Package: ChrAccR (via r-universe)

August 27, 2024

Title Analyzing chromatin accessibility data in R

Description Tools for analyzing chromatin accessibility data in R.
Bulk and single-cell ATAC-seq data are supported.

Date 2023-03-08

Suggests ChrAccRex, muReportR, DelayedArray, HDF5Array, chromVAR, motifmatchr, chromVARmotifs, GenomicFeatures, Rsamtools, Biostrings, knitr, rmarkdown, jsonlite, cluster, matrixStats, preprocessCore, R.utils, tools, grid, cowplot, uwot, rtracklayer, DESeq2, limma, LOLA, qvalue, TFBSTools, seqLogo, pheatmap, grid, circlize, ComplexHeatmap, grDevices, BSgenome.Hsapiens.NCBI.GRCh38, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Hsapiens.1000genomes.hs37d5, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Mmusculus.UCSC.mm10, BSgenome.Mmusculus.UCSC.mm9, JASPAR2018

Imports S4Vectors, IRanges, GenomeInfoDb, GenomicRanges, GenomicAlignments, SummarizedExperiment, ggplot2, muLogR, muRtools, Matrix, methods, stats, utils, data.table

License GPL-2

Version 0.9.21

Collate 'ChrAccR.R' 'classDefs.R' 'utils.R' 'config.R' 'utils_atac.R' 'utils_motifs.R' 'annotation.R' 'utils_chromVAR.R' 'utils_filehandling.R' 'DsAcc-class.R' 'DsATAC-class.R' 'DsATACsc-class.R' 'DsNOMe-class.R' 'import_atac.R' 'import_atac_10x.R' 'import_atac_archr.R' 'importTools_nome.R' 'differential_nome.R' 'differential.R' 'report_atac_summary.R' 'report_atac_filtering.R' 'report_atac_normalization.R' 'report_atac_exploratory.R' 'report_atac_differential.R' 'workflow.R'

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.1

Repository https://blaserlab.r-universe.dev

RemoteUrl https://github.com/GreenleafLab/ChrAccR

RemoteRef HEAD

RemoteSha 43d010896dc95cedac3a8ea69aae3f67b2ced910

Contents

`addSampleAnnotCol,DsAcc-method`
addSampleAnnotCol-methods

Description

add a sample annotation column to the sample annotation table

Usage

```
## S4 method for signature 'DsAcc'
addSampleAnnotCol(.object, name, vals)
```

Arguments

.object	DsAcc object
name	a name for the new column
vals	vector of values

Value

a new DsAcc object with added sample annotation

Author(s)

Fabian Mueller

`aggregateRegionCounts,DsATAC-method`
aggregateRegionCounts-methods

Description

Agregate counts across a set of regions, e.g. for footprinting analysis

Usage

```
## S4 method for signature 'DsATAC'
aggregateRegionCounts(
  .object,
  regionGr,
  samples = getSamples(.object),
  countAggrFun = "sum",
  norm = "tailMean",
  normTailW = 0.1,
  kmerBiasAdj = TRUE,
  k = 6,
  sampleCovg = NULL,
  sampleKmerFreqM = NULL,
  regionKmerFreqM = NULL,
  silent = FALSE
)
```

Arguments

.object	DsATAC object
regionGr	GRanges object specifying the regions to aggregate over
samples	sample identifiers
countAggrFun	aggration function to be used for summarizing the insertion counts at each position. Possible values include "sum", "mean", and "median"
norm	method used for normalizing the resulting position-wise counts. Currently only 'tailMean' is supported, which computes normalization factors as the mean signal in the tails of the window
normTailW	fraction of the region window to be used on each side of the window to be used for normalization if norm is one of 'tailMean'
kmerBiasAdj	compute Tn5 bias and use it to adjust the counts as in Corces, et al., Science, (2018)
k	length of the kmer to be used for sequence bias correction. Only relevant if kmerBiasAdj==TRUE.
sampleCovg	to save compute time, a sample coverage track list (as computed by getCoverage(.object)) can be supplied. If not, it will be computed on the fly.
sampleKmerFreqM	to save compute time, a matrix of sample kmer frequency at insertion sites (as computed by getInsertionKmerFreq(.object, ...)) can be supplied. If not, it will be computed on the fly. Only relevant if kmerBiasAdj==TRUE.
regionKmerFreqM	to save compute time, a matrix of region kmer frequencies (kmers X window width). Must have the same number of rows as the specified (or computed) sampleKmerFreqM (kmers) and the same number of columns as the window width (median width of regionGr). Only relevant if kmerBiasAdj==TRUE.
silent	limit log messages

Value

a data.frame containing position-wise counts (raw, normalized and optionally Tn5-bias-corrected) for each sample

Author(s)

Fabian Mueller

callPeaks,DsATAC-method
callPeaks-methods

Description

Performs peak calling based on insertion sites

Usage

```
## S4 method for signature 'DsATAC'
callPeaks(
  .object,
  samples = getSamples(.object),
  method = "macs2_summit_fw_no",
  methodOpts = list(macs2.exec = "macs2", macs2.params = c("--shift", "-75",
   "--extsize", "150", "-p", "0.01"), fixedWidth = 250, genomeSizesFromObject = FALSE)
)
```

Arguments

.object	DsATAC object
samples	sample identifiers for which peak calling is performed
method	peak calling method. Currently only 'macs2_summit_fw_no' is supported. See details section.
methodOpts	list of other options depending on the 'method' parameter (see details section).

Details

The following methods are currently supported

'macs2_summit_fw_no' Fixed-width, non-overlapping peaks based on MACS2 summit calls: 1. Call peaks using system call to MACS2. You can specify the MACS2 executable in methodOpts$macs2.exec. 2. Identify peak summits 3. extend peak summits on each side by a number of basepairs (specified in methodOpts$fixedWidth; default: 250bp) to obtain unified peak widths 4. Find non-overlapping peaks by taking the peak with the best MACS2 score from each set of partially overlapping peaks

Value

GRangesList of peak coordinates for each sample

Author(s)

Fabian Mueller

ChrAccR *ChrAccR: Analyzing chromatin accessibility data in R.*

Description

Tools for analyzing chromatin accessibility data in R. Currently supports ATAC-seq and NOMe-seq data analysis.

cleanMem *cleanMem*

Description

clean the system mory by invoking the garbage collector

Usage

```
cleanMem(iter.gc = 1L)
```

Arguments

iter.gc	number of times to invoke the garbage collector

Value

nothing of particular interest

Author(s)

Fabian Mueller

collapseMotifMatrix *collapseMotifMatrix*

Description

Collapse TF motif matrix of arbitrary values by aggregating values over motif cluster assignment

Usage

```
collapseMotifMatrix(
  X,
  motifClust = NULL,
  assembly = "hg38",
  motifs = "jaspar",
  aggrFun = mean
)
```

Arguments

X	matrix to be collapsed. Must have the motif names as rownames. E.g. matrix obtained by chromVAR::deviationScores
motifClust	optional: motif clustering computed by getMotifClustering. If NULL (default) the default clustering will be retrieved
assembly	genome assembly for which the motif clustering should be retrieved. Only required if for automatic mode (i.e. motifClust=NULL).
motifs	a character string specifying the motif set (currently only "jaspar" is supported)
aggrFun	function to use to aggregate values

Value

list containing two elements: X: Collapsed matrix containing motif cluster aggregated values; clustering: clustering result used for aggregation (seegetMotifClustering for details)

Author(s)

Fabian Mueller

---

colZscores *colZscores*

Description

Performs z-score normalization on the columns of a matrix. (Basically a wrapper around matrixStats)

Usage

```
colZscores(X, na.rm = FALSE)
```

Arguments

X	input matrix
na.rm	should NAs be omitted?

Value

z-score normalized matrix

Author(s)

Fabian Mueller

computeDiffAcc.rnb.nome

computeDiffAcc.rnb.nome

Description

computes differential accessibility for NOMe datasets using RnBeads functionality

Usage

```
computeDiffAcc.rnb.nome(
  dsn,
  cmpCols,
  regionTypes = getRegionTypes(dsn),
  covgThres = 5L,
  allPairs = TRUE,
  adjPairCols = NULL,
  adjCols = NULL,
  skipSites = FALSE,
  disk.dump = rnb.getOption("disk.dump.big.matrices"),
  disk.dump.dir = tempfile(pattern = "diffMethTables_"),
  ...
)
```

Arguments

dsn	DsNOMe object
cmpCols	column names of the sample annotation of the dataset that will be used for comparison
regionTypes	which region types should be processed for differential analysis.
covgThres	coverage threshold for computing the summary statistics. See RnBeads::computeDiffTab.extended.si for details.
allPairs	Logical indicating whether all pairwise comparisons should be conducted, when more than 2 groups are present
adjPairCols	argument passed on to rnb.sample.groups. See its documentation for details.
adjCols	not used yet
skipSites	flag indicating whether differential methylation in regions should be computed directly and not from sites. This leads to skipping of site-specific differential methylation
disk.dump	Flag indicating whether the resulting differential methylation object should be file backed, ie.e the matrices dumped to disk
disk.dump.dir	disk location for file backing of the resulting differential methylation object. Only meaningful if disk.dump=TRUE. must be a character specifying an NON-EXISTING valid directory.
...	arguments passed on to binary differential methylation calling. See RnBeads::computeDiffTab.extende for details.

Value

an RnBDiffMeth object. See class description for details.

Author(s)

Fabian Mueller

Fabian Mueller

---

createReport_differential,DsATAC-method
createReport_differential-methods

Description

Create a report summarizing differential accessibility analysis

Usage

```
## S4 method for signature 'DsATAC'
createReport_differential(.object, reportDir)
```

Arguments

.object	DsATAC object
reportDir	directory in which the report will be created
chromVarObj	[optional] pre-computed result of a call to run_atac_chromvar(...)

Value

(invisible) muReportR::Report object containing the report

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
reportDir <- file.path(".", "ChrAccR_reports")
setConfigElement("regionTypes", setdiff(getRegionTypes(dsa), c("promoters_gc_protein_coding", "t10k")))
setConfigElement("differentialColumns", c("stimulus", "cellType"))
# adjust for the donor annotation in the differential test
setConfigElement("differentialAdjColumns", c("donor"))
# create the report
createReport_differential(dsa, reportDir)

## End(Not run)
```

createReport_exploratory,DsATAC-method
createReport_exploratory-methods

Description

Create a report summarizing exploratory analyses of an accessibility dataset

Usage

```
## S4 method for signature 'DsATAC'
createReport_exploratory(
  .object,
  reportDir,
  chromVarObj = NULL,
  itLsiObj = NULL,
  geneActSe = NULL
)
```

Arguments

.object	DsATAC object
reportDir	directory in which the report will be created
chromVarObj	[optional] pre-computed result of a call to run_atac_chromvar(...)
itLsiObj	[for single-cell only; optional] pre-computed result of a call to iterativeLSI(.object, ...)
geneActSe	[for single-cell only; optional] pre-computed result of a call to getCiceroGeneActivities(.object, ...)

Value

(invisible) muReportR::Report object containing the report

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
dsa_qnorm <- transformCounts(dsa, method="quantile")
setConfigElement("annotationColumns", c("cellType", "donor", "stimulus"))
setConfigElement("regionTypes", setdiff(getRegionTypes(dsa), c("promoters_gc_protein_coding", "t10k")))
reportDir <- file.path(".", "ChrAccR_reports")
createReport_exploratory(dsa_qnorm, reportDir)

## End(Not run)
```

createReport_filtering,DsATAC-method
createReport_filtering-methods

Description

Create a report summarizing steps and statistics

Usage

```
## S4 method for signature 'DsATAC'
createReport_filtering(.object, reportDir, unfilteredObj, filterStats = NULL)
```

