Package 'ChromSCape'

March 30, 2021

Title Analysis of single-cell epigenomics datasets with a Shiny App

Version 1.0.0

Description ChromSCape - Chromatin landscape profiling for Single Cells - is a ready-to-launch user-friendly Shiny Application for the analysis of single-cell epigenomics datasets (scChIP-seq, scATAC-seq, scCUT&Tag, ...) from aligned data to differential analysis & gene set enrichment analysis. It is highly interactive, enables users to save their analysis and covers a wide range of analytical steps: QC, preprocessing, filtering, batch correction, dimensionality reduction, vizualisation, clustering, differential analysis and gene set analysis.

License GPL-3

biocViews Software, SingleCell, ChIPSeq, ATACSeq, MethylSeq, Classification, Clustering, Epigenetics, PrincipalComponent, SingleCell, ATACSeq, ChIPSeq, Annotation, BatchEffect, MultipleComparison, Normalization, Pathways, Preprocessing, QualityControl, ReportWriting, Visualization, GeneSetEnrichment, DifferentialPeakCalling

VignetteBuilder knitr

URL https://github.com/vallotlab/ChromSCape

BugReports https://github.com/vallotlab/ChromSCape/issues

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown, BiocStyle

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

Imports shiny, colourpicker, shinyjs, rtracklayer, shinyFiles, shinyhelper, shinycssloaders, Matrix, plotly, shinydashboard, colorRamps, kableExtra, viridis, batchelor, BiocParallel, parallel, Rsamtools, ggplot2, qualV, stringdist, fs, DT, scran, scater, ConsensusClusterPlus, Rtsne, dplyr, tidyr, GenomicRanges, IRanges, irlba, rlist, umap, tibble, methods, jsonlite, edgeR, stats, graphics, grDevices, utils, S4Vectors, SingleCellExperiment, SummarizedExperiment, msigdbr

Depends R (>= 4.0)

git_url https://git.bioconductor.org/packages/ChromSCape

git_branch RELEASE_3_12

git_last_commit c84df0d

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

Author Pacome Prompsy [aut, cre] (<https://orcid.org/0000-0003-4375-7583>), Celine Vallot [aut] (<https://orcid.org/0000-0003-1601-2359>)

Maintainer Pacome Prompsy <pacome.prompsy@curie.fr>

R topics documented:

`annotation_from_merged_peaks` *Find nearest peaks of each gene and return refined annotation*

Description

Find nearest peaks of each gene and return refined annotation

Usage

```
annotation_from_merged_peaks(scExp, merged_peaks, geneTSS_annotation)
```

Arguments

`scExp`	A SingleCellExperiment object
`merged_peaks`	A list of GRanges object containing the merged peaks
`geneTSS_annotation`	A GRanges object with reference genes

Value

A data.frame with refined annotation

`annotToCol2` *annotToCol2*

Description

annotToCol2

Usage

```
annotToCol2(
  annotS = NULL,
  annotT = NULL,
  missing = c("", NA),
  anotype = NULL,
  maxnumcateg = 2,
  categCol = NULL,
  quantitCol = NULL,
  plotLegend = TRUE,
  plotLegendFile = NULL
)
```

Arguments

annotS	A color matrix
annotT	A color matrix
missing	Convert missing to NA
anotype	Annotation type
maxnumcateg	Maximum number of categories
categCol	Categorical columns
quantitCol	Quantitative columns
plotLegend	Plot legend ?
plotLegendFile	Which file to plot legend ?

Value

A matrix of continuous or discrete colors

Examples

```
data("scExp")
annotToCol2(SingleCellExperiment::colData(scExp), plotLegend = FALSE)
```

---

anocol_binary *Helper binary column for anocol function*

Description

Helper binary column for anocol function

Usage

```
anocol_binary(anocol, anotype, plotLegend, annotS)
```

Arguments

anocol	The color feature matrix
anotype	The feature types
plotLegend	Plot legend ?
annotS	A color matrix

Value

A color matrix similar to anocol with binrary columns colored

anocol_categorical *Helper binary column for anocol function*

Description

Helper binary column for anocol function

Usage

```
anocol_categorical(anocol, categCol, anotype, plotLegend, annotS)
```

Arguments

anocol	The color feature matrix
categCol	Colors for categorical features
anotype	The feature types
plotLegend	Plot legend ?
annotS	A color matrix

Value

A color matrix similar to anocol with binrary columns colored

bams_to_matrix_indexes *Count bam files on interval to create count indexes*

Description

Count bam files on interval to create count indexes

Usage

```
bams_to_matrix_indexes(files_dir, which)
```

Arguments

files_dir	Directory containing the single cell BAM files
which	Genomic Range on which to count

Value

A list containing a "feature index" data.frame and a count vector for non 0 entries, both used to form the sparse matrix

