analysis will proceed without GC content bias corrections.

If the **BAM** files in treated samples are provided at the arguments bam_treated_ip and bam_treated_input, the statistics of differential modification analysis on peaks/sites will be reported.

Under default setting, exomePeak2 will save the results of (differential) modification analysis under a folder named by 'exomePeak2_output'. The results generated include a **BED** file and a **CSV** table that store the locations and statistics of the (differential) modified peaks/sites.

Value

a SummarizedExomePeak object.

See Also

exomePeakCalling, glmM, glmDM, normalizeGC, exportResults, plotLfcGC

Examples

```
#Specify File Directories

GENE_ANNO_GTF = system.file("extdata", "example.gtf", package="exomePeak2")

f1 = system.file("extdata", "IP1.bam", package="exomePeak2")
f2 = system.file("extdata", "IP2.bam", package="exomePeak2")
f3 = system.file("extdata", "IP3.bam", package="exomePeak2")
f4 = system.file("extdata", "IP4.bam", package="exomePeak2")
IP_BAM = c(f1,f2,f3,f4)
f1 = system.file("extdata", "Input1.bam", package="exomePeak2")
f2 = system.file("extdata", "Input2.bam", package="exomePeak2")
f3 = system.file("extdata", "Input3.bam", package="exomePeak2")
INPUT_BAM = c(f1,f2,f3)

# Peak Calling

sep <- exomePeak2(bam_ip = IP_BAM,
                  bam_input = INPUT_BAM,
                  gff_dir = GENE_ANNO_GTF,
                  genome = "hg19",
                  paired_end = FALSE)
sep

# Differential Modification Analysis on Modification Peaks (Comparison of Two Conditions)

f1=system.file("extdata", "treated_IP1.bam", package="exomePeak2")
TREATED_IP_BAM=c(f1)
f1=system.file("extdata", "treated_Input1.bam", package="exomePeak2")
TREATED_INPUT_BAM=c(f1)

sep <- exomePeak2(bam_ip = IP_BAM,
                  bam_input = INPUT_BAM,
                  bam_treated_input = TREATED_INPUT_BAM,
                  bam_treated_ip = TREATED_IP_BAM,
                  gff_dir = GENE_ANNO_GTF,
                  genome = "hg19",
                  paired_end = FALSE)
sep

# Modification Level Quantification on Single Based Modification Annotation

f2 = system.file("extdata", "mod_annot.rds", package="exomePeak2")
```

```
MOD_ANNO_GRANGE <- readRDS(f2)

sep <- exomePeak2(bam_ip = IP_BAM,
                  bam_input = INPUT_BAM,
                  gff_dir = GENE_ANNO_GTF,
                  genome = "hg19",
                  paired_end = FALSE,
                  mod_annot = MOD_ANNO_GRANGE)
sep

# Differential Modification Analysis on Single Based Modification Annotation

sep <- exomePeak2(bam_ip = IP_BAM,
                  bam_input = INPUT_BAM,
                  bam_treated_input = TREATED_INPUT_BAM,
                  bam_treated_ip = TREATED_IP_BAM,
                  gff_dir = GENE_ANNO_GTF,
                  genome = "hg19",
                  paired_end = FALSE,
                  mod_annot = MOD_ANNO_GRANGE)
sep
```

---

exomePeakCalling-methods

Method exomePeakCalling

Description

exomePeakCalling call peaks of RNA modification from a MeRIP-seq data set.

Usage

```
exomePeakCalling(merip_bams = NULL, txdb = NULL, bsgenome = NULL,
  genome = NA, mod_annot = NULL, glm_type = c("DESeq2", "NB",
  "Poisson"), background = c("Gaussian_mixture", "m6Aseq_prior",
  "manual", "all"), manual_background = NULL, correct_GC_bg = FALSE,
  qtnorm = FALSE, gff_dir = NULL, fragment_length = 100,
  binding_length = 25, step_length = binding_length,
  peak_width = fragment_length/2, pc_count_cutoff = 5,
  bg_count_cutoff = 50, p_cutoff = 1e-04, p_adj_cutoff = NULL,
  log2FC_cutoff = 0, consistent_peak = FALSE,
  consistent_log2FC_cutoff = 0, consistent_fdr_cutoff = 0.05,
  alpha = 0.05, p0 = 0.8, parallel = FALSE, bp_param = NULL)

## S4 method for signature 'MeripBamFileList'
exomePeakCalling(merip_bams = NULL,
  txdb = NULL, bsgenome = NULL, genome = NA, mod_annot = NULL,
  glm_type = c("DESeq2", "NB", "Poisson"), background = c("all",
  "Gaussian_mixture", "m6Aseq_prior", "manual"),
  manual_background = NULL, correct_GC_bg = FALSE, qtnorm = FALSE,
```

```
gff_dir = NULL, fragment_length = 100, binding_length = 25,
step_length = binding_length, peak_width = fragment_length/2,
pc_count_cutoff = 5, bg_count_cutoff = 50, p_cutoff = 1e-04,
p_adj_cutoff = NULL, log2FC_cutoff = 1, consistent_peak = FALSE,
consistent_log2FC_cutoff = 1, consistent_fdr_cutoff = 0.05,
alpha = 0.05, p0 = 0.8, parallel = FALSE, bp_param = NULL)
```