Arguments

.object	filtered DsATAC object
reportDir	directory in which the report will be created
unfilteredObj	unfiltered DsATAC object
filterStats	filtering statistics as output by run_atac_filtering

Value

(invisible) muReportR::Report object containing the report

Author(s)

Fabian Mueller

createReport_normalization,DsATAC-method
createReport_normalization-methods

Description

Create a report summarizing normalization

Usage

```
## S4 method for signature 'DsATAC'
createReport_normalization(.object, reportDir, unnormObj)
```

Arguments

.object	normalized DsATAC object
reportDir	directory in which the report will be created
unnormObj	unnormalized DsATAC object

Value

(invisible) `muReportR::Report` object containing the report

Author(s)

Fabian Mueller

`createReport_summary,DsATAC-method`
createReport_summary-methods

Description

Create a report summarizing an accessibility dataset

Usage

```
## S4 method for signature 'DsATAC'
createReport_summary(.object, reportDir)
```

Arguments

`.object`	`DsATAC` object
`reportDir`	directory in which the report will be created

Value

(invisible) `muReportR::Report` object containing the report

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
reportDir <- file.path(".", "ChrAccR_reports")
createReport_summary(dsa, reportDir)

## End(Not run)
```

dimRed_UMAP,DsATACsc-method
dimRed_UMAP-methods

Description

Retrieve dimension reduction embedding and object using UMAP

Usage

```
## S4 method for signature 'DsATACsc'
dimRed_UMAP(
  .object,
  regions,
  tfidf = TRUE,
  pcs = 1:50,
  normPcs = FALSE,
  umapParams = list(distMethod = "euclidean", min_dist = 0.5, n_neighbors = 25),
  rmDepthCor = 1
)
```

Arguments

.object	DsATACsc object
regions	character string specifying the region type to retrieve the UMAP coordinates from. Alternatively, a GRanges object specifying coordinates that fragment counts will be aggregated over
tfidf	normalize the counts using TF-IDF transformation
pcs	components to use to compute the SVD
normPcs	flag indicating whether to apply z-score normalization to PCs for each cell
umapParams	parameters to compute UMAP coordinates (passed on to muRtools::getDimRedCoords.umap and further to uwot::umap)
rmDepthCor	correlation cutoff to be used to discard principal components associated with fragment depth (all iteratiοna). By default (value >=1) no filtering will be applied.

Details

The output object includes the final singular values/principal components (result$pcaCoord), the low-dimensional coordinates (result$umapCoord) as well as region set that provided the basis for the dimension reduction (result$regionGr).

Value

an S3 object containing dimensionality reduction results

Author(s)

Fabian Mueller

`DsAcc-class` *DsAcc*

Description

A class for accessibility datasets

Slots

`coord` List of coordinates (GRanges objects) for accessibility summarized regions.

`sampleAnnot` Sample annotation Table

`genome` Genome assembly

`diskDump` Flag indicating whether large matrices and objects will be kept on disk rather than in main memory.

`pkgVersion` Version number of the ChrAccR package that created the object

Author(s)

Fabian Mueller

`DsATAC-class` *DsATAC*

Description

A class for storing ATAC-seq accessibility data inherits from `DsAcc`

Slots

`fragments` `GRanges` object storing sequencing fragments. Alternativily pointers to files in which this data isf stored as R data object

`counts` List of count matrices for each summarized region type (dimension: regions X samples). Depending on the settings for the slots `diskDump` and `sparseCounts`, the matrices are either (a) regular matrices, (b) `HDF5Array`/`DelayedArray` or (c) sparse matrices.

`countTransform` list of character vectors specifying which transformations have been applied to the count matrices

`sparseCounts` Flag indicating whether count data will be stored as sparse matrices rather than regular matrices

`diskDump.fragments` Flag indicating whether fragment data will be kept on disk rather than in main memory.

Author(s)

Fabian Mueller

DsATAC.bam *DsATAC.bam*

Description

Create a DsATAC dataset from multiple input bam files

Usage

```
DsATAC.bam(
  sampleAnnot,
  bamFiles,
  genome,
  regionSets = NULL,
  sampleIdCol = NULL,
  diskDump = FALSE,
  keepInsertionInfo = TRUE,
  pairedEnd = TRUE
)
```

Arguments

sampleAnnot	data.frame specifying the sample annotation table
bamFiles	either a character vector of the same length as sampleAnnot has rows, specifying the file paths of the bam files for each sample or a single character string specifying the column name in sampleAnnot where the file paths can be found
genome	genome assembly
regionSets	a list of GRanges objects which contain region sets over which count data will be aggregated
sampleIdCol	column name in the sample annotation table containing unique sample identifiers. If NULL (default), the function will look for a column that contains the word "sample"
diskDump	should large data objects (count matrices, fragment data, ...) be disk-backed to save main memory
keepInsertionInfo	flag indicating whether to maintain the insertion information in the resulting object. Only relevant when type=="insBam".
pairedEnd	is the input data paired-end? Only relevant when type=="insBam".

Value

DsATAC object

Author(s)

Fabian Mueller

Examples

```
## Not run:
# download and unzip the dataset
datasetUrl <- "https://s3.amazonaws.com/muellerf/data/ChrAccR/data/tutorial/tcells.zip"
downFn <- "tcells.zip"
download.file(datasetUrl, downFn)
unzip(downFn, exdir=".")
# prepare the sample annotation table
sampleAnnotFn <- file.path("tcells", "samples.tsv")
bamDir <- file.path("tcells", "bam")
sampleAnnot <- read.table(sampleAnnotFn, sep="\t", header=TRUE, stringsAsFactors=FALSE)
# add a column that ChrAccR can use to find the correct bam file for each sample
sampleAnnot[,"bamFilenameFull"] <- file.path(bamDir, sampleAnnot[,"bamFilename"])
# prepare the dataset
dsa_fromBam <- DsATAC.bam(sampleAnnot, "bamFilenameFull", "hg38", regionSets=NULL, sampleIdCol="sampleId")

## End(Not run)
```

`DsATAC.cellranger` *DsATAC.cellranger*

Description

Create a DsATAC dataset from multiple input files output by 10x cellranger

Usage

```
DsATAC.cellranger(
  sampleAnnot,
  sampleDirPrefixCol,
  genome,
  dataDir = "",
  regionSets = NULL,
  addPeakRegions = TRUE,
  sampleIdCol = sampleDirPrefixCol,
  keepInsertionInfo = FALSE,
  diskDump.fragments = keepInsertionInfo
)
```

Arguments

`sampleAnnot`	data.frame specifying the sample annotation table
`sampleDirPrefixCol`	column name specifying the directory prefix for each sample in the sample annotation table
`genome`	genome assembly
`dataDir`	directory where the files are located

regionSets — a list of GRanges objects which contain region sets over which count data will be aggregated

addPeakRegions — should a merged set of peaks be created as one of the region sets (merged, non-overlapping peaks of width=500bp from the peaks of individual samples)

sampleIdCol — column name or index in the sample annotation table containing unique sample identifiers

keepInsertionInfo — flag indicating whether to maintain the insertion information in the resulting object.

diskDump.fragments — Keep fragment coordinates stored on disk rather than in main memory. This saves memory, but increases runtime and I/O.

Value

DsATAC object

Author(s)

Fabian Mueller

DsATAC.fragmentBed *DsATAC.fragmentBed*

Description

Create a DsATAC dataset from multiple input fragment bed files

Usage

```
DsATAC.fragmentBed(
  sampleAnnot,
  bedFiles,
  genome,
  regionSets = NULL,
  sampleIdCol = NULL,
  diskDump = FALSE,
  keepInsertionInfo = TRUE
)
```

Arguments

sampleAnnot — data.frame specifying the sample annotation table

bedFiles — either a character vector of the same length as sampleAnnot has rows, specifying the file paths of the bed files for each sample or a single character string specifying the column name in sampleAnnot where the file paths can be found

genome — genome assembly

regionSets — a list of GRanges objects which contain region sets over which count data will be aggregated

sampleIdCol — column name in the sample annotation table containing unique sample identifiers. If NULL (default), the function will look for a column that contains the word "sample"

diskDump — should large data objects (count matrices, fragment data, ...) be disk-backed to save main memory

keepInsertionInfo — flag indicating whether to maintain the insertion information in the resulting object. Only relevant when type=="insBam".

Value

DsATAC object

Author(s)

Fabian Mueller

DsATAC.snakeATAC *DsATAC.snakeATAC*

Description

Create a DsATAC dataset from multiple input files output by snakeATAC

Usage

```
DsATAC.snakeATAC(
  sampleAnnot,
  filePrefixCol,
  genome,
  dataDir = "",
  regionSets = NULL,
  sampleIdCol = filePrefixCol,
  type = "insBam",
  diskDump = FALSE,
  keepInsertionInfo = TRUE,
  bySample = FALSE,
  pairedEnd = TRUE
)
```

Arguments

sampleAnnot	data.frame specifying the sample annotation table
filePrefixCol	column name specifying the file prefix for each sample in the sample annotation table. If dataDir is not empty (i.e. not "") filenames are assumed to be relative to that directory and a corresponding filename suffix will be appended
genome	genome assembly
dataDir	directory where the files are located. If it is the empty character (""; default) it is assumed that filePrefixCol specifies the full path to the input files
regionSets	a list of GRanges objects which contain region sets over which count data will be aggregated
sampleIdCol	column name or index in the sample annotation table containing unique sample identifiers
type	input data type. Currently only "insBed" (insertion beds), "insBam" (insertion info inferred from bam files (aligned reads); default) and "bam" (aligned reads) are supported
diskDump	should large data objects (count matrices, fragment data, ...) be disk-backed to save main memory
keepInsertionInfo	flag indicating whether to maintain the insertion information in the resulting object. Only relevant when type=="insBam".
bySample	process sample-by-sample to save memory (currently only has an effect for type=="insBam")
pairedEnd	is the input data paired-end? Only relevant when type=="insBam".

Value

DsATAC object

Author(s)

Fabian Mueller

DsATACsc-class *DsATACsc*

Description

A class for storing single-cell ATAC-seq accessibility data inherits from DsATAC. Provides a few additional methods but is otherwise identical to DsATAC.

Author(s)

Fabian Mueller

DsATACsc.archr *DsATACsc.archr*

Description

Create a DsATACsc dataset from an ArchR project

Usage

```
DsATACsc.archr(
  ap,
  useTiling = TRUE,
  keepInsertionInfo = FALSE,
  diskDump.fragments = keepInsertionInfo
)
```

Arguments

ap	ArchR project object
useTiling	flag indicating whether to use tiling information from the ArchR project
keepInsertionInfo	flag indicating whether to maintain the insertion information in the resulting object.
diskDump.fragments	Keep fragment coordinates stored on disk rather than in main memory. This saves memory, but increases runtime and I/O.

Value

DsATACsc object

Author(s)

Fabian Mueller

DsATACsc.fragments *DsATACsc.fragments*

Description

Create a DsATACsc dataset from multiple input fragment files

Usage