`beds_to_matrix_indexes` *Count bed files on interval to create count indexes*

Description

Count bed files on interval to create count indexes

Usage

```
beds_to_matrix_indexes(files_dir, which)
```

Arguments

`files_dir`	Directory containing the single cell BAM files
`which`	Genomic Range on which to count

Value

A list containing a "feature index" data.frame and a names of cells as vector both used to form the sparse matrix

`call_macs2_merge_peaks` *Calling MACS2 peak caller and merging resulting peaks*

Description

Calling MACS2 peak caller and merging resulting peaks

Usage

```
call_macs2_merge_peaks(affectation, odir, p.value, ref, peak_distance_to_merge)
```

Arguments

`affectation`	Annotation data.frame with cell cluster and cell id information
`odir`	Output directory to write MACS2 output
`p.value`	P value to detect peaks, passed to MACS2
`ref`	Reference genome
`peak_distance_to_merge`	Distance to merge peaks

Value

A list of merged GRanges peaks

changeRange *changeRange*

Description

changeRange

Usage

```
changeRange(v, newmin = 1, newmax = 10)
```

Arguments

v	A numeric vector
newmin	New min
newmax	New max

Value

A matrix with values scaled between newmin and newmax

---

check_correct_datamatrix *Check if matrix rownames are well formated and correct if needed*

Description

Throws warnings / error if matrix is in the wrong format

Usage

```
check_correct_datamatrix(datamatrix_single, sample_name = "")
```

Arguments

datamatrix_single	A sparse matrix
sample_name	Matrix sample name for warnings

Value

A sparseMatrix in the right rownames format

choose_cluster_scExp *Choose a number of clusters*

Description

This functions takes as input a SingleCellExperiment object with consclust and a number of cluster to select. It outputs a SingleCellExperiment object with each cell assigned to a correlation cluster in colData. Also calculates a hierarchical clustering of the consensus associations calculated by ConsensusClusterPlus.

Usage

```
choose_cluster_scExp(
  scExp,
  nclust = 3,
  consensus = TRUE,
  hc_linkage = "ward.D"
)
```

Arguments

scExp	A SingleCellExperiment object containing consclust in metadata.
nclust	Number of cluster to pick (3)
consensus	Use consensus clustering results instead of simple hierarchical clustering ? (TRUE)
hc_linkage	A linkage method for hierarchical clustering. See cor. ('ward.D')

Value

Returns a SingleCellExperiment object with each cell assigned to a correlation cluster in colData.

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = choose_cluster_scExp(scExp_cf,nclust=3,consensus=FALSE)
table(scExp_cf$cell_cluster)

scExp_cf = consensus_clustering_scExp(scExp)
scExp_cf_consensus = choose_cluster_scExp(scExp_cf,nclust=3,consensus=TRUE)
table(scExp_cf_consensus$cell_cluster)
```

choose_perplexity *Choose perplexity depending on number of cells for Tsne*

Description

Choose perplexity depending on number of cells for Tsne

Usage

```
choose_perplexity(dataset)
```

Arguments

dataset	A matrix of features x cells (rows x columns)

Value

A number between 5 and 30 to use in Rtsne function

col2hex *Col2Hex*

Description

Transform character color to hexadecimal color code.

Usage

```
col2hex(cname)
```

Arguments

cname	Color name

Value

The HEX color code of a particular color

colors_scExp *Adding colors to cells & features*

Description

Adding colors to cells & features

Usage

```
colors_scExp(
  scExp,
  annotCol = "sample_id",
  color_by = "sample_id",
  color_df = NULL
)
```

Arguments

scExp	A SingleCellExperiment Object
annotCol	Column names to color
color_by	If specifying color_df, column names to color
color_df	Color data.frame to specify which color for which condition

Value

A SingleCellExperiment with additionnal "color" columns in colData

Examples

```
data("scExp")
scExp = colors_scExp(scExp,annotCol = c("sample_id",
"total_counts"),
 color_by =  c("sample_id","total_counts"))

#Specific colors using a manually created data.frame :
color_df = data.frame(sample_id=unique(scExp$sample_id),
 sample_id_color=c("red","blue","green","yellow"))
scExp = colors_scExp(scExp,annotCol="sample_id",
color_by="sample_id",color_df=color_df)
```

combine_datamatrix *Combine two matrices and emit warning if no regions are in common*

Description

Combine two matrices and emit warning if no regions are in common

Usage

```
combine_datamatrix(datamatrix, datamatrix_single, file_names, i)
```

Arguments

datamatrix	A sparse matrix or NULL if empty
datamatrix_single	Another sparse matrix
file_names	File name corresponding to the matrix for warnings
i	file number

Value

A combined sparse matrix

---

combine_enrichmentTests

Run enrichment tests and combine into list

Description

Run enrichment tests and combine into list

Usage