Arguments

merip_bams	a MeripBamFileList object returned by scanMeripBAM.
txdb	a TxDb object for the transcript annotation. If the TxDb object is not available, it could be a character string of the UCSC genome name which is acceptable by makeTxDbFromUCSC. For example: "hg19".
bsgenome	a BSgenome object for the genome sequence information. If the BSgenome object is not available, it could be a character string of the UCSC genome name which is acceptable by getBSgenome. For example: "hg19".
genome	a character string of the UCSC genome name which is acceptable by getBSgenome or/and makeTxDbFromUCSC. For example: "hg19". By default, the argument = NA, it should be provided when the BSgenome or/and the TxDb object are not available.
mod_annot	a GRanges or GRangesList object for user provided single based RNA modification annotation, the widths of the ranged object should be all equal to 1. If user provides the single based RNA modification annotation, exomePeak2 will perform reads count and (differential) modification quantification on the provided annotation. The single base annotation will be flanked by length = floor(fragment_length - binding_length/2) to account for the fragment length of the sequencing library.
glm_type	a character speciefies the type of Generalized Linear Model (GLM) fitted for the purpose of statistical inference during peak calling, which can be one of the c("DESeq2", "NB", "Poisson"). DESeq2 Fit the GLM defined in function DESeq, which is the NB GLM with regulated estimation of the overdispersion parameters. NB Fit the Negative Binomial (NB) GLM. Poisson Fit the Poisson GLM. By default, the DESeq2 GLMs are fitted on the data set with > 1 biological replicates for both the IP and input samples, the Poisson GLM will be fitted otherwise.
background	a character specifies the method for the background finding, i.e. to identify the windows without modification signal. It could be one of the "Gaussian_mixture", "m6Aseq_prior", "manual", and "all"; default = "Gaussian_mixture". In order to accurately account for the technical variations, it is often neccessary to estimate the GC content linear effects on windows without modification signals (background). The following methods are supported in ExomePeak2 to differentiate the no modification background windows from the modification containig windows. Gaussian_mixture The background is identified by Multivariate Gaussian Mixture Model (MGMM) with 2 mixing components on the vectors containing methylation level estimates and GC content, the background regions are predicted by the Bayes Classifier on the learned GMM.

`m6Aseq_prior` The background is identified by the prior knowledge of m6A topology, the windows that are not overlapped with long exons (exon length >= 400bp) and 5'UTR are treated as the background windows.
This type of background should not be used if the MeRIP-seq data is not using anti-m6A antibody.

`manual` The background regions are defined by user manually at the argument `manual_background`.

`all` Use all windows as the background. This is equivalent to not differentiating background and signal. It can lead to biases during the sequencing depth and the GC content correction factors estimation.

`manual_background`
: a `GRanges` object for the user provided unmodified background; default = `NULL`.

`correct_GC_bg`
: a `logical` value of whether to estimate the GC content linear effect on background regions; default = `FALSE`.
If = `TRUE`, it could lead to a more accurate estimation of GC content bias for the RNA modifications that are highly biologically related to GC content.

`qtnorm`
: a `logical` of whether to perform subset quantile normalization after the GC content linear effect correction; default = `FASLE`.
If `qtnorm = TRUE`, subset quantile normalization will be applied within the IP and input samples seperately to account for the inherent differences between the marginal distributions of IP and input samples.

`gff_dir`
: optional, a `character` which specifies the directory toward a gene annotation GFF/GTF file, it is applied when the `TxDb` object is not available, default = `NULL`.

`fragment_length`
: a positive integer number for the expected fragment length in nucleotides; default = `100`.

`binding_length`
: a positive integer number for the expected binding length of the anti-modification antibody in IP samples; default = `25`.

`step_length`
: a positive integer number for the shift distances of the sliding window; default = `binding_length`.

`peak_width`
: a `numeric` value for the minimum width of the merged peaks; default = `fragment_length`.

`pc_count_cutoff`
: a `numeric` value for the cutoff on average window's reads count in peak calling; default = `5`.

`bg_count_cutoff`
: a `numeric` value for the cutoff on average window's reads count in background identification; default = `50`.

`p_cutoff`
: a `numeric` value for the cutoff on p values in peak calling; default = `0.0001`.

`p_adj_cutoff`
: a `numeric` value for the cutoff on Benjamini Hochberg adjusted p values in peak calling; default = `NULL`.

`log2FC_cutoff`
: a `numeric` value for the cutoff on log2 IP over input fold changes in peak calling; default = `1`.

`consistent_peak`
: a `logical` of whether the positive peaks returned should be consistent among all the replicates; default = `FALSE`.

`consistent_log2FC_cutoff`
: a `numeric` for the modification log2 fold changes cutoff in the peak consisency calculation; default = 1.

`consistent_fdr_cutoff`
: a `numeric` for the BH adjusted C-test p values cutoff in the peak consistency calculation; default = 0.05. Check `ctest`.

`alpha`
: a `numeric` for the binomial quantile used in the consistent peak filter; default = `0.05`.

`p0`
: a `numeric` for the binomial proportion parameter used in the consistent peak filter; default = `0.8`.

  For a peak to be consistently methylated, the minimum number of significant enriched replicate pairs is defined as the 1 - alpha quantile of a binomial distribution with p = p0 and N = number of possible pairs between replicates.

  The consistency defined in this way is equivalent to the rejection of an exact binomial test with null hypothesis of p < p0 and N = replicates number of IP * replicates number of input.

`parallel`
: a `logical` value of whether to use parallel computation, typlically it requires more than 16GB of RAM if `parallel = TRUE`; default = `FALSE`.

`bp_param`
: optional, a `BiocParallelParam` object that stores the configuration parameters for the parallel execution.

Details

`exomePeakCalling` perform peak calling from the MeRIP-seq BAM files on exon regions defined by the user provided transcript annotations. If the `BSgenome` object is provided, the peak calling will be conducted with the GC content bias correction.

Under the default setting, for each window, exomePeak2 will fit a GLM of Negative Binomial (NB) with regulated estimation of the overdispersion parameters developed in `DESeq`. Wald tests with H0 of IP/input Log2 Fold Change (LFC) <= 0 are performed on each of the sliding windows. The significantly modified peaks are selected using the cutoff of p value < 0.0001.

Value

a `SummarizedExomePeak` object.