```
DsATACsc.fragments(
  sampleAnnot,
  fragmentFiles,
  genome,
  regionSets = NULL,
  sampleIdCol = NULL,
  minFragsPerBarcode = 500L,
  maxFragsPerBarcode = Inf,
  cellAnnot = NULL,
  keepInsertionInfo = FALSE,
  diskDump.fragments = keepInsertionInfo,
  cellQcStats = TRUE
)
```

Arguments

sampleAnnot	data.frame specifying the sample annotation table
fragmentFiles	vector of fragment files or the column name in the sample annotation table containing thse file names. fragment files must be tab-separated with columns "chrom", "chromStart", "chromEnd", "barcode" and "duplicateCount" and must not contain a header line
genome	genome assembly
regionSets	a list of GRanges objects which contain region sets over which count data will be aggregated
sampleIdCol	column name or index in the sample annotation table containing unique sample identifiers
minFragsPerBarcode	minimum number of fragments required for a barcode to be kept. [Only relevant if cellAnnot==NULL]
maxFragsPerBarcode	maximum number of fragments per barcode. Only barcodes with fewer fragments will be kept. [Only relevant if cellAnnot==NULL]
cellAnnot	(optional) annotation table of all cells in the dataset. Must contain a 'cellId' and 'cellBarcode' columns.
keepInsertionInfo	flag indicating whether to maintain the insertion information in the resulting object.
diskDump.fragments	Keep fragment coordinates stored on disk rather than in main memory. This saves memory, but increases runtime and I/O.
cellQcStats	flag indicating whether to compute additional cell QC statistics (TSS enrichment, etc.).

Value

DsATACsc object

Author(s)

Fabian Mueller

`DsNOMe-class` *DsNOMe*

Description

A class for storing NOMe accessibility data

Slots

`meth` List of GC methylation for sites and summarized regions.

`covg` List of GC read coverage for sites and summarized regions.

Author(s)

Fabian Mueller

`DsNOMe.bisSNP` *DsNOMe.bisSNP*

Description

Create a DsNOMe dataset from multiple input files in bisSNP output format

Usage

```
DsNOMe.bisSNP(inputFns, sampleAnnot, genome, sampleIds = rownames(sampleAnnot))
```

Arguments

`inputFns`	a NAMED vector of input file names
`sampleAnnot`	data.frame specifying the sample annotation table
`genome`	genome assembly
`sampleIds`	character vector of sample names

Value

`DsNOMe` object

Author(s)

Fabian Mueller

`exportCountTracks,DsATAC-method` *exportCountTracks-methods*

Description

export count data as genome tracks (e.g. for visualization in the browser)

Usage

```
## S4 method for signature 'DsATAC'
exportCountTracks(
  .object,
  type,
  outDir,
  formats = c("bed", "igv"),
  groupBy = NULL
)
```

Arguments

`.object`	DsATAC object
`type`	character string specifying the region type
`outDir`	output directory. Must be existing.
`formats`	browser format. Currently only bed and "igv" are supported
`groupBy`	a column in the sample annotation table to group by (the mean will be computed)

Value

nothing of particular interest

Author(s)

Fabian Mueller

`fastqDirToTable` *fastqDirToTable*

Description

scan a directory containing fastq files and create a sample annotation table from the file names

Usage

```
fastqDirToTable(fqDir, tabFn = NULL, pat = "")
```

Arguments

fqDir	string specifying a directory with fastq files
tabFn	filename specifying the where the table should be written to. If NULL (default), the table will just be returned as data frame
pat	(optional) regular expression that fastq file names have to pass

Value

data frame of parsed annotation

Author(s)

Fabian Mueller

---

filterByGRanges,DsATAC-method

filterByGRanges-methods

Description

Filter out regions based on a GRanges object

Usage

```
## S4 method for signature 'DsATAC'
filterByGRanges(.object, gr, method = "black")
```

Arguments

.object	DsATAC object
gr	GRanges object used for filtering
method	character string specifying the method. Can be "white" to treat gr as a whitelist (i.e. only regions and fragments overlapping with it are retained) or "black" (default) to treat it as a blacklist.

Value

a new, filtered DsATAC object

Author(s)

Fabian Mueller

`filterCellsTssEnrichment,DsATACsc-method`
filterCellsTssEnrichment-methods

Description

Filter out cells with low TSS enrichment

Usage

```
## S4 method for signature 'DsATACsc'
filterCellsTssEnrichment(.object, cutoff = 6)
```

Arguments

`.object`	DsATAC object
`cutoff`	TSS enrichment cutoff to filter cells

Value

modified DsATAC object without filtered cells

Author(s)

Fabian Mueller

`filterChroms,DsATAC-method`
filterChroms-methods

Description

Filter out regions based on chromosome list

Usage

```
## S4 method for signature 'DsATAC'
filterChroms(.object, exclChrom = c("chrX", "chrY", "chrM"))
```

Arguments

`.object`	DsATAC object
`exclChrom`	vector of chromosome names to filter out

Value

a new DsATAC object filtered for chromosomes

Author(s)

Fabian Mueller

`filterLowCovg,DsATAC-method`
filterLowCovg-methods

Description

Filter regions with low read counts

Usage

```
## S4 method for signature 'DsATAC'
filterLowCovg(
  .object,
  thresh = 1L,
  reqSamples = 0.75,
  regionTypes = getRegionTypes(.object)
)
```

Arguments

`.object`	DsATAC object
`thresh`	regions with read counts below this threshold will be considered lowly covered regions (default: regions with fewer than 1 read will be discarded)
`reqSamples`	the percentile of samples required to meet or exceed the threshold in order for a region to be retained. must be in the interval [0, 1) (default: 0.75 = 75 percent)
`regionTypes`	character vector specifying the names of the region types to which filtering should be applied (default: all region types)

Value

a new DsATAC object with low coverage regions removed

Author(s)

Fabian Mueller

findNearestGeneForGr *findNearestGeneForGr*

Description

get gene annotation for a GRanges object by linking to the nearest gene

Usage

```
findNearestGeneForGr(gr, geneGr = NULL, maxDist = 1e+05)
```

Arguments

gr	GRanges object
geneGr	gene annotation from which to pull the annotation. Can be NULL for automatic retrieval of annotation. Must be named or have a gene name column in the metadata
maxDist	maximum distance for matching to nearest gene

Value

data.frame containing information on the nearest gene for each element in gr

findOrderedNames *findOrderedNames*

Description

find the first occurrence of a name in a vector of strings

Usage

```
findOrderedNames(x, orderedNames, exact = TRUE, ignore.case = FALSE)
```

Arguments

x	character vector in which the name should be found
orderedNames	vector of names that will be queried. This method will go through them one by one and find the first occurrence in the order of the orderedNames provided
exact	should only be exact matches be reported
ignore.case	should casing be ignored

Value

the string that matches the first occurrence in the order of orderedNames. Returns NA if no match is found.

Author(s)

Fabian Mueller

---

getATACfragments *getATACfragments*

Description

Given a GAlignmentPairs or GAlignments object, return a GRanges object containing the fragment (or insertion site for single-end data)

Usage

```
getATACfragments(ga, offsetTn = TRUE)
```

Arguments

ga	GAlignmentPairs (or GAlignments for single-end sequencing) object
offsetTn	apply offsets for Tn5 dimer cut site (+4 bp on genomic + strand; -4 bp on genomic - strand)

Value

GRanges object containing derived insertions. For paired-end data (recommended), the width of the resulting ranges corresponds to the insert size for single-end data, the width is set to 1bp

Author(s)

Fabian Mueller

getChrAccRAnnotationPackage

getChrAccRAnnotationPackage

Description

retrieve the corresponding ChrAccRAnnotation package for a given genome

Usage

```
getChrAccRAnnotationPackage(genome)
```

Arguments

genome	character string specifying the genome

Value

name of the annotation package, if installed. NULL and a warning if the package is not installed

Author(s)

Fabian Mueller

getChromVarDev,DsATAC-method

getChromVarDev-methods

Description

Compute chromVar deviations

Usage

```
## S4 method for signature 'DsATAC'
getChromVarDev(.object, type, motifs = "jaspar")
```

Arguments

.object	DsATAC object
type	character string specifying the region type
motifs	either a character string (currently only "jaspar" and sets contained in chromVARmotifs ("homer", "encode", "cisbp") are supported) or an object containing PWMs that can be used by motifmatchr::matchMotifs (such as an PFMatrixList or PWMatrixList object)

Value

Deviations object as returned by `chromVAR::computeDeviations`

Author(s)

Fabian Mueller

`getCiceroGeneActivities,DsATAC-method`
getCiceroGeneActivities-methods

Description

Obtain Cicero gene activities

Usage

```
## S4 method for signature 'DsATAC'
getCiceroGeneActivities(
  .object,
  regionType,
  promoterGr = NULL,
  maxDist = 250000L,
  corCutOff = 0.35,
  dimRedCoord = NULL,
  knn.k = 50
)
```

Arguments

`.object`	`DsATAC` object
`regionType`	region type of regions that will be linked to the promoter (typical some sort of peak annotation)
`promoterGr`	`GRanges` object of promoter coordinates
`maxDist`	maximum distance to consider for region-region interactions
`corCutOff`	cutoff of correlation coefficients (Pearson) to consider for region-region interactions
`dimRedCoord`	matrix of reduced dimension coordinates. must have coordinates for all samples/cells in the dataset
`knn.k`	parameter k for Cicero's k-nearest-neighbor method

Value

an `SummarizedExperiment` object containing gene activities for all cells/samples in the dataset

Author(s)

Fabian Mueller

`getComparisonInfo` *getComparisonInfo*

Description

retrieve the comparison information for an DsAcc object. Analogous to `RnBeads::get.comparison.info`

Usage

```
getComparisonInfo(
  dsa,
  cmpNames = NULL,
  regionTypes = getRegionTypes(dsa),
  allPairs = TRUE,
  adjPairCols = NULL,
  minGrpSize = 1L,
  maxGrpCount = NULL
)
```

Arguments

`dsa`	`DsAcc` object
`cmpNames`	column names of the sample annotation of the dataset that will be used for comparison
`regionTypes`	which region types should be processed for differential analysis.
`allPairs`	Logical indicating whether all pairwise comparisons should be conducted, when more than 2 groups are present
`adjPairCols`	argument passed on to `rnb.sample.groups`. See its documentation for details.
`minGrpSize`	Minimum number of samples required to form a group in comparison
`maxGrpCount`	maximum number of groups to consider for comparisons

Value

a list containing one element for each comparison to be conducted. Each element is again a list containing:

`comparison` the name of the comparison

`pheno.colname` the column name of the sample annotation table the comparison is derived from

`group.names` the names of the two groups being compared

`group.inds` the sample indices of the samples belonging to the two groups

`paired` flag indicating whether paired analysis is conducted

`adj.sva` flag indicating whether adjustment for SVA is conducted

`adj.celltype` flag indicating whether adjustment for cell type is conducted

`adjustment.table` the covariate adjustment table. `NULL` if the comparison is not adjusted

`region.types` the region types applicable to the analysis

Author(s)

Fabian Mueller

`getComparisonTable,DsAcc-method`
getComparisonTable-methods

Description

Retrieve a table describing pairwise comparisons on a `DsAcc` object

Usage

```
## S4 method for signature 'DsAcc'
getComparisonTable(
  .object,
  cols = NULL,
  cols1vAll = NULL,
  compNames = NULL,
  minGroupSize = 2L,
  maxGroupCount = length(.object) - 1
)
```

Arguments

`.object`	`DsAcc` object
`cols`	column names in the sample annotation table to consider for pairwise comparisons
`cols1vAll`	column names in the sample annotation table to consider for 1-vs-all comparisons
`compNames`	vector of character strings specifying a fixed comparison names to be parsed (format "$GRP1_NAME vs $GRP1_NAME [$ANNOTATION_COLUMN]")
`minGroupSize`	Minimum size of a group to be used in comparison. Affects the annotation columns that will be used for comparisons.
`maxGroupCount`	Maximum number of groups for a column to be considered for comparison.

Value

a `data.frame` with comparison inforamtion containing columns for the comparison name (`compName`), column in the annotation table (`compCol`) and group names for the two groups in the comparison (`grp1Name`, `grp2Name`),

Author(s)

Fabian Mueller

`getConfigElement` *getConfigElement*

Description

Get the value for a configuration item

Usage