```
combine_enrichmentTests(
  diff,
  enrichment_qval,
  qval.th,
  cdiff.th,
  annotFeat_long,
  peak_distance,
  refined_annotation,
  GeneSets,
  GeneSetsDf,
  GenePool
)
```

Arguments

diff	Differential list
enrichment_qval	Adusted p-value threshold above which a pathway is considered significative list
qval.th	Differential analysis adjusted p.value threshold
cdiff.th	Differential analysis log-fold change threshold
annotFeat_long	Long annotation
peak_distance	Maximum gene to peak distance
refined_annotation	Refined annotation data.frame if peak calling is done

GeneSets	List of pathways
GeneSetsDf	Data.frame of pathways
GenePool	Pool of possible genes for testing

Value

A list of list of pathway enrichment data.frames for Both / Over / Under and for each cluster

---

`CompareedgeRGLM` *Creates a summary table with the number of genes under- or overexpressed in each group and outputs several graphical representations*

Description

Creates a summary table with the number of genes under- or overexpressed in each group and outputs several graphical representations

Usage

```
CompareedgeRGLM(
  dataMat = NULL,
  annot = NULL,
  ref_group = NULL,
  groups = NULL,
  featureTab = NULL,
  norm_method = "TMMwsp"
)
```

Arguments

dataMat	reads matrix
annot	selected annotation of interest
ref_group	List containing one or more vectors of reference samples. Name of the vectors will be used in the results table. The length of this list should be 1 or the same length as the groups list
groups	List containing the IDs of groups to be compared with the reference samples. Names of the vectors will be used in the results table
featureTab	Feature annotations to be added to the results table
norm_method	Which method to use for normalizing ('upperquantile')

Value

A dataframe containing the foldchange and p.value of each feature

Author(s)

Eric Letouze & Celine Vallot

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = choose_cluster_scExp(scExp_cf,nclust=2,consensus=FALSE)
featureTab = as.data.frame(SummarizedExperiment::rowRanges(scExp_cf))
rownames(featureTab) = featureTab$ID
ref_group = list("C1"=scExp_cf$cell_id[which(scExp_cf$cell_cluster=="C1")])
groups = list("C2"=scExp_cf$cell_id[which(scExp_cf$cell_cluster=="C2")])
myres = CompareedgeRGLM(as.matrix(SingleCellExperiment::counts(scExp_cf)),
annot=as.data.frame(SingleCellExperiment::colData(scExp_cf)),
   ref_group=ref_group,groups=groups, featureTab=featureTab)
```

`CompareWilcox` *CompareWilcox*

Description

CompareWilcox

Usage

```
CompareWilcox(
  dataMat = NULL,
  annot = NULL,
  ref_group = NULL,
  groups = NULL,
  featureTab = NULL,
  block = NULL
)
```

Arguments

`dataMat`	A raw count matrix
`annot`	A cell annotation data.frame
`ref_group`	List with cells in reference group(s)
`groups`	List with cells in group(s) to test
`featureTab`	data.frame with feature annotation
`block`	Use a blocking factor to conteract batch effect ?

Value

A dataframe containing the foldchange and p.value of each feature

Author(s)

Eric Letouze & Celine Vallot

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = choose_cluster_scExp(scExp_cf,nclust=2,consensus=FALSE)
featureTab = as.data.frame(SummarizedExperiment::rowRanges(scExp_cf))
rownames(featureTab) = featureTab$ID
ref_group = list("C1"=scExp_cf$cell_id[which(scExp_cf$cell_cluster=="C1")])
groups = list("C2"=scExp_cf$cell_id[which(scExp_cf$cell_cluster=="C2")])
myres = CompareWilcox(as.matrix(SingleCellExperiment::normcounts(scExp_cf)),
annot=as.data.frame(SingleCellExperiment::colData(scExp_cf)),
   ref_group=ref_group,groups=groups, featureTab=featureTab)
```

---

consensus_clustering_scExp

Wrapper to apply ConsensusClusterPlus to scExp object

Description

Runs consensus hierarchical clustering on PCA feature space of scExp object. Plot consensus scores for each number of clusters. See ConsensusClusterPlus - Wilkerson, M.D., Hayes, D.N. (2010). ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics, 2010 Jun 15;26(12):1572-3.

Usage