See Also

`exomePeak2`, `glmM`, `glmDM`, `normalizeGC`, `exportResults`

Examples

```
### Define File Directories

GENE_ANNO_GTF = system.file("extdata", "example.gtf", package="exomePeak2")

f1 = system.file("extdata", "IP1.bam", package="exomePeak2")
f2 = system.file("extdata", "IP2.bam", package="exomePeak2")
f3 = system.file("extdata", "IP3.bam", package="exomePeak2")
f4 = system.file("extdata", "IP4.bam", package="exomePeak2")
IP_BAM = c(f1,f2,f3,f4)
f1 = system.file("extdata", "Input1.bam", package="exomePeak2")
f2 = system.file("extdata", "Input2.bam", package="exomePeak2")
f3 = system.file("extdata", "Input3.bam", package="exomePeak2")
INPUT_BAM = c(f1,f2,f3)

f1 = system.file("extdata", "treated_IP1.bam", package="exomePeak2")
```

```
TREATED_IP_BAM = c(f1)
f1 = system.file("extdata", "treated_Input1.bam", package="exomePeak2")
TREATED_INPUT_BAM = c(f1)

### Peak Calling

MeRIP_Seq_Alignment <- scanMeripBAM(
                         bam_ip = IP_BAM,
                         bam_input = INPUT_BAM,
                         paired_end = FALSE
                        )

sep <- exomePeakCalling(
            merip_bams = MeRIP_Seq_Alignment,
            gff_dir = GENE_ANNO_GTF,
            genome = "hg19"
            )

sep <- normalizeGC(sep)

sep <- glmM(sep)

exportResults(sep)

### Differential Modification Analysis (Comparison of Two Conditions)

MeRIP_Seq_Alignment <- scanMeripBAM(
                         bam_ip = IP_BAM,
                         bam_input = INPUT_BAM,
                         bam_treated_ip = TREATED_IP_BAM,
                         bam_treated_input = TREATED_INPUT_BAM,
                         paired_end = FALSE
                        )

sep <- exomePeakCalling(
            merip_bams = MeRIP_Seq_Alignment,
            gff_dir = GENE_ANNO_GTF,
            genome = "hg19"
            )

sep <- normalizeGC(sep)

sep <- glmDM(sep)

exportResults(sep)
```

`exome_bins_from_txdb` *extract exome bins for peak calling given a txdb object*

Description

extract exome bins for peak calling given a txdb object

Usage

```
exome_bins_from_txdb(txdb, window_size = 25, step_size = 25)
```

Arguments

txdb	A txdb object.
window_size	An integer valued number of the width of the sliding windows or bins.
step_size	An integer valued number of the width of the bin steps.

Value

A GRanges object of exonic bins with the names corresponding to the indexes of bins. A metadata colomn named gene_id is attached to indicate its gene ID, which is provided by the txdb object. The gene IDs are divided into multiple ones if the gene contains exons that belong to different chromosomes and strands.

---

exonPlot *plot the distribution for the length of the overlapped exons.*

Description

plot the distribution for the length of the overlapped exons.

Usage

```
exonPlot(gfeatures, txdb, save_pdf_prefix = NULL, save_dir = ".")
```

Arguments

gfeatures	a list of GRanges or GRangesList.
txdb	a txdb object.
save_pdf_prefix	provided to save a pdf file.
save_dir	a character indicating the directory to save the plot; default ".".

---

exons_by_unique_gene *Extracting exons by the corresponding genes that are on the same chromosomes and strand.*

Description

Extracting exons by the corresponding genes that are on the same chromosomes and strand.

Usage

```
exons_by_unique_gene(txdb)
```

Arguments

txdb	A TXDB object.

Value

A GRangesList object, each element in it corresponds to GRanges of the reduced exons of an unique gene, the name corresponds to the original gene with .integer indexed if they have exons on different strands and chromosomes.

---

exportResults-methods *Method exportResults*

Description

Method exportResults

Export the (Differential) Modification Peaks/Sites and their associated Statistics

Usage

```
exportResults(sep, format = c("CSV", "BED", "RDS"),
  table_style = c("bed", "granges"), save_dir = "exomepeaks_result",
  cut_off_pvalue = NULL, cut_off_padj = 0.1, cut_off_log2FC = 0,
  min_num_of_positive = 30, expected_direction = c("both", "hyper",
  "hypo"), inhibit_filter = FALSE, reads_count = TRUE,
  GC_sizeFactors = TRUE)

## S4 method for signature 'SummarizedExomePeak'
exportResults(sep, format = c("CSV",
  "BED", "RDS"), table_style = c("bed", "granges"),
  save_dir = "exomePeak2_output", cut_off_pvalue = NULL,
  cut_off_padj = 0.1, cut_off_log2FC = 0, min_num_of_positive = 100,
  expected_direction = c("both", "hyper", "hypo"),
  inhibit_filter = FALSE, reads_count = TRUE, GC_sizeFactors = TRUE)
```

Arguments

sep	a SummarizedExomePeak object.
format	a character for the exported format, could be a vector that contains c("CSV", "BED", "RDS").
	CSV export a comma separated values (CSV) table with the genomic location and LFC statistics.
	BED export a BEDGraph file with the score column = -log2(adjusted p value).
	RDS export the RDS file of the SummarizedExperiment object.
table_style	a character for the style of the CSV table being exported, could be one in c("bed","granges").
	bed the genomic locations in the table are represented by BEDgraph style.
	granges the genomic locations in the table are represented by GRanges style.
save_dir	a character for the name of the directory being saved; Default = "exomePeak2_output".
cut_off_pvalue	a numeric value for the p value cutoff in the exported result; Default = NULL.

cut_off_padj a numeric value for the adjusted p value cutoff in the exported result; Default = 0.05.

cut_off_log2FC a numeric value for the log2 fold change cutoff of the exported result, only the sites with abs(LFC) larger than this value are kept; Default = 0.

min_num_of_positive
a numeric value for the minimum number of reported sites. If the number of remaining sites is less than this number after filtering, additional sites will be reported by the increasing order of the p value to meet this number.

expected_direction
a character for the expected differential modification direction, could be one in c("hyper", "hypo", "both").

hyper only report the peaks/sites with interactive LFC > 0.

hypo only report the peaks/sites with interactive LFC < 0.

both report the peaks/sites in both directions.

This argument is useful when the treated group involves the perturbation of a known writer or eraser protein; Default "both".

inhibit_filter a logical of whether to remove all the filters, this option is useful when quantification on single based site annotation; Default = FALSE.

reads_count a logical of whether to export the reads count for each sample; Default = TRUE.

GC_sizeFactors a logical of whether to export the GC content correction size factors; Default = TRUE.