```
getConfigElement(name)
```

Arguments

`name`	name of the config item

Value

the value of the config item

Author(s)

Fabian Mueller

`getConsensusPeakSet` *getConsensusPeakSet*

Description

Retrieve a consensus peak set from a set of peak lists

Usage

```
getConsensusPeakSet(
  grl,
  mode = "no_by_score",
  grouping = NULL,
  groupAgreePerc = 1,
  groupConsSelect = FALSE,
  scoreCol = "score",
  keepOvInfo = FALSE
)
```

Arguments

grl	list or GRangesList object containing the peak sets for each sample
mode	consensus mode. Currently only "no_by_score" (non-overlapping; i.e. select the peak with the highest score from each set of overlapping peaks) is supported.
grouping	vector of group memberships (numeric, character or factor). must be of the same length as grl
groupAgreePerc	percentile of members in a group required to contain a peak in order to keep it. E.g. a value of 1 (default) means that all replicates in a group are required to contain that peak in order to keep it.
groupConsSelect	if set, the peak set will also be checked for consistency, i.e. in order to retain a peak it has to be consistently be present or absent in each group (as specified in groupAgreePerc percent of samples)
scoreCol	name of the column to be used as score in the elementMetadata of the peak sets. This will determine which peak is selected if multiple peaks overlap
keepOvInfo	keep annotation columns in the elementMetadata of the results specifying whether a consensus peak overlaps with a peak in each sample

Value

GRanges object the containing consensus peak set

Author(s)

Fabian Mueller

---

getCoord,DsAcc-method *getCoord-methods*

Description

Return coordinates of sites/regions in a dataset

Usage

```
## S4 method for signature 'DsAcc'
getCoord(.object, type)
```

Arguments

.object	DsAcc object
type	character string specifying the rgion type or "sites" (default)

Value

GRanges object containing coordinates for covered sites/regions

Author(s)

Fabian Mueller

`getCounts,DsATAC-method` *getCounts-methods*

Description

Return table of count values

Usage

```
## S4 method for signature 'DsATAC'
getCounts(
  .object,
  type,
  i = NULL,
  j = NULL,
  asMatrix = TRUE,
  naIsZero = TRUE,
  allowSparseMatrix = FALSE
)
```

Arguments

`.object`	`DsATAC` object
`type`	character string specifying the region type
`i`	(optional) row (region) indices
`j`	(optional) column (sample) indices
`asMatrix`	return a matrix object instead of the internal representation
`naIsZero`	should `NA`s in the count matrix be considered 0 value (instead of unknown/missing)
`allowSparseMatrix`	if `asMatrix`: allow for sparse matrices as returned data format

Value

Matrix containing counts for each region and sample

Author(s)

Fabian Mueller

getCountsSE,DsATAC-method
getCountsSE-methods

Description

Return a SummarizedExperiment object of count values

Usage

```
## S4 method for signature 'DsATAC'
getCountsSE(.object, type, naIsZero = TRUE)
```

Arguments

.object	DsATAC object
type	character string specifying the region type
naIsZero	should NAs in the count matrix be considered 0 value (instead of unknown/missing)

Value

SummarizedExperiment containing counts for each region and sample

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
se <- getCountsSE(dsa, "IA_prog_peaks")
se

## End(Not run)
```

getCoverage,DsATAC-method
getCoverage-methods

Description

Return a list of genome-wide coverage from insertion sites

Usage

```
## S4 method for signature 'DsATAC'
getCoverage(.object, samples = getSamples(.object))
```

Arguments

.object	DsATAC object
samples	sample identifiers

Value

list of Rle objects of sample coverage tracks

Author(s)

Fabian Mueller

---

getCovg,DsNOMe-method *getCovg-methods*

Description

Return table of read coverage values

Usage

```
## S4 method for signature 'DsNOMe'
getCovg(.object, type = "sites", asMatrix = FALSE)
```

Arguments

.object	DsNOMe object
type	character string specifying the rgion type or "sites" (default)
asMatrix	return a matrix instead of a data.table

Value

data.table or matrix containing read coverage for each site/region and sample

Author(s)

Fabian Mueller

`getDESeq2Dataset,DsATAC-method`
getDESeq2Dataset-methods

Description

Retrieve a differential expression dataset computed with DESeq2

Usage

```
## S4 method for signature 'DsATAC'
getDESeq2Dataset(.object, regionType, designCols = NULL, compTab = NULL, ...)
```

Arguments

`.object`	`DsATAC` object
`regionType`	character string specifying the region type
`designCols`	column names in the sample annotation potentially used to create the design matrix
`compTab`	if design columns are not specified, you can specify a comparison table directly. These comparison tables can be obtained by `getComparisonTable(...)`
`...`	parameters passed on to `DESeq2::DESeq`

Value

`DESeqDataSet` as returned by `DESeq2::DESeq`

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
dds <- getDESeq2Dataset(dsa, "IA_prog_peaks", designCols=c("donor", "stimulus", "cellType"))
dds

## End(Not run)
```

getDiffAcc,DsATAC-method
getDiffAcc-methods

Description

Compute differential accessibility

Usage

```
## S4 method for signature 'DsATAC'
getDiffAcc(
  .object,
  regionType,
  comparisonCol,
  grp1Name = NULL,
  grp2Name = NULL,
  adjustCols = character(0),
  method = "DESeq2",
  diffObj = NULL
)
```

Arguments

.object	DsATAC object
regionType	character string specifying the region type
comparisonCol	column name in the sample annotation table to base the comparison on
grp1Name	name of the first group in the comparison. if not specified, it will be taken as the first factor level specified in the sample annotation table in 'comparisonCol'.
grp2Name	name of the second group (reference) in the comparison. if not specified, it will be taken as the first factor level specified in the sample annotation table in 'comparisonCol'.
adjustCols	column names in the sample annotation potentially used to create the design matrix
method	Method for determining differential accessibility. Currently only 'DESeq2' is supported
diffObj	optional differential object to avoid computing it for each comparison and thus reduce runtime

Value

a data.frame containing differential accessibility information

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
daTab <- getDiffAcc(dsa, "IA_prog_peaks", "stimulus", grp1Name="S", grp2Name="U", adjustCols=c("cellType", "donor

## End(Not run)
```

`getFragmentGr,DsATAC-method`
getFragmentGr-methods

Description

Return a GRanges object of fragment data for a given sample

Usage

```
## S4 method for signature 'DsATAC'
getFragmentGr(.object, sampleId)
```

Arguments

.object	DsATAC object
sampleId	sample identifier

Value

GRanges object containing fragment data

Author(s)

Fabian Mueller

`getFragmentGrl,DsATAC-method`
getFragmentGrl-methods

Description

Return a list of GRanges objects of fragment data for a given set of samples

Usage

```
## S4 method for signature 'DsATAC'
getFragmentGrl(.object, sampleIds, asGRangesList = FALSE)
```

Arguments

.object	DsATAC object
sampleIds	sample identifiers
asGRangesList	should a GRangesList object be returned instead of a regular list

Value

A named list of GRanges objects containing fragment data

Author(s)

Fabian Mueller

---

getFragmentNum,DsATAC-method
getFragmentNum-methods

Description

Return the number of fragments in the DsATAC object

Usage

```
## S4 method for signature 'DsATAC'
getFragmentNum(.object, sampleIds = getSamples(.object))
```

Arguments

.object	DsATAC object
sampleIds	sample identifiers

Value

a vector of fragment counts per sample

Author(s)

Fabian Mueller

---

getGenome,DsAcc-method *getGenome-methods*

Description

Return the genome assembly

Usage

```
## S4 method for signature 'DsAcc'
getGenome(.object)
```

Arguments

.object	DsAcc object

Value

Character string containing the genome assembly

Author(s)

Fabian Mueller

---

getGenomeObject *getGenomeObject*

Description

retrieve the appropriate BSgenome for an assembly string

Usage

```
getGenomeObject(assembly, adjChrNames = TRUE)
```

Arguments

assembly	string specifying the assembly
adjChrNames	should the prefix "chr" be added to main chromosomes if not already present and chrMT be renamed to chrM?

Value

BSgenome object

getGroupsFromTable *getGroupsFromTable*

Description

Retrieve groupings given a table containing some categorical columns

Usage

```
getGroupsFromTable(tt, cols = NULL, minGrpSize = 2, maxGrpCount = nrow(tt) - 1)
```

Arguments

tt	table to retrieve groupings for
cols	(Optional) predefined column names (in the form of a character vector) or indices (an integer vector) to consider. All other columns in the annotation table will be ignored.
minGrpSize	Minimum number of items required to form a group in comparison
maxGrpCount	Maximum number of groups to be considered

Value

List of groupings. Each element corresponds to a categorical column in the table and contains the row indices for each category

Author(s)

Fabian Mueller

---

getInsertionKmerFreq,DsATAC-method *getInsertionKmerFreq-methods*

Description

compute kmer frequencies at insertion sites for each sample

Usage

```
## S4 method for signature 'DsATAC'
getInsertionKmerFreq(
  .object,
  samples = getSamples(.object),
  k = 6,
  normGenome = FALSE
)
```

Arguments

.object	DsATAC object
samples	sample identifiers
k	length of the kmer
normGenome	should the result be normalized by genome-wide kmer frequencies

Value

a matrix containing kmer frequencies (one row for each kmer and one column for each sample in the dataset)

Author(s)

Fabian Mueller

---

getInsertionSites,DsATAC-method
getInsertionSites-methods

Description

Return a list of insertion sites (Tn5 cut sites) for each sample

Usage

```
## S4 method for signature 'DsATAC'
getInsertionSites(
  .object,
  samples = getSamples(.object),
  asGRangesList = FALSE
)
```

Arguments

.object	DsATAC object
samples	sample identifiers
asGRangesList	should a GRangesList object be returned instead of a regular list

Value

list or GRangesList containing Tn5 cut sites for each sample

Author(s)

Fabian Mueller

`getJasparAnnot` *getJasparAnnot*

Description

retrieve motif annotation data

Usage

```
getJasparAnnot(ss, type = "humantfs")
```

Arguments

`ss`	character vector or JASPAR identifiers
`type`	annotation type. Currently only `"humantfs"` (pulls info from humantfs.ccbr.utoronto.ca) is supported

Value

list of data frames of TF annotation (a motif can have multiple annotated TFs)

Author(s)

Fabian Mueller

`getJasparSymbols` *getJasparSymbols*

Description

Retrieve the TF names (symbols) from a JASPAR identifier

Usage

```
getJasparSymbols(ss)
```

Arguments

`ss`	character vector or JASPAR identifiers

Value

a list of TF names (symbols) for each identifier

getMeth,DsNOMe-method *getMeth-methods*

Description

Return table of methylation values

Usage

```
## S4 method for signature 'DsNOMe'
getMeth(.object, type = "sites", asMatrix = FALSE)
```

Arguments

.object	DsNOMe object
type	character string specifying the rgion type or "sites" (default)
asMatrix	return a matrix instead of a data.table

Value

data.table or matrix containing methylation levels for each site/region and sample

Author(s)

Fabian Mueller

getMonocleCellDataSet,DsATAC-method *getMonocleCellDataSet-methods*

Description

Obtain cell_data_set object for analysis using the monocle3 package

Usage

```
## S4 method for signature 'DsATAC'
getMonocleCellDataSet(.object, regionType, binarize = TRUE)
```

Arguments

.object	DsATAC object
regionType	name of the region type to be exported
binarize	should the counts be binarized

Value

a `cell_data_set` object containing the counts for the specified region type

Author(s)

Fabian Mueller

`getMotifClustering` *getMotifClustering*

Description

Retrieve motif clustering of TF motifs

Usage

```
getMotifClustering(
  k = 0,
  distM = NULL,
  assembly = "hg38",
  motifs = "jaspar",
  clusterMethod = "pam"
)
```

Arguments

`k`	number of clusters. `k<1` will result in an automatically selected clustering which is precomputed and stored in `ChrAccR`. For `distMethod=="jaspar"` and `clusterMethod=="pam"` this corresponds to the k corresponding to the best silhouette value before a drop (in the silhouette elbow-curve) occurs
`distM`	distance matrix (`dist` object) containing motif dissimilarities/distances. Only required if `k>0`.
`assembly`	genome assembly for which the motifs dissimilarity should be retrieved. Only the species information of the assembly is really relevant. Can be `"vert"` for all vertebrate motifs. Only required if for automatic mode (i.e. `k<1`).
`motifs`	a character string specifying the motif set (currently only "jaspar" is supported)
`clusterMethod`	method to be used for motif clustering (currently only `'pam'` (PAM - partitioning around medoids) is supported)

Value

a list structure containing the clustering result

Author(s)

Fabian Mueller

getMotifDistMat *getMotifDistMat*

Description

Retrieve motif dissimilarity/distance matrix for TF motifs

Usage

```
getMotifDistMat(assembly = "hg38", mmObj = NULL, method = "jaspar")
```

Arguments

assembly	genome assembly for which the motifs dissimilarity should be retrieved. Only the species information of the assembly is really relevant. Can be "vert" for all vertebrate motifs.
mmObj	optional motifmatchr object as returned by ChrAccR::prepareMotifmatchr
method	method of dissimilarity quantification. Currently only 'jaspar' (retrieve motif similarities from the annotation of the JASPAR website) is supported.

Value

a matrix of motif DISsimilarities (dist object)

Author(s)

Fabian Mueller

---

getMotifDistMat.jaspar *getMotifDistMat.jaspar*

Description

Retrieve motif a comparison table from JASPAR annotation website and construct a dissimilarity matrix for given motif IDs

Usage

```
getMotifDistMat.jaspar(motifIds = NULL, scoreCol = "Ncor")
```

Arguments

motifIds	string vector of motif ids whose dissimilarities are retrieved
scoreCol	namew of the annotation column in the JASPAR annotation that contains the motif similarity

Value

a matrix of motif DISsimilarities

`getMotifEnrichment,DsATAC-method`
getMotifEnrichment-methods

Description

Perform enrichment analysis for (TF) motifs of a query set of regions. Fisher's Exact Test is employed to test the association of motif present in the query set against the background of all regions of that type covered in the object

Usage

```
## S4 method for signature 'DsATAC'
getMotifEnrichment(.object, type, idx, motifs = "jaspar")
```

Arguments

`.object`	`DsATAC` object
`type`	character string specifying the region type
`idx`	logical vector or indices of the same length as `length(getCoord(.object))` specifies the query set
`motifs`	either a character string (currently only "jaspar" and sets contained in `chromVARmotifs` ("homer", "encode", "cisbp") are supported) or an object containing PWMs that can be used by `motifmatchr::matchMotifs` (such as an `PFMatrixList` or `PWMatrixList` object)

Value

a `data.frame` summarizing Fisher's Exact Test enrichment statistics for each motif

Author(s)

Fabian Mueller

getMotifFootprints,DsATAC-method
getMotifFootprints-methods

Description

Perform enrichment analysis for (TF) motif footprinting

Usage

```
## S4 method for signature 'DsATAC'
getMotifFootprints(
  .object,
  motifNames,
  samples = getSamples(.object),
  motifFlank = 250L,
  type = ".genome",
  motifDb = "jaspar"
)
```

Arguments

.object	DsATAC object
motifNames	character vector of motif names
samples	sample identifiers
motifFlank	number of base pairs flanking the motif on each side
type	(PLACEHOLDER ARGUMENT: NOT IMPLEMENTED YET) character string specifying the region type or ".genome" (default) for genome-wide profiling
motifDb	either a character string (currently only "jaspar" and sets contained in chromVARmotifs ("homer", "encode", "cisbp") are supported) or an object containing PWMs that can be used by motifmatchr::matchMotifs (such as an PFMatrixList or PWMatrixList object) OR a list of GRanges objects specifying motif occurrences

Value

a list of footprinting results with one element for each motif. Each motif's results contain summary data frames with aggregated counts across all motif occurrences and a ggplot object for plotting footprints

Author(s)

Fabian Mueller

Examples

```
## Not run:
dsa <- ChrAccRex::loadExample("dsAtac_ia_example")
motifNames <- c("MA1419.1_IRF4", "MA0139.1_CTCF", "MA0037.3_GATA3")
# motifNames <- grep("(IRF4|CTCF|GATA3)$", names(prepareMotifmatchr("hg38", "jaspar")$motifs), value=TRUE, ignore
samples <- c("TeffNaive_U_1001", "TeffNaive_U_1002", "TeffMem_U_1001", "TeffMem_U_1002")
fps <- getMotifFootprints(dsa, motifNames, samples)
fps[["MA1419.1_IRF4"]]$plot

## End(Not run)
```

`getMotifOccurrences` *getMotifOccurrences*

Description

Find occurrences of motifs in a given genome

Usage

```
getMotifOccurrences(motifNames = NULL, motifDb = "jaspar", genome = "hg38")
```

Arguments

`motifNames`	character vector of motif names
`motifDb`	either a character string (currently only "jaspar" and sets contained in `chromVARmotifs` ("homer", "encode", "cisbp") are supported) or an object containing PWMs that can be used by `motifmatchr::matchMotifs` (such as an `PFMatrixList` or `PWMatrixList` object)
`genome`	character string specifying genome assembly

Value

a `GenomicRangesList` containing motif occurrences

Author(s)

Fabian Mueller

getNonOverlappingByScore

getNonOverlappingByScore

Description

Retrieve the set of non-verlapping regions by iteratively picking the region with maximum score for each set of consecutively overlapping regions

Usage

```
getNonOverlappingByScore(gr, scoreCol = "score")
```

Arguments

gr	GRanges object
scoreCol	name of the column to be used as score in the elementMetadata of the gr object

Value

GRanges object containing non-overlapping regions

Author(s)

Fabian Mueller

getNRegions,DsAcc-method

getNRegions-methods

Description

Return the number of regions of a given type

Usage

```
## S4 method for signature 'DsAcc'
getNRegions(.object, type = "sites")
```

Arguments

.object	DsAcc object
type	character string specifying the rgion type or "sites" (default)

Value

the number of regions of that type

Author(s)

Fabian Mueller

`getPeakSet.snakeATAC` *getPeakSet.snakeATAC*

Description

Retrieve a consensus set of ATAC peaks from the snakeATAC pipline run

Usage

```
getPeakSet.snakeATAC(
  sampleAnnot,
  filePrefixCol,
  genome,
  dataDir,
  sampleIdCol = filePrefixCol,
  type = "summits_no_fw",
  unifWidth = 500L,
  replicateCol = NA,
  replicatePercReq = 1,
  replicateConsSelect = FALSE,
  keepOvInfo = FALSE
)
```

Arguments

`sampleAnnot`	data.frame specifying the sample annotation table
`filePrefixCol`	column name specifying the file prefix for each sample in the sample annotation table
`genome`	genome assembly
`dataDir`	directory where the files are located
`sampleIdCol`	column name or index in the sample annotation table containing unique sample identifiers
`type`	input data type. Currently only "summits_no_fw" (non-overlapping, fixed-width peaks deduced from summits)
`unifWidth`	width of the peaks if the results have uniform peak lengths
`replicateCol`	column name specifying the replicate group for cross-checking coverage across replicates

`replicatePercReq`
: percentile of replicates in a group required to contain a peak in order to keep it. E.g. a value of 1 (default) means that all replicates in a group are required to contain that peak in order to keep it.

`replicateConsSelect`
: if set, the peak set will also be checked for consistency, i.e. in order to retain a peak it has to be consistently be present or absent in each replicate group (as specified in `replicatePercReq` percent of samples)

`keepOvInfo`
: keep annotation columns in the elementMetadata of the results specifying whether a consensus peak overlaps with a peak in each sample

Value

`GRanges` object containing consensus peak set

Author(s)

Fabian Mueller

`getQuickTssEnrichment,DsATAC-method`
getQuickTssEnrichment-methods

Description

[Experimental] Quick, heuristic version of TSS enrichment to just get scores for each sample in the dataset. Useful, e.g. for single cells.

Usage

```
## S4 method for signature 'DsATAC'
getQuickTssEnrichment(
  .object,
  tssGr = NULL,
  sampleIds = getSamples(.object),
  tssW = 201L,
  distBg = 1900L
)
```

Arguments

`.object`	`DsATAC` object
`tssGr`	`GRanges` object containing TSS coordinates
`sampleIds`	sampleIds for which TSS enrichment should be computed
`tssW`	width to consider arount the TSS
`distBg`	number of bases flanking each TSS that will be added on each side

Details

Computes TSS enrichment as the ratio of total insertion sites at a window (of width `tssW` bp) directly at the TSS and 2 background regions symmetrically located (`distBg` bp) upstream and downstream of the TSS

Value

a vector of TSS enrichment values for each sample/cell in the dataset

Author(s)

Fabian Mueller

`getRegionMapping,DsNOMe-method`
getRegionMapping-methods

Description

Retrieve a mapping from regions to GC indices in the dataset

Usage

```
## S4 method for signature 'DsNOMe'
getRegionMapping(.object, type)
```

Arguments

`.object`	`DsNOMe` object
`type`	character string specifying a name for the region type

Value

list containing vectors of indices of GCs for each region of the specified type

Author(s)

Fabian Mueller

`getRegionTypes,DsAcc-method`
getRegionTypes-methods

Description

Return sample IDs in a dataset

Usage

```
## S4 method for signature 'DsAcc'
getRegionTypes(.object, inclSites = FALSE)
```

Arguments

`.object`	`DsAcc` object
`inclSites`	include `"sites"` in the result

Value

Character vector of sample IDs in the dataset

Author(s)

Fabian Mueller

`getSampleAnnot,DsAcc-method`
getSampleAnnot-methods

Description

Return sample annotation table of a dataset

Usage

```
## S4 method for signature 'DsAcc'
getSampleAnnot(.object)
```

Arguments

`.object`	`DsAcc` object

Value

`data.frame` containing sample annotation

Author(s)

Fabian Mueller

`getSampleMetrics.snakeATAC` *getSampleMetrics.snakeATAC*

Description

Retrieve sample summary statistics from the output a snakeATAC pipline run

Usage

```
getSampleMetrics.snakeATAC(sampleAnnot, snakeDir, withPeaks = TRUE)
```

Arguments

`sampleAnnot`	data.frame specifying the sample annotation table. Must have valid rownames corresponding to the sample ids used in the snakeAtac filenames
`snakeDir`	snakeATAC base directory (where the files are located)
`withPeaks`	flag indicating whether to output peak statistics

Value

`data.frame` containing sample summary statistics. the original sample annotation table will be appended to the summary output

Author(s)

Fabian Mueller

`getSamples,DsAcc-method` *getSamples-methods*

Description

Return sample IDs in a dataset

Usage

```
## S4 method for signature 'DsAcc'
getSamples(.object)
```

Arguments

.object	DsAcc object

Value

Character vector of sample IDs in the dataset

Author(s)