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Save the modification peaks/sites statistics on the current directory.
exportResults(sep)
```

---

flank_on_exons *Flank ranges on exon coordinate.*

Description

This function provides extention of the ends of the GRangesList on transcript coordinate or exons.

Usage

```
flank_on_exons(grl, flank_length, txdb, index_flank = TRUE)
```

Arguments

grl a GRangesList object which is the target of the flanking.

flank_length the length of the flanking regions (on both left and right).

txdb the TxDb object for the transcript annotation.

index_flank whether to store the flanking regions seperately; default TRUE.

Value

A GRanges object with the extented ends on exons. The flanking regions and the binding regions are indexed by the names of the returned GRangesList. The original input GRangesList will be dropped if it is not mapped to the exonic regions.

GCsizeFactors-methods *Method GCsizeFactors*

Description

Method GCsizeFactors

Method GCsizeFactors<-

Accessor to the slot `GCsizeFactors` in class `SummarizedExomePeak`.

Accessor to the slot `GCsizeFactors` in class `SummarizedExomePeak`.

Usage

```
GCsizeFactors(x1)

GCsizeFactors(x2) <- value

## S4 method for signature 'SummarizedExomePeak'
GCsizeFactors(x1)

## S4 replacement method for signature 'SummarizedExomePeak'
GCsizeFactors(x2) <- value
```

Arguments

`x1`	A `SummarizedExomePeak` object.
`x2`	A `SummarizedExomePeak` object.
`value`	A `matrix` object.

GC_content_over_grl *Retrieve GC content level from genome over regions defined by a GRanges or GRangesList object.*

Description

The GC content is calculated only on exon sequences, for regions that failed to mapped to exons in txdb, the function would return NA instead. The ranges in grl will be flanked by the size fragment_length - binding_length before GC content calculation.

Usage

```
GC_content_over_grl(bsgenome, txdb, grl, fragment_length = 100,
  binding_length = 25, effective_GC = FALSE)
```

Arguments

bsgenome	A BSgenome object.
txdb	A TxDb object.
grl	A GRangesList object.
fragment_length	A positive integer of the expected fragment length in the RNA-Seq library; Default 100.
binding_length	A positive integer of the expected antibody binding length of IP; Default 25.
effective_GC	If TRUE, the GC content calculation will be weighted by the fragment mapping probabilities, currently it is only supported for the single based modification annotation; Default FALSE.

Value

a DataFrame contains 2 columns:

1. GC_content: the (effective) GC content of each GRanges / GRangesList element 2. Indx_length: the calculated total feature length used to quantify reads.

---

glmDM-methods *Method glmDM*

Description

glmDM perform inference and estimation on RNA differential modification log2FC.

GLMs with interactive design between dummy variables of IP/input and Treatment/control are fitted for each peaks/sites:

$$log2(Q) = intercept + I(Treatment) + I(IP) + I(IP) * I(Treatment)$$

The log2FC and the associated statistics are based on the coefficient estimate of the interactive term: $I(IP) * I(Treated)$.

Under default setting, the returned log2FC are the RR estimates with Couchey priors defined in apeglm.

Usage

```
glmDM(sep, glm_type = c("auto", "Poisson", "NB", "DESeq2"),
  LFC_shrinkage = c("apeglm", "ashr"), ...)

## S4 method for signature 'SummarizedExomePeak'
glmDM(sep, glm_type = c("auto",
  "Poisson", "NB", "DESeq2"), LFC_shrinkage = c("apeglm", "ashr",
  "none"), ...)
```

Arguments

sep	a SummarizedExomePeak object.
glm_type	a character speciefies the type of Generalized Linear Model (GLM) fitted for the purpose of statistical inference during peak calling, which can be one of the c("DESeq2", "NB", "Poisson").
	DESeq2 Fit the GLM defined in the function DESeq, which is the NB GLM with regulated estimation of the overdispersion parameters.
	NB Fit the Negative Binomial (NB) GLM.
	Poisson Fit the Poisson GLM.
	By default, the DESeq2 GLMs are fitted on the data set with > 1 biological replicates for both the IP and input samples, the Poisson GLM will be fitted otherwise.
LFC_shrinkage	a character for the method of emperical bayes shrinkage on log2FC, could be one of c("apeglm", "ashr", "none"); Default = "apeglm".
	see lfcShrink for details; if "none" is selected, only the MLE will be returned.
...	Optional arguments passed to DESeq

See Also

glmM

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_dm.rds", package="exomePeak2")

sep <- readRDS(f1)

### Normalize the GC contents biases
sep <- normalizeGC(sep)

### Calculate GLM Statistics on the Modification Peaks
sep <- glmDM(sep)
```

---

glmM-methods *Method glmM*

Description

glmM performs inference and estimation on IP/input log2FC.

GLMs with the design of an indicator of IP samples are fitted for each peaks/sites:

$$log2(Q) = intercept + I(IP)$$

The log2FC and the associated statistics are based on the coefficient estimate of the dummy variable term: $I(IP)$.

Under default setting, the returned log2FC are the RR estimates with Couchey priors defined in apeglm.

Usage

```
glmM(sep, glm_type = c("auto", "Poisson", "NB", "DESeq2"),
  LFC_shrinkage = c("apeglm", "Gaussian", "ashr"), ...)

## S4 method for signature 'SummarizedExomePeak'
glmM(sep, glm_type = c("auto", "Poisson",
  "NB", "DESeq2"), LFC_shrinkage = c("apeglm", "Gaussian", "ashr"), ...)
```

Arguments

sep	a summarizedExomePeak object.
glm_type	a character speciefies the type of Generalized Linear Model (GLM) fitted for the purpose of statistical inference during peak calling, which can be one of the c("DESeq2", "NB", "Poisson").
	DESeq2 Fit the GLM defined in the function DESeq, which is the NB GLM with regulated estimation of the overdispersion parameters.
	NB Fit the Negative Binomial (NB) GLM.
	Poisson Fit the Poisson GLM.
	By default, the DESeq2 GLMs are fitted on the data set with > 1 biological replicates for both the IP and input samples, the Poisson GLM will be fitted otherwise.
LFC_shrinkage	a character for the method of emperical bayes shrinkage on log2FC, could be one of c("apeglm", "Gaussian", "ashr", "none"); Default = "apeglm".
	see lfcShrink for details; if "none" is selected, only the MLE will be returned.
...	Optional arguments passed to DESeq

See Also

glmDM

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Select only the control group to avoid warning.
sep <- sep[,!colData(sep)$design_Treatment]

### Normalize the GC contents biases
sep <- normalizeGC(sep)

### Calculate GLM Statistics on the Modification Peaks
sep <- glmM(sep)
```

GLM_inference *Statistical Inference with DESeq2 on IP over input fold change.*

Description

GLM_inference conduct inference on log2 fold changes of IP over input using the GLM defined in DESeq2.

Usage

```
GLM_inference(SE_bins, glm_type = c("Poisson", "NB", "DESeq2"),
  p_cutoff = 1e-04, p_adj_cutoff = NULL, count_cutoff = 5,
  log2FC_mod = 1, min_mod_number = floor(sum(rowSums(assay(SE_bins)) >
  0) * 0.001), correct_GC_bg = FALSE, qtnorm = TRUE,
  consistent_peak = FALSE, consistent_log2FC_cutoff = 1,
  consistent_fdr_cutoff = 0.05, alpha = 0.05, p0 = 0.8)
```

Arguments

SE_bins	a SummarizedExperiment of read count. It should contains a colData with column named design_IP, which is a character vector with values of "IP" and "input". The column helps to index the design of MeRIP-Seq experiment.
glm_type	a character, which can be one of the "Poisson", "NB", and "DESeq2". This argument specify the type of generalized linear model used in peak calling; Default to be "Poisson". The DESeq2 method is only recommended for high power experiments with more than 3 biological replicates for both IP and input.
p_cutoff	a numeric for the p value cutoff used in DESeq inference.
p_adj_cutoff	a numeric for the adjusted p value cutoff used in DESeq2 inference; if provided, values in p_cutoff will be ignored.
count_cutoff	an integer indicating the cutoff of the mean of reads count in a row, inference is only performed on the windows with read count bigger than the cutoff. Default value is 10.
log2FC_mod	a non negative numeric for the log2 fold change cutoff used in DESeq inferene for modification containing peaks (IP > input).
min_mod_number	a non negative numeric for the minimum number of the reported modification containing bins. If the bins are filtered less than this number by the p values or effect sizes, more sites will be reported by the order of the p value until it reaches this number; Default to be floor(sum(rowSums(assay(SE_bins)) > 0)*0.001).
correct_GC_bg	a logical of whether to estimate the GC content linear effect on background regions; default = FALSE. If correct_GC_bg = TRUE, it may result in a more accurate estimation of the technical effect of GC content for the RNA modifications that are highly biologically related to GC content.
qtnorm	a logical of whether to perform subset quantile normalization after the GC content linear effect correction; default = TRUE. Subset quantile normalization will be applied within the IP and input samples seperately to account for the inherent differences between the marginal distributions of IP and input samples.

consistent_peak
: a logical of whether the positive peaks returned should be consistent among replicates; default = TRUE.

consistent_log2FC_cutoff
: a numeric for the modification log2 fold changes cutoff in the peak consisency calculation; default = 1.

consistent_fdr_cutoff
: a numeric for the BH adjusted C-test p values cutoff in the peak consistency calculation; default = 0.05. Check ctest.

alpha
: a numeric for the binomial quantile used in the consitent peak filter; default = 0.05.

p0
: a numeric for the binomial proportion parameter used in the consistent peak filter; default = 0.8.

  For a peak to be consistently methylated, the minimum number of significant enriched replicate pairs is defined as the 1 - alpha quantile of a binomial distribution with p = p0 and N = number of possible pairs between replicates.

  The consistency defined in this way is equivalent to the rejection of an exact binomial test with null hypothesis of p < p0 and N = replicates number of IP * replicates number of input.

Value

a list of the index for the significant modified peaks (IP > input) and control peaks (peaks other than modification containing peaks).

LibraryType-methods *Method LibraryType*

Description

Method LibraryType

Accessor to the slot LibraryType in class MeripBamFileList.

Usage

```
LibraryType(x)

## S4 method for signature 'MeripBamFileList'
LibraryType(x)
```

Arguments

x
: a MeripBamFileList object.

`mclust_bg` *Find background of merip-seq signal with model based clustering.*

Description

The function generate background index on the rowData of the summarizedExperiment.

Usage

```
mclust_bg(se_peak_counts, alpha = 1)
```

Arguments

`se_peak_counts`	A SummarizedExperiment object.
`alpha`	The small offset added when calculating the M value.

Details

This function will do the following jobs:

1. Filter the rows (modification sites) by average count.

2. Fit multivariate Gaussian mixture model with 2 mixing component to differentiate background and biological signal.

- depend on whether the bsgenome is provided, the values GC content will be used as one of the dimention of the clustering.

3. Classify the background and signal using bayes classifier by the learned model.

4. Return the index for the bins that are classified into background.

Value

A dummy variable of the background index.

`MeripBamFileList-class` *MeripBamFileList*

Description

An object that summarizes the BAM files used in a MeRIP-Seq experiment.

normalizeGC-methods *Method normalizeGC*

Description

normalizeGC estimates the feature specific size factors in order to reduce the technical variation during modification peak statistics quantification.

Usage

```
normalizeGC(sep, bsgenome = "hg19", txdb = "hg19", gff_dir = NULL,
  fragment_length = 100, binding_length = 25, feature = c("All",
  "Modification", "Background"), qtnorm = FALSE, effective_GC = FALSE)

## S4 method for signature 'SummarizedExomePeak'
normalizeGC(sep, bsgenome = NULL,
  txdb = NULL, gff_dir = NULL, fragment_length = 100,
  binding_length = 25, feature = c("All", "Modification",
  "Background"), qtnorm = FALSE, effective_GC = FALSE)
```

Arguments

sep	a SummarizedExomePeak object returned by exomePeak2 or exomePeakCalling.
bsgenome	a BSgenome object for the genome reference, If the BSgenome object is not available, it could be a character string of the UCSC genome name which is acceptable by getBSgenome, example: "hg19".
txdb	a TxDb object for the transcript annotation, If the TxDb object is not available, it could be a character string of the UCSC genome name which is acceptable by makeTxDbFromUCSC, example: "hg19".
gff_dir	optional, a character which specifies the directory toward a gene annotation GFF/GTF file, it is applied when the TxDb object is not available, default = NULL.
fragment_length	a positive integer number for the expected fragment length in nucleotides; default = 100.
binding_length	a positive integer number for the expected binding length of the anti-modification antibody in IP samples; default = 25.
feature	a character specifies the region used in the GC content linear effect estimation, can be one in c("All","Modification","Background"); default is "All". All The GC content linear effects will be estimated on all regions, i.e. both the region of modification and the background control regions. Modification The GC content linear effects will be estimated on the modification peaks/sites. Background The GC content linear effects will be estimated on the background control regions.
qtnorm	a logical of whether to perform subset quantile normalization after the GC content linear effect correction; default = FALSE. If qtnorm = TRUE, subset quantile normalization will be applied within the IP and input samples seperately to account for the inherent differences between the marginal distributions of IP and input samples.

effective_GC a logical of whether to calculate the effective GC content weighted by the fragment alignment probabilities; default = FALSE.

Details

PCR amplication bias related to GC content is a major source of technical variation in RNA-seq. The GC content biases are usually correlated within the same laboratory environment, and this will result in the batch effect between different studies.

The GC content normalization can result in an improvement of peak accuracy for most published m6A-seq data, and it is particullarly recommended if you want to compare the quantifications on methylation levels between different laboratory conditions.

Value

a SummarizedExomePeak object with the updated slot GCsizeFactors.

See Also

estimateSeqDepth

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Normalize the GC content biases
sep <- normalizeGC(sep)
```