Fabian Mueller

---

getScQcStatsTab,DsATACsc-method
getScQcStatsTab-methods

Description

Retrieve a table of QC statistics for single cells

Usage

```
## S4 method for signature 'DsATACsc'
getScQcStatsTab(.object)
```

Arguments

.object	DsATACsc object

Value

an data.frame contain QC statistics for each cell

Author(s)

Fabian Mueller

getTfAnnot *getTfAnnot*

Description

retrieve TF annotation data

Usage

```
getTfAnnot(type = "humantfs")
```

Arguments

type	annotation type. Currently only "humantfs" (pulls info from humantfs.ccbr.utoronto.ca) is supported

Value

a data frame of TF annotation

Author(s)

Fabian Mueller

getTssEnrichment,DsATAC-method *getTssEnrichment-methods*

Description

Get TSS enrichment data and plot

Usage

```
## S4 method for signature 'DsATAC'
getTssEnrichment(
  .object,
  sampleId,
  tssGr = NULL,
  flank = 2000L,
  normTailW = 100L,
  smoothW = 25L,
  silent = FALSE
)
```

Arguments

.object	DsATAC object
sampleId	sample to be plotted
tssGr	GRanges object containing TSS coordinates or NULL to get default set from annotation package
flank	number of bases flanking each TSS that will be added on each side
normTailW	number of bases on each side whose counts will be used to normalize the data
smoothW	radius of the window (in bp) that will be used to smooth the data, i.e. the total width of the smoothing window will be twice that number
silent	limit log messages

Value

a list containing TSS enrichment data and a ggplot object containing TSS enrichment plot

Author(s)

Fabian Mueller

---

getTssEnrichmentBatch,DsATAC-method
getTssEnrichmentBatch-methods

Description

Get TSS enrichment data and plot

Usage

```
## S4 method for signature 'DsATAC'
getTssEnrichmentBatch(
  .object,
  tssGr = NULL,
  sampleIds = getSamples(.object),
  tssW = 201L,
  flank = 2000L,
  normTailW = 200L,
  smoothW = 51L
)
```

Arguments

.object	DsATAC object
tssGr	GRanges object containing TSS coordinates or NULL to get default set from annotation package
sampleIds	sampleIds for which TSS enrichment should be computed
tssW	size of the core TSS window
flank	number of bases flanking each TSS that will be added on each side
normTailW	number of bases on each side whose counts will be used to normalize the data
smoothW	diameter of the window (in bp) that will be used to smooth the data

Value

a list containing TSS enrichment data

Author(s)

Fabian Mueller

---

hmSeqLogo *hmSeqLogo*

Description

Draw a sequence motif logo in a Complex Heatmap using grid. adapted from seqLogo::seqLogo()

Usage

```
hmSeqLogo(
  pwm,
  x = unit(0.5, "npc"),
  y = unit(0.5, "npc"),
  width = 1,
  height = 1,
  ic.scale = TRUE
)
```

Arguments

pwm	PWM (from TFBSTools package)
x	x center coordinate where the motif should be drawn
y	y center coordinate where the motif should be drawn
width	drawing width
height	drawing height
ic.scale	logical If TRUE, the height of each column is proportional to its information content. Otherwise, all columns have the same height.

Value

Draws the motif

Author(s)

Fabian Mueller

Examples

```
## Not run:
mm <- prepareMotifmatchr("hg38", "jaspar")$motifs[["MA0137.3_STAT1"]]
hmSeqLogo(mm, unit(0.5, "npc"), unit(0.5, "npc"), 0.5, 0.5, ic.scale=TRUE)

## End(Not run)
```

`isCanonicalChrom` *isCanonicalChrom*

Description

for a character string of chromosome names, determine if it is a canonical chromosome (i.e. not not ChrUn*, *_random, ...)

Usage

```
isCanonicalChrom(ss)
```

Arguments

`ss`	character vector of chromosome names

Value

logical vector stating whether the given chromosome names correspond to canonical chromosomes

Author(s)

Fabian Mueller

iterativeLSI,DsATACsc-method
iterativeLSI-methods

Description

Perform iterative LSI clustering and dimension reduction as described in doi:10.1038/s41587-019-0332-7

Usage

```
## S4 method for signature 'DsATACsc'
iterativeLSI(
  .object,
  it0regionType = "t5k",
  it0nMostAcc = 20000L,
  it0pcs = 1:25,
  it0clusterResolution = 0.8,
  it0clusterMinCells = 200L,
  it0nTopPeaksPerCluster = 2e+05,
  it1pcs = 1:50,
  it1clusterResolution = 0.8,
  it1mostVarPeaks = 50000L,
  it2pcs = 1:50,
  it2clusterResolution = 0.8,
  rmDepthCor = 0.5,
  normPcs = FALSE,
  umapParams = list(distMethod = "euclidean", min_dist = 0.5, n_neighbors = 25)
)
```

Arguments

`.object`	DsATACsc object
`it0regionType`	character string specifying the region type to start with
`it0nMostAcc`	the number of the most accessible regions to consider in iteration 0
`it0pcs`	the principal components to consider in iteration 0
`it0clusterResolution`	resolution paramter for Seurat's clustering (`Seurat::FindClusters`) in iteration 0
`it0clusterMinCells`	the minimum number of cells in a cluster in order for it to be considered in peak calling (iteration 0)
`it0nTopPeaksPerCluster`	the number of best peaks to be considered for each cluster in the merged peak set (iteration 0)

`it1pcs` the principal components to consider in iteration 0

`it1clusterResolution`
: resolution paramter for Seurat's clustering (`Seurat::FindClusters`) in iteration 1

`it1mostVarPeaks`
: the number of the most variable peaks to consider after iteration 1

`it2pcs` the principal components to consider in the final iteration (2)

`it2clusterResolution`
: resolution paramter for Seurat's clustering (`Seurat::FindClusters`) in the final iteration (2)

`rmDepthCor` correlation cutoff to be used to discard principal components associated with fragment depth (all iterationa)

`normPcs` flag indicating whether to apply z-score normalization to PCs for each cell (all iterations)

`umapParams` parameters to compute UMAP coordinates (passed on to `muRtools::getDimRedCoords.umap` and further to `uwot::umap`)

Details

In order to obtain a low dimensional representation of single-cell ATAC datasets in terms of principal components and UMAP coordinates, we recommend an iterative application of the Latent Semantic Indexing approach [10.1016/j.cell.2018.06.052] described in [doi:10.1038/s41587-019-0332-7]. This approach also identifies cell clusters and a peak set that represents a consensus peak set of cluster peaks in a given dataset. In brief, in an initial iteration clusters are identified based on the most accessible regions (e.g. genomic tiling regions). Here, the counts are first normalized using the term frequency–inverse document frequency (TF-IDF) transformation and singular values are computed based on these normalized counts in selected regions (i.e. the most accessible regions in the initial iteration). Clusters are identified based on the singular values using Louvain clustering (as implemented in the `Seurat` package). Peak calling is then performed on the aggregated insertion sites from all cells of each cluster (using MACS2) and a union/consensus set of peaks uniform-length non-overlapping peaks is selected. In a second iteration, the peak regions whose TF-IDF-normalized counts which exhibit the most variability across the initial clusters provide the basis for a refined clustering using derived singular values. In the final iteration, the most variable peaks across the refined clusters are identified as the final peak set and singular values are computed again. Based on these final singular values UMAP coordinates are computed for low-dimensional projection.

The output object includes the final singular values/principal components (`result$pcaCoord`), the low-dimensional coordinates (`result$umapCoord`), the final cluster assignment of all cells (`result$clustAss`), the complete, unfiltered initial cluster peak set (`result$clusterPeaks_unfiltered`) as well as the final cluster-variable peak set (`result$regionGr`).

Value

an `S3` object containing dimensionality reduction results, peak sets and clustering

Author(s)

Fabian Mueller

join,DsATAC-method *join-methods*

Description

Combine two DsATAC objects

Usage

```
## S4 method for signature 'DsATAC'
join(.object, objectB, joinRegionTypes = "union")
```

Arguments

.object	DsATAC object
objectB	DsATAC object
combineRegionTypes	how to combine region types: 'union' (default): the resulting object will have counts aggregated over region types from both objects. 'intersect': only region types present in both objects will occur in the output

Value

a new DsATAC object combining both input objects. It contains untransformed counts.

Author(s)

Fabian Mueller

length,DsAcc-method *Retrieve the number of samples contained in a DsAcc object*

Description

Retrieve the number of samples contained in a DsAcc object

Usage

```
## S4 method for signature 'DsAcc'
length(x)
```

Arguments

x	DsAcc object

loadConfig *loadConfig*

Description

Sets the configuration from a configuration file (JSON)

Usage

```
loadConfig(cfgFile)
```

Arguments

cfgFile	Config file in JSON format. As output by saveConfig

Value

nothing of particular interest. The configuration is set for the current environment

Author(s)

Fabian Mueller

loadDsAcc *loadDsAcc*

Description

Load a DsAcc dataset from disk

Usage

```
loadDsAcc(path)
```

Arguments

path	Location of saved DsAcc object

Value

DsAcc object

Author(s)

Fabian Mueller

maskMethNA,DsNOMe-method
maskMethNA-methods

Description

Set the indices specified in a mask to NA

Usage

```
## S4 method for signature 'DsNOMe'
maskMethNA(.object, mask, type = "sites", reaggregate = TRUE)
```

Arguments

.object	DsNOMe object
mask	a mask, i.e. a logical matrix of indices to set to NA
type	character string specifying a name for the region type (default: sites)
reaggregate	redo region aggregation (only has an effect if type is sites and there are aggregated regions in the dataset)

Value

a new DsNOMe object with sites/regions masked

Author(s)

Fabian Mueller

mergePseudoBulk,DsATACsc-method
mergePseudoBulk-methods

Description

Merge cells into pseudobulk samples based on annotation

Usage

```
## S4 method for signature 'DsATACsc'
mergePseudoBulk(.object, mergeGroups, cleanSampleAnnot = TRUE)
```

Arguments

.object	DsATACsc object
mergeGroups	factor or character vector or column name in sample annotation table. Can alternatively be a (named) list containing sample indices or names for each group to merge.
cleanSampleAnnot	clean up sample annotation table in the new object

Value

a new DsATAC object with cells merged into pseudobulk samples

Author(s)

Fabian Mueller

---

mergeSamples,DsATAC-method

mergeSamples-methods

Description

Merge signal and insertion data across samples

Usage

```
## S4 method for signature 'DsATAC'
mergeSamples(.object, mergeGroups, countAggrFun = "sum")
```

Arguments

.object	DsATAC object
mergeGroups	factor or character vector or column name in sample annotation table. Can alternatively be a (named) list containing sample indices or names for each group to merge.
countAggrFun	aggregation function for signal counts. Currently sum (default), mean and median are supported.

Value

a new DsATAC object with samples merged

Author(s)

Fabian Mueller

`mergeStrands,DsNOMe-method`
mergeStrands-methods

Description

Merge + and - strands of the dataset by adding read coverage and recomputing Methylation levels

Usage

```
## S4 method for signature 'DsNOMe'
mergeStrands(.object, reaggregate = TRUE)
```

Arguments

`.object`	DsNOMe object
`reaggregate`	redo region aggregation (only has an effect if there are aggregated regions in the dataset)

Value

a new DsNOMe object with the strands merged

Author(s)

Fabian Mueller

`normalizeMeth,DsNOMe-method`
normalizeMeth-methods

Description

Normalize methylation levels

Usage

```
## S4 method for signature 'DsNOMe'
normalizeMeth(.object, type = "sites", method = "quantile", reaggregate = TRUE)
```

Arguments

`.object`	DsNOMe object
`type`	character string specifying a name for the region type (default: sites)
`method`	normalization method to be applied. Currently only 'quantile' is supported
`reaggregate`	redo region aggregation (only has an effect if type is sites and there are aggregated regions in the dataset)

Value

a new DsNOMe object with normalized methylation levels

Author(s)

Fabian Mueller

`plotInsertSizeDistribution,DsATAC-method` *plotInsertSizeDistribution-methods*

Description

Plot insert size distribution

Usage

```
## S4 method for signature 'DsATAC'
plotInsertSizeDistribution(.object, sampleId)
```

Arguments

`.object`	DsATAC object
`sampleId`	sample to be plotted

Value

`ggplot` object containing insert size distribution plot

Author(s)

Fabian Mueller

`prepareMotifmatchr` *prepareMotifmatchr*

Description

prepare objects for a `motifmatchr` analysis

Usage

```
prepareMotifmatchr(genome, motifs)
```

Arguments

genome	character string specifying genome assembly
motifs	either a character string (currently only "jaspar" and sets contained in chromVARmotifs ("homer", "encode", "cisbp") are supported) or an object containing PWMs that can be used by motifmatchr::matchMotifs (such as an PFMatrixList or PWMatrixList object)

Value

a list containing objects to be used as arguments for motifmatchr

Author(s)

Fabian Mueller

---

projectMatrix_UMAP *projectMatrix_UMAP*

Description

given a (count) matrix and dimension reduction result, return the projected UMAP coordinates in the embedding space

Usage

```
projectMatrix_UMAP(X, umapObj, binarize = TRUE, addPcCoord = FALSE)
```

Arguments

X	matrix to be projected (features X samples)
umapObj	dimension reduction result as returned by dimRed_UMAP
binarize	binarize the counts before projecting
addPcCoord	also add PC coordinates to the resulting matrix

Value

Projected UMAP coordinates

Author(s)

Fabian Mueller

PWMatrixToProbMatrix *PWMatrixToProbMatrix*

Description

convert a log2probratio PWM (`PWMatrix` from TFBSTools package) to a matrix containing probabilities in [0,1]

Usage

```
PWMatrixToProbMatrix(x)
```

Arguments

`x`	log2probratio PWM (`PWMatrix` from TFBSTools package)

Value

PWM probability matrix with values in

Author(s)

Fabian Mueller

readMACS2peakFile *readMACS2peakFile*

Description

Reads the MACS2 ouput as GRanges

Usage

```
readMACS2peakFile(fn)
```

Arguments

`fn`	Filename for MACS2 narrow peak file

Value

`GRanges` object containing peak information

Author(s)

Fabian Mueller

regionAggregation,DsATAC-method
regionAggregation-methods

Description

Aggregate signal counts across a set of regions

Usage

```
## S4 method for signature 'DsATAC'
regionAggregation(
  .object,
  regGr,
  type,
  signal = NULL,
  aggrFun = "median",
  dropEmpty = TRUE,
  bySample = TRUE,
  chunkSize = 5000L
)
```

Arguments

.object	DsATAC object
regGr	GRanges object containing regions to summarize
type	character string specifying a name for the region type
signal	character string specifying a name for the region type for the signal to be aggregated If it is NULL (default), the new region type will be initialized with NA values. If it is "insertions" count data will be initialized from insertion sites (if fragment data is present in the object).
aggrFun	aggregation function for signal counts. Will only be used if signal!="insertions" Currently sum, mean and median (default) are supported.
dropEmpty	discard all regions with no observed signal counts
bySample	[only relevant if signal=="insertions"]. Process data sample-by-sample to save memory.
chunkSize	[only relevant if signal=="insertions" & !bySample] number of samples to process per chunk (saves memory). If NULL or larger than the number of samples, only one chunk will be processed.

Value

a new DsATAC object with aggregated signal counts per regions

Author(s)

Fabian Mueller

`regionAggregation,DsNOMe-method`
regionAggregation-methods

Description

Aggregate methylation levels and coverage values accross a set of regions

Usage

```
## S4 method for signature 'DsNOMe'
regionAggregation(
  .object,
  regGr,
  type,
  methAggrFun = "weightedMean",
  dropEmpty = TRUE
)
```

Arguments

`.object`	`DsNOMe` object
`regGr`	`GRanges` object containing regions to summarize
`type`	character string specifying a name for the region type
`methAggrFun`	aggregation function for methylation levels. Currently `mean`, `median` and `weightedMean` (default) are supported.
`dropEmpty`	discard all regions with no observed methylation levels

Details

Coverage values are aggregated by summing up coverage values for individual GCs while the aggregation function for methylation levels is specified by the `methAggrFun` parameter.

Value

a new `DsNOMe` object with aggregated regions

Author(s)

Fabian Mueller

regionSetCounts,DsATAC-method
regionSetCounts-methods

Description

Overlap the insertion data with a list of region sets

Usage

```
## S4 method for signature 'DsATAC'
regionSetCounts(.object, rsl, bySample = FALSE)
```

Arguments

.object	DsATAC object
rsl	GRangesList or NAMED list of GRanges objects. Each element corresponds to a region set for which the summary statistics are reported
bySample	for internal use: iterate over samples (instead of retrieving one giant insertion list for all samples) in order to save memory (at the tradeoff of compute time)

Value

a matrix of overlap counts for each region set and sample

Author(s)

Fabian Mueller

removeFragmentData,DsATAC-method
removeFragmentData-methods

Description

Removes fragment data from DsATAC object (e.g. to save space)

Usage

```
## S4 method for signature 'DsATAC'
removeFragmentData(object)
```

Arguments

object	DsATAC object

Value

the modified object (without fragment data)

Author(s)

Fabian Mueller

`removeRegionData,DsATAC-method`
removeRegionData-methods

Description

Remove all region data from a `DsATAC` object

Usage

```
## S4 method for signature 'DsATAC'
removeRegionData(.object)
```

Arguments

`.object`	`DsATAC` object

Value

a new `DsATAC` object with region data removed

Author(s)

Fabian Mueller

`removeRegions,DsAcc-method`
removeRegions-methods

Description

Remove the specified sites or regions from an object

Usage

```
## S4 method for signature 'DsAcc'
removeRegions(.object, indices, type)
```

Arguments

.object	DsAcc object
indices	a vector of indices of sites/regions to be removed. Can be numeric, integer or logical.
type	character string specifying a name for the region type (sefault: sites)

Value

a new DsAcc object with sites/regions removed

Author(s)

Fabian Mueller

---

removeRegions,DsATAC-method
removeRegions-methods

Description

Remove the specified sites or regions from an object

Usage

```
## S4 method for signature 'DsATAC'
removeRegions(.object, indices, type)
```

Arguments

.object	DsATAC object
indices	a vector of indices of sites/regions to be removed. Can be numeric, integer or logical.
type	character string specifying a name for the region type (sefault: sites)

Value

a new DsATAC object with sites/regions removed

Author(s)

Fabian Mueller

`removeRegions,DsNOMe-method`
removeRegions-methods

Description

Remove the specified sites or regions from an object

Usage

```
## S4 method for signature 'DsNOMe'
removeRegions(.object, indices, type = "sites", reaggregate = TRUE)
```

Arguments

`.object`	DsNOMe object
`indices`	a vector of indices of sites/regions to be removed. Can be numeric, integer or logical.
`type`	character string specifying a name for the region type (sefault: sites)
`reaggregate`	redo region aggregation (only has an effect if type is sites and there are aggregated regions in the dataset)

Value

a new DsNOMe object with sites/regions removed

Author(s)

Fabian Mueller

`removeRegionType,DsATAC-method`
removeRegionType-methods

Description

Remove the specified region type from an object

Usage

```
## S4 method for signature 'DsATAC'
removeRegionType(.object, type)
```

Arguments

.object	DsATAC object
type	character string specifying a name for the region type (sefault: sites)

Value

a new DsATAC object with the region type removed

Author(s)

Fabian Mueller

---

removeSamples,DsATAC-method
removeSamples-methods

Description

Remove samples from a DsATAC object

Usage

```
## S4 method for signature 'DsATAC'
removeSamples(.object, indices)
```

Arguments

.object	DsATAC object
indices	a vector of indices of samples to be removed. Can be numeric, integer or logical.

Value

a new DsATAC object with sites/regions removed

Author(s)

Fabian Mueller

`rowZscores` *rowZscores*

Description

Performs z-score normalization on the rows of a matrix. (Basically a wrapper around `matrixStats`)

Usage

```
rowZscores(X, na.rm = FALSE)
```

Arguments

`X`	input matrix
`na.rm`	should NAs be omitted?

Value

z-score normalized matrix

Author(s)

Fabian Mueller

`run_atac` *run_atac*

Description

Run the complete ChrAccR analysis for ATAC-seq data

Usage

```
run_atac(
  anaDir,
  input = NULL,
  sampleAnnot = NULL,
  genome = NULL,
  sampleIdCol = NULL,
  regionSets = NULL,
  startStage = "raw",
  resetStage = NULL
)
```

Arguments

anaDir	analysis directory
input	Input object. Can be either NULL, a character string, a DsATAC. Set to NULL when you want to continue a previous analysis
sampleAnnot	sample annotation table (data.frame) or NULL if continuing existing analysis or input is a DsATAC object
genome	genome assembly. Only relevant if not continuing existing analysis and input is not a DsATAC object
sampleIdCol	column name in the sample annotation table containing unique sample Only relevant if not continuing existing analysis and input is not a DsATAC object
regionSets	a list of GRanges objects which contain region sets over which count data will be aggregated. Only relevant if not continuing existing analysis and input is not a DsATAC object
startStage	stage where to start the analysis from. can be one of "raw", "filtered", "processed". Only relevant if not continuing existing analysis.
resetStage	flag indicating whether to reset the analysis directory (i.e. deleting previously generated reports and datasets), when continuing previous analyses (input argument is NULL).

Value

DsATAC object (invisible)

Author(s)

Fabian Mueller

---

run_atac_chromvar *run_atac_chromvar*

Description

Run chromVAR analysis for ATAC-seq data

Usage

```
run_atac_chromvar(.object)
```

Arguments

.object	DsATAC object

Value

An S3 object containing a list of chromVAR Deviations objects as returned by chromVAR::computeDeviations. One object for each region type specified in the chromVarRegionTypes configuration.

Author(s)

Fabian Mueller

`run_atac_differential` *run_atac_differential*

Description

Run differential analyses for ATAC-seq data

Usage

```
run_atac_differential(dsa, anaDir, chromVarObj = NULL)
```

Arguments

`dsa`	`DsATAC` object
`anaDir`	analysis directory
`chromVarObj`	[optional] pre-computed result of a call to `run_atac_chromvar(...)`

Value

`S3` object containing differential analysis results and an analysis report object

Author(s)

Fabian Mueller

`run_atac_exploratory` *run_atac_exploratory*

Description

Run exploratory analyses for ATAC-seq data

Usage

```
run_atac_exploratory(
  dsa,
  anaDir,
  chromVarObj = NULL,
  itLsiObj = NULL,
  geneActSe = NULL
)
```

Arguments

dsa	DsATAC object
anaDir	analysis directory
chromVarObj	[optional] pre-computed result of a call to run_atac_chromvar(...)
itLsiObj	[for single-cell only; optional] pre-computed result of a call to iterativeLSI(.object, ...)
geneActSe	[for single-cell only; optional] pre-computed result of a call to getCiceroGeneActivities(.object, ...)