Parameter-methods *Method Parameter*

Description

Method Parameter

Accessor to the slot Parameter in class MeripBamFileList.

Usage

```
Parameter(x)

## S4 method for signature 'MeripBamFileList'
Parameter(x)
```

Arguments

x a MeripBamFileList object.

plotExonLength-methods
Method plotExonLength

Description

This function plot the distribution of the exon length for peaks containing exons.

Usage

```
plotExonLength(sep, txdb = NULL, save_pdf_prefix = NULL,
  include_control_regions = TRUE, save_dir = ".")

## S4 method for signature 'SummarizedExomePeak'
plotExonLength(sep, txdb = NULL,
  save_pdf_prefix = NULL, include_control_regions = TRUE,
  save_dir = ".")
```

Arguments

sep	a SummarizedExomePeak object.
txdb	a TxDb object containing the transcript annotation.
save_pdf_prefix	a character if provided, a pdf file with the given name will be saved under the current working directory.
include_control_regions	a logical for whether to include the control regions or not; Default = TRUE.
save_dir	a character for the directory to save the plot; Default = ".".

Details

If the SummarizedExomePeaks object contains LFC statistics, the significantly modified peaks with IP to input log2FC > 0 and DESeq2 Wald test padj < .05 will be plotted .

If the SummarizedExomePeaks object contains interactive LFC statistics, both the hyper modification and hypo modification peaks with DESeq2 Wald test p values < .05 will be plotted.

Value

a ggplot object

Examples

```
### Make TxDb object from the gff file
library(GenomicFeatures)
GENE_ANNO_GTF = system.file("extdata", "example.gtf", package="exomePeak2")

txdb <- makeTxDbFromGFF(GENE_ANNO_GTF)

### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")
```

```
sep <- readRDS(f1)

### Visualize the linear relationships between GC content and normalized reads count under different regions
plotExonLength(sep,txdb)
```

---

plotLfcGC-methods *Method plotLfcGC*

Description

plotLfcGC plot the scatter plot between GC content and the (differential) modification LFCs.

Usage

```
plotLfcGC(sep, bsgenome = NULL, txdb = NULL, save_pdf_prefix = NULL,
  fragment_length = 100, binding_length = 25, effective_GC = FALSE,
  save_dir = ".")

## S4 method for signature 'SummarizedExomePeak'
plotLfcGC(sep, bsgenome = NULL,
  txdb = NULL, save_pdf_prefix = NULL, fragment_length = 100,
  binding_length = 25, effective_GC = FALSE, save_dir = ".")
```

Arguments

sep	a SummarizedExomePeak object.
bsgenome	a BSgenome object for the genome sequence, it could be the name of the reference genome recognized by getBSgenome.
txdb	a TxDb object for the transcript annotation, it could be the name of the reference genome recognized by makeTxDbFromUCSC.
save_pdf_prefix	a character, if provided, a pdf file with the given name will be saved under the current directory; Default = NULL.
fragment_length	a numeric value for the expected fragment length in the RNA-seq library; Default = 100.
binding_length	a numeric value for the expected antibody binding length in IP samples; Default = 25.
effective_GC	a logical value of whether to calculate the weighted GC content by the probability of reads alignment; default = FALSE.
save_dir	a character for the directory to save the plot; default ".".

Details

By default, this function will generate a scatter plot between GC content and the log2FC value. The significant modification sites will be lebeled in different colours.

Value

a ggplot object.

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Visualize the relationship between GC content and the (differential) LFC
plotLfcGC(sep)
```

---

plotReadsGC-methods *Method plotReadsGC*

Description

plotReadsGC visualizes the local regression curves between the normalized reads abundance and the local GC content.

Usage

```
plotReadsGC(sep, bsgenome = NULL, txdb = NULL,
  save_pdf_prefix = NULL, fragment_length = 100, binding_length = 25,
  effective_GC = FALSE, pool_replicates = FALSE, save_dir = ".")

## S4 method for signature 'SummarizedExomePeak'
plotReadsGC(sep, bsgenome = NULL,
  txdb = NULL, save_pdf_prefix = NULL, fragment_length = 100,
  binding_length = 25, effective_GC = FALSE, pool_replicates = FALSE,
  save_dir = ".")
```

Arguments

sep	a SummarizedExomePeak object.
bsgenome	a BSgenome object for the genome sequence, it could be the name of the reference genome recognized by getBSgenome.
txdb	a TxDb object for the transcript annotation, it could be the name of the reference genome recognized by makeTxDbFromUCSC.
save_pdf_prefix	a character, if provided, a pdf file with the given name will be saved under the current directory; Default = NULL.
fragment_length	a numeric value for the expected fragment length in the RNA-seq library; Default = 100.
binding_length	a numeric value for the expected antibody binding length in IP samples; Default = 25.

effective_GC — a logical value of whether to calculate the weighted GC content by the probability of reads alignment; default = FALSE.

pool_replicates — a logical value of whether to pool the replicates in the local regression fit; default = FALSE.

save_dir — a character for the directory to save the plot; default ".".

Details

The read abundances of both the control and the modification site regions are plotted, the read counts are normalized using the following method:

$$normalizedfeatureabundance = ((readcount/sizefactor)/regionlength) * 500$$

By default, it will use the sequencing depth size factor defined in the SummarizedExomePeak object, if the sequencing depth size factor is not found, new size factors will be estimated with the default method in estimateSeqDepth.

Value

a ggplot object

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Visualize the linear relationships between GC content and normalized reads count
plotReadsGC(sep)
```

---

plotSizeFactors-methods

Method plotSizeFactors

Description

Method plotSizeFactors

plot the size factors using different strategies.

Usage

```
plotSizeFactors(sep)

## S4 method for signature 'SummarizedExomePeak'
plotSizeFactors(sep)
```

Arguments

sep	a SummarizedExomePeak object.

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Visualize the size factors estimated using different regions
plotSizeFactors(sep)
```

---

quiet *Silencing unwanted function output.*

Description

Silencing unwanted function output.

Usage

```
quiet(x)
```

Arguments

x	any R expression.

---

reads_five_POS *extract reads five prime POS.*

Description

extract reads five prime POS.

Usage

```
reads_five_POS(reads, width = 1, fix = "start", ...)
```

Arguments

reads	a GAlignmentList object.
width	the width of the five prime POS.
fix	the end of the POS, start means 5', end means 3'.
...	arguments path to function resize.

reads_five_POS_rev *extract reads five prime POS and reverse its strand.*

Description

extract reads five prime POS and reverse its strand.

Usage

```
reads_five_POS_rev(reads, width = 1, fix = "start", ...)
```

Arguments

reads	a GAlignmentList object.
width	the width of the five prime POS.
fix	the end of the POS, start means 5', end means 3'.
...	arguments path to function resize.

reduce_peaks *A function to reduce ranges between elements within a GRangesList.*

Description

`reduce_inter_grl` redivide and reduce the overlapping GRangesList element.

Usage

```
reduce_peaks(peaks_grl, txdb)
```

Arguments

peaks_grl	The GRangesList of the peaks region to be reduced.
txdb	The txdb object used during the generation of the peaks.

Value

GRangesList object that is reduced between its inner elements. The metadata collumn is preserved, however the names is not tracked anymore.

remove_introns *Removing introns from a provided GRanges object.*

Description

Removing introns from a provided GRanges object.

Usage

```
remove_introns(gr_bins, grl_exbg)
```

Arguments

gr_bins	A GRanges object of exomePeak bins before the intron removal.
grl_exbg	A GRangesList object that define the exon regions of each genes.

Value

A GRangesList object with the same length of gr_bins, each list element corresponds to the original GRanges after the removal of introns.

replace_bg *Replace the control regions with user provided background.*

Description

Replace the control regions with user provided background.

Usage

```
replace_bg(grl, bg, txdb)
```

Arguments

grl	A GRangesList of the merged peaks which the background regions are waiting to be replaced.
bg	A GRanges or GRangesList object of the user provided background.
txdb	A TxDb object that define the transcript annotation.

Value

A GRangesList object. The first portion is the exons regions that is not overlapped with annoation.

If the resulting ranges have less number and width compared with what defined in cut_off_width or cut_off_num, the exon regions of txdb will be returned as the background.

The second portion is the restructed user provided annotation with gene id annotated.

Results-methods *Method Results*

Description

Method Results

Report the (Differential) Modification Peaks/Sites and their associated LFC Statistics

Usage

```
Results(sep, cut_off_pvalue = NULL, cut_off_padj = 0.05,
  cut_off_log2FC = 0, min_num_of_positive = 30,
  expected_direction = c("both", "hyper", "hypo"),
  inhibit_filter = FALSE, table_style = c("bed", "granges"))

## S4 method for signature 'SummarizedExomePeak'
Results(sep, cut_off_pvalue = NULL,
  cut_off_padj = 0.05, cut_off_log2FC = 0, min_num_of_positive = 30,
  expected_direction = c("both", "hyper", "hypo"),
  inhibit_filter = FALSE, table_style = c("bed", "granges"))
```

Arguments

sep	a SummarizedExomePeak object.
cut_off_pvalue	a numeric value for the p value cutoff in the exported result; Default = NULL.
cut_off_padj	a numeric value for the adjusted p value cutoff in the exported result; Default = 0.05.
cut_off_log2FC	a numeric value for the log2 fold change (LFC) cutoff of the exported result, only the sites with abs(LFC) larger than this value are kept; Default = 0.
min_num_of_positive	a numeric value for the minimum number of reported sites. If the number of remaining sites is less than this number after the filter, additional sites will be reported by the increasing order of the p value to meet this number.
expected_direction	a character for the expected direction of the differential modification, could be one in c("hyper", "hypo", "both"). hyper only report the peaks/sites with interactive LFC > 0. hypo only report the peaks/sites with interactive LFC < 0. both report the peaks/sites in both directions. This argument is useful when the treated group involves the perturbation of a known writer or eraser protein; Default "both".
inhibit_filter	a logical for whether to remove all the filters, this option is useful when quantification on single based site annotation; Default = FALSE.
table_style	a character for the style of the table being returned, could be one in c("bed","granges"). bed The genomic locations in the table are represented by BEDgraph style. granges The genomic locations in the table are represented by GRanges style.

Value

a data.frame containing the genomic locations of modification peaks/sites, gene ids, and their statistics.

Examples

```
### Load the example SummarizedExomPeak object
f1 = system.file("extdata", "sep_ex_mod.rds", package="exomePeak2")

sep <- readRDS(f1)

### Check the modification peaks/sites statistics.
head(Results(sep))
```

---

scanMeripBAM *Organize the BAM Files Information of a MeRIP-seq Data Set.*

Description

scanMeripBAM check and organize the BAM files in MeRIP-seq data before peak calling using exomePeakCalling. The library types of the RNA-seq and the filters such as SAM FLAG score are specified in this function.

Usage