Value

S3 object containing exploratory metrics and an analysis report object

Author(s)

Fabian Mueller

---

run_atac_filtering *run_atac_filtering*

Description

Run the filtering for ATAC-seq data

Usage

```
run_atac_filtering(dsa, anaDir)
```

Arguments

dsa	DsATAC object
anaDir	analysis directory

Value

S3 object containing the filtered DsATAC object, filtering statistics and an analysis report object

Author(s)

Fabian Mueller

run_atac_normalization *run_atac_normalization*

Description

Run count normalization for ATAC-seq data

Usage

```
run_atac_normalization(dsa, anaDir)
```

Arguments

dsa	DsATAC object
anaDir	analysis directory

Value

S3 object containing the normalized DsATAC object and an analysis report object

Author(s)

Fabian Mueller

run_atac_peakcalling *run_atac_peakcalling*

Description

Run peak calling for ATAC-seq data

Usage

```
run_atac_peakcalling(dsa, anaDir)
```

Arguments

dsa	DsATAC object
anaDir	analysis directory

Value

S3 object containing the annotated DsATAC object, per-sample peak calls, a consensus peak set and an analysis report object

Author(s)

Fabian Mueller

`run_atac_qc` *run_atac_qc*

Description

Run the summary QC analysis for ATAC-seq data

Usage

```
run_atac_qc(dsa, anaDir)
```

Arguments

`dsa`	`DsATAC` object
`anaDir`	analysis directory

Value

`S3` object containing QC statistics and an analysis report object

Author(s)

Fabian Mueller

`run_atac_sc_unsupervised` *run_atac_sc_unsupervised*

Description

Run unsupervised analysis for single-cell ATAC-seq data (i.e. iterative LSI, clustering and cluster peak detection)

Usage

```
run_atac_sc_unsupervised(dsa, anaDir)
```

Arguments

`dsa`	`DsATAC` object
`anaDir`	analysis directory

Value

S3 object containing the annoted DsATAC object, the results of running iterative LSI and an analysis report object

Author(s)

Fabian Mueller

`safeMatrixStats` *safeMatrixStats*

Description

Compute matrix statistics selecting the appropriate function depending on the matrix class of the input (supports sparse matrices and DelayedArrays)

Usage

```
safeMatrixStats(X, statFun = "rowSums", ...)
```

Arguments

`X`	input matrix
`statFun`	statistic. E.g. `"rowSums"`, `"colSums"`, `"rowMeans"`, `"colMeans"`, ...
`...`	arguments passed on to the matrix stats function. E.g. `na.rm`.

Value

result of the corresponding matrix statistic

Author(s)

Fabian Mueller

samplePseudoBulk,DsATACsc-method
samplePseudoBulk-methods

Description

Samples pseudo-bulk samples from single-cells

Usage

```
## S4 method for signature 'DsATACsc'
samplePseudoBulk(.object, nnData, nSamples, nCellsPerSample = 100)
```

Arguments

.object	DsATACsc object
nnData	Data to use for nearest neighbor matching. Can either be the name of a region type in .object or a data matrix with the same number of rows as .object has cells.
nSamples	number of pseudobulk samples to be returned
nCellsPerSample	number of cells to be aggregated per sample

Details

Samples pseudo-bulk samples from single-cells by sampling nSamples individual cells and then merging it with its nCellsPerSample - 1 nearest neighbors (according to nnData).

Value

S3 data structure containing a list of sampling results as well as a DsATAC object containing pseudo-bulk aggregates

Author(s)

Fabian Mueller

saveConfig *saveConfig*

Description

Save the current configuration to a configuration file (JSON)

Usage

```
saveConfig(dest)
```

Arguments

dest	Filename for the config file in JSON format.

Value

nothing of particular interest.

Author(s)

Fabian Mueller

saveDsAcc *saveDsAcc*

Description

Save a DsAcc dataset to disk for later loading

Usage

```
saveDsAcc(.object, path, forceDiskDump = FALSE, updateDiskRef = TRUE)
```

Arguments

.object	DsAcc object
path	destination to save the object to
forceDiskDump	force large matrices (counts) to be stored as HDF5 (even when the object was not created using diskDump=TRUE)
updateDiskRef	update disk dumped (HDF5) references (e.g. for count data)

Value

(invisibly) The object (with potentially updated disk dumped references)

Author(s)

Fabian Mueller

`setConfigElement` *setConfigElement*

Description

Set a configuration item to a given value

Usage

```
setConfigElement(name, value)
```

Arguments

`name`	name of the config item
`value`	value of the config item

Value

nothing of particular interest.

Options used by the package

`tmpDir = tempdir()` Directory for temporary files. Must be existing.

`cleanMem = TRUE` During runtime, regularly clean-out the memory in order to reduce memory overuse

`colorSchemes` named `list` of DISCRETE color schemes to be used for plotting. Each element should be a named vector specifying colors for groups/annotations.

`colorSchemesCont` named `list` of CONTINOUS color schemes to be used for plotting. Each element should be a vector specifying a range of colors.

`geneModelVersions` Gene model versions to be used for various genomes

`analysisName = "ChrAccR analysis"` A title for the analysis (a string).

`regionTypes` Region types to be used in the analysis

`chromVarRegionTypes = NULL` Region types to be used for chromVar analysis. If `NULL` (default), ChrAccR will automatically look for region types with the keyword `"peak"` in their name.

`chromVarMotifs = "jaspar_vert"` Character vector of names of TF motif sets to be used in ChromVAR analyses. By default the vertebrate set of the JASPAR database will be used.

`chromVarMotifNamesForDimRed` Names of motifs to be used for dimension reduction plots in the reports. [only relevant for single-cell data]

`genesOfInterest` Names of genes of interest to be highlighted in the reports (e.g. dimension reduction) in the reports. [currently only relevant for single-cell data and only when `scGeneActivity` is activated]

annotationColumns Sample annotation columns to be used for reporting

annotationMinGroupSize = 2 Minimum size of a group to be used in the reports. Influences which annotation columns are automatically selected for reporting.

annotationMaxGroupCount = NULL Maximum number of groups to be used in the reports. Influences which annotation columns are automatically selected for reporting. If NULL (default) it will effectively be the number of samples - 1.

doPeakCalling = FALSE Perform per-sample peak calling and retrieve consensus peak set. Requires that macs2 is installed and can be called from the command line. [for bulk data analysis only]

peakCallingProfile = NULL If set to a string describing a valid profile, will apply a special profile for macs2 peak calling. [only valid in combination with the doPeakCalling option]

annotationPeakGroupColumn Annotation column to base the consensus peak set replication filtering on.

annotationPeakGroupAgreePerc = 1.0 Percent of samples that have to agree to identify consensus peaks. See getConsensusPeakSet for details.

filteringCovgCount = 1L Minimum insertion count to filter count matrices by. See filterLowCovg,DsATAC-method for details. [for bulk data analysis only]

filteringCovgReqSamples = 0.75 Minimum required samples to apply low coverage filtering to. See filterLowCovg,DsATAC-method for details. [for bulk data analysis only]

filteringSexChroms = FALSE Flag indicating whether to remove sex chromosomes.

filteringScMinFragmentsPerCell = 1000L Minimum number of fragments per cell to retain a cell in the analysis. [for single-cell data analysis only]

filteringScMaxFragmentsPerCell = Inf Maximum number of fragments allowed per cell to retain a cell in the analysis. [for single-cell data analysis only]

filteringScMinTssEnrichment = 6 Minimum TSS enrichment score per cell to retain a cell in the analysis. [for single-cell data analysis only]

normalizationMethod = "quantile" Normalization method to use for count normalization. Allowed methods include the ones listed in transformCounts,DsATAC-method. [for bulk data analysis only]

exploratoryLogNormCounts = TRUE Should a log-normalization be applied in the exploratory plot sections of the reports (dimension reduction, heatmaps)

exploratoryNSubsample = 2e6 Number of regions to subsample in exploratory analysis in order to increase computational performance.

differentialColumns Sample annotation columns to be used for differential testing and reporting

differentialColumns1vsAll Sample annotation columns to be used for differential testing and reporting in a 1-vs-all group setting. Should be a subset of differentialColumns.

differentialCompNames Comparison names from which comparison information is derived. Must be in the format of "$GRP1_NAME vs $GRP2_NAME [$ANNOTATION_COLUMN]".

differentialAdjColumns Sample annotation columns to be adjusted for in differential testing

differentialCutoffL2FC Cutoff on log2 fold-change to be used for reporting differential accessibility.

lolaDbPaths Precomputed LOLA databases to be used for enrichment analysis. If NULL (default), ChrAccR will download an apropriate core database.

`scIterativeLsiRegType` For single-cell analysis only: region type to be used for clustering and dimension reduction using iterative LSI. By default (`NULL`), ChrAccR will look for a region type named `"tiling"`.

`scIterativeLsiParams` Parameters to use for iterative LSI. See `iterativeLSI,DsATACsc-method` for details.

`scGeneActivity = FALSE` For single-cell analysis only: Compute gene activity from accessibility. Possible options are `"RBF"` for radial-basis-function-weighted count aggregation (default when set to `TRUE`) or `"Cicero"` for Cicero correlation-based aggregation

Author(s)

Fabian Mueller

`transformCounts,DsATAC-method`
transformCounts-methods

Description

transform count data for an ATAC seq dataset

Usage

```
## S4 method for signature 'DsATAC'
transformCounts(
  .object,
  method = "quantile",
  regionTypes = getRegionTypes(.object),
  ...
)
```

Arguments

`.object`	`DsATAC` object
`method`	transformation method to be applied. Currently only 'log2', 'log10', 'quantile' (quantile normalization), 'percentile' (percentile normalization),'rankPerc' (rank percentile), 'vst' (DESeq2 Variance Stabilizing Transformation), 'batchCorrect' (limma batch effect removal), tf-idf', 'CPM' (counts per million), and 'RPKM' (RPKM normalization) are supported
`regionTypes`	character vector specifying a name for the region type in which count data should be normalized(default: all region types)
`...`	other arguments depending on the `method` used. For `'batchCorrect'` it should be arguments passed on to `limma::removeBatchEffect` (most importantly, the `batch` argument).

Value

a new DsATAC object with normalized count data

Author(s)

Fabian Mueller

`unsupervisedAnalysisSc,DsATACsc-method`
unsupervisedAnalysisSc-methods

Description

Perform unsupervised analysis on single-cell data. Performs dimensionality reduction and clustering.

Usage

```
## S4 method for signature 'DsATACsc'
unsupervisedAnalysisSc(
  .object,
  regionType,
  regionIdx = NULL,
  dimRedMethod = "tf-idf_irlba",
  usePcs = 1:50,
  clusteringMethod = "seurat_louvain"
)
```

Arguments

`.object`	DsATACsc object
`regionType`	character string specifying the region type
`regionIdx`	indices of regions to be used (logical or integer vector). If `NULL` (default) all regions of the specified regionType will be used.
`dimRedMethod`	character string specifying the dimensionality reduction method. Currently on `"tf-idf_irlba"` is supported
`usePcs`	integer vector specifying the principal components to use for UMAP and clustering
`clusteringMethod`	character string specifying the clustering method. Currently on `"seurat_louvain"` is supported

Value

an `S3` object containing dimensionality reduction results and clustering

Author(s)

Fabian Mueller

`[,DsATAC,ANY,ANY,ANY-method`

Subsetting DsATAC datasets by sample

Description

NOTE: '[' operator for DsATAC does not reorder samples or deal with index multiplicity

Usage

```
## S4 method for signature 'DsATAC,ANY,ANY,ANY'
x[i]
```

Arguments

x	DsATAC object
i	sample names or indices

Index