```
scanMeripBAM(bam_ip = NULL, bam_input = NULL, bam_treated_ip = NULL,
  bam_treated_input = NULL, paired_end = FALSE,
  library_type = c("unstranded", "1st_strand", "2nd_strand"),
  index_bam = TRUE, bam_files = NULL, design_ip = NULL,
  design_treatment = NULL, mapq = 30L, isSecondaryAlignment = FALSE,
  isNotPassingQualityControls = FALSE, isDuplicate = FALSE,
  isPaired = NA, isProperPair = NA, hasUnmappedMate = NA, ...)
```

Arguments

bam_ip	a character vector for the BAM file directories of the (control) IP samples.
bam_input	a character vector for the BAM file directories of the (control) input samples.
bam_treated_ip	a character vector for the BAM file directories of the treated IP samples.
bam_treated_input	a character vector for the BAM file directories of the treated input samples. If the bam files do not contain treatment group, user should only fill the arguments of BAM_ip and BAM_input.
paired_end	a logical of whether the data comes from the Paired-End Library, TRUE if the data is Paired-End sequencing; default = FALSE.
library_type	a character specifying the protocal type of the RNA-seq library, can be one in c("unstranded", "1st_strand", "2nd_strand"); default = "unstranded".

unstranded The randomly primed RNA-seq library type, i.e. both the strands generated during the first and the second strand sythesis are sequenced; example: Standard Illumina.

1st_strand The first strand-specific RNA-seq library, only the strand generated during the first strand sythesis is sequenced; examples: dUTP, NSR, NNSR.

2nd_strand The second strand-specific RNA-seq library, only the strand generated during the second strand sythesis is sequenced; examples: Ligation, Standard SOLiD.

`index_bam` a `logical` value indicating whether to sort and index BAM files automatically if the bam indexes are not found; default = `TRUE`.

The BAM index files will be named by adding ".bai" after the names of the corresponding BAM files.

`bam_files` optional, a `character` vector for all the BAM file directories, if it is provided, the first 4 arguments above will be ignored.

`design_ip` optional, a `logical` vector indicating the information of IP/input, `TRUE` represents IP samples.

`design_treatment`

optional, a `logical` vector indicating the design of treatment/control, `TRUE` represents treated samples.

`mapq` a non-negative integer specifying the minimum reads mapping quality. BAM records with mapping qualities less than `mapq` are discarded; default = `30L`.

`isPaired, isProperPair, hasUnmappedMate, isSecondaryAlignment, isNotPassingQualityControls, isDu`

arguments specifying the filters on SAM FLAG scores, inherited from `ScanBamParam`.

Details

`scanMeripBAM` takes the input of the BAM file directories for the MeRIP-seq datasets. It first checks the completeness of the BAM files and the BAM indexes. Then, the design information of IP/input and treated/control are returned as a `MeripBamFileList` object. If the BAM file indexes are missing, the BAM files will be automatically indexed with the package `Rsamtools`.

Value

a `MeripBamFileList` object.

See Also

`exomePeakCalling`

Examples

```
### Define BAM File Directories

f1 = system.file("extdata", "IP1.bam", package="exomePeak2")
f2 = system.file("extdata", "IP2.bam", package="exomePeak2")
f3 = system.file("extdata", "IP3.bam", package="exomePeak2")
f4 = system.file("extdata", "IP4.bam", package="exomePeak2")
IP_BAM = c(f1,f2,f3,f4)
f1 = system.file("extdata", "Input1.bam", package="exomePeak2")
f2 = system.file("extdata", "Input2.bam", package="exomePeak2")
f3 = system.file("extdata", "Input3.bam", package="exomePeak2")
```

```
INPUT_BAM = c(f1,f2,f3)

f1 = system.file("extdata", "treated_IP1.bam", package="exomePeak2")
TREATED_IP_BAM = c(f1)
f1 = system.file("extdata", "treated_Input1.bam", package="exomePeak2")
TREATED_INPUT_BAM = c(f1)

### For MeRIP-Seq Experiment Without the Treatment Group

MeRIP_Seq_Alignment <- scanMeripBAM(
  bam_ip = IP_BAM,
  bam_input = INPUT_BAM,
  paired_end = FALSE
)

### For MeRIP-Seq Experiment With the Treatment Group

MeRIP_Seq_Alignment <- scanMeripBAM(
bam_ip = IP_BAM,
bam_input = INPUT_BAM,
bam_treated_ip = TREATED_IP_BAM,
bam_treated_input = TREATED_INPUT_BAM,
paired_end = FALSE
)
```

`split_by_name` *a function to split a GRanges into GRangesList by its names.*

Description

a function to split a GRanges into GRangesList by its names.

Usage

```
split_by_name(x)
```

Arguments

x	a list like object.

`SummarizedExomePeak-class` *SummarizedExomePeak*

Description

An S4 object defined in exomePeak2 that summarizes the information of modification peaks/sites, reads counts, size factors, GC contents, and the LFC related statistics.

This class contains SummarizedExperiment.

Usage

```
SummarizedExomePeak(...)
```

Arguments

...	arguments passed to new().

Details

Constructors:

The SummarizedExomePeak object can be contructed by 3 functions.

1. SummarizedExomePeak
2. exomePeakCalling
3. exomePeak2

Accessors:

The SummarizedExomePeak object share all the accessors with the SummarizedExperiment class.

It has 2 additional accessors:

1. GCsizeFactors
2. DESeq2Results

Examples

```
# Generate the SummarizedExomePeak object by peak calling

GENE_ANNO_GTF = system.file("extdata", "example.gtf", package="exomePeak2")

f1 = system.file("extdata", "IP1.bam", package="exomePeak2")
f2 = system.file("extdata", "IP2.bam", package="exomePeak2")
f3 = system.file("extdata", "IP3.bam", package="exomePeak2")
f4 = system.file("extdata", "IP4.bam", package="exomePeak2")
IP_BAM = c(f1,f2,f3,f4)
f1 = system.file("extdata", "Input1.bam", package="exomePeak2")
f2 = system.file("extdata", "Input2.bam", package="exomePeak2")
f3 = system.file("extdata", "Input3.bam", package="exomePeak2")
INPUT_BAM = c(f1,f2,f3)

sep <- exomePeak2(bam_ip = IP_BAM,
                  bam_input = INPUT_BAM,
                  gff_dir = GENE_ANNO_GTF,
                  genome = "hg19",
                  paired_end = FALSE)

#Access to the slots in the SummarizedExomePeak object

## Access to reads count
assay(sep)

## Access to the sequencing depth size factors and experimental design
colData(sep)
```

```
## Access to the GC content and feature length information
elementMetadata(sep)

## Access to the genomic locations of the modification peaks/sites and the background control regions
rowRanges(sep)

## Access to the feature specific size factors
GCsizeFactors(sep)

## Access to the statistics on (differential) modification LFC
DESeq2Results(sep)
```

Index