```
consensus_clustering_scExp(
  scExp,
  prefix = NULL,
  maxK = 10,
  reps = 100,
  pItem = 0.8,
  pFeature = 1,
  distance = "pearson",
  clusterAlg = "hc",
  innerLinkage = "ward.D",
  finalLinkage = "ward.D",
  plot_consclust = "pdf",
  plot_icl = "png"
)
```

Arguments

scExp	A SingleCellExperiment object containing 'PCA' in reducedDims.
prefix	character value for output directory. Directory is created only if plot_consclust is not NULL. This title can be an abosulte or relative path.
maxK	integer value. maximum cluster number to evaluate. (10)
reps	integer value. number of subsamples. (100)
pItem	numerical value. proportion of items to sample. (0.8)
pFeature	numerical value. proportion of features to sample. (1)

distance	character value. 'pearson': (1 - Pearson correlation), 'spearman' (1 - Spearman correlation), 'euclidean', 'binary', 'maximum', 'canberra', 'minkowski' or custom distance function. ('pearson')
clusterAlg	character value. cluster algorithm. 'hc' heirarchical (hclust), 'pam' for partitioning around medoids, 'km' for k-means upon data matrix, 'kmdist' ('hc') for k-means upon distance matrices (former km option), or a function that returns a clustering. ('hc')
innerLinkage	hierarchical linkage method for subsampling. ('ward.D')
finalLinkage	hierarchical linkage method for consensus matrix. ('ward.D')
plot_consclust	character value. NULL - print to screen, 'pdf', 'png', 'pngBMP' for bitmap png, helpful for large datasets. ('pdf')
plot_icl	same as above for item consensus plot. ('png')

Details

This functions takes as input a SingleCellExperiment object that must have 'PCA' in reducedDims and outputs a SingleCellExperiment object containing consclust list calculated cluster consensus and item consensus scores in metadata.

Value

Returns a SingleCellExperiment object containing consclust list, calculated cluster consensus and item consensus scores in metadata.

References

ConsensusClusterPlus package by Wilkerson, M.D., Hayes, D.N. (2010). ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics, 2010 Jun 15;26(12):1572-3.

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = consensus_clustering_scExp(scExp)
```

---

correlation_and_hierarchical_clust_scExp

Correlation and hierarchical clustering

Description

Calculates cell to cell correlation matrix based on the PCA feature space and runs hierarchical clustering taking 1 - correlation scores as distance.

Usage

```
correlation_and_hierarchical_clust_scExp(
  scExp,
  correlation = "pearson",
  hc_linkage = "ward.D"
)
```

Arguments

scExp	A SingleCellExperiment object, containing 'PCA' in reducedDims.
correlation	A correlation method to use. See hclust. ('pearson')
hc_linkage	A linkage method for hierarchical clustering. See cor. ('ward.D')

Details

This functions takes as input a SingleCellExperiment object that must have PCA calculated and outputs a SingleCellExperiment object with correlation matrix and hierarchical clustering.

Value

Return a SingleCellExperiment object with correlation matrix & hiearchical clustering.

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
```

---

create_sample_name_mat

Create a sample name matrix

Description

Create a sample name matrix

Usage

```
create_sample_name_mat(nb_samples, samples_names)
```

Arguments

nb_samples	Number of samples
samples_names	Character vector of sample names

Value

A matrix

`create_scDataset_raw` *Create a simulated single cell datamatrix & cell annotation*

Description

Create a simulated single cell datamatrix & cell annotation

Usage

```
create_scDataset_raw(
  cells = 300,
  features = 600,
  featureType = c("window", "peak", "gene"),
  sparse = TRUE,
  nsamp = 4,
  ref = "hg38",
  batch_id = factor(rep(1, nsamp))
)
```

Arguments

`cells`	Number of cells (300)
`features`	Number of features (600)
`featureType`	Type of feature (window)
`sparse`	Is matrix sparse ? (TRUE)
`nsamp`	Number of samples (4)
`ref`	Reference genome ('hg38')
`batch_id`	Batch origin (factor((1,1,1,1))

Value

A list composed of * mat : a sparse matrix following an approximation of the negative binomial law (adapted to scChIPseq) * annot : a data.frame of cell annotation * batches : an integer vector with the batch number for each cell

Examples

```
# Creating a basic sparse 600 genomic bins x 300 cells matrix and annotation
l = create_scDataset_raw()
head(l$mat)
head(l$annot)
head(l$batches)

# Specifying number of cells, features and samples
l2 = create_scDataset_raw(cells = 500, features = 500, nsamp=2)

# Specifying species
mouse_l = create_scDataset_raw(ref="mm10")

# Specifying batches
batch_l = create_scDataset_raw(nsamp=4, batch_id = factor(c(1,1,2,2)))
```

```
# Peaks of different size as features
peak_l = create_scDataset_raw(featureType="peak")
head(peak_l$mat)

# Genes as features
gene_l = create_scDataset_raw(featureType="gene")
head(gene_l$mat)
```

`create_scExp` *Wrapper to create the single cell experiment from count matrix and feature dataframe*

Description

Create the single cell experiment from (sparse) datamatrix and feature dataframe containing feature names and location. Also optionally removes zero count Features, zero count Cells, non canconical chromosomes, and chromosome M. Calculates QC Metrics (scran).

Usage

```
create_scExp(
  datamatrix,
  annot,
  remove_zero_cells = TRUE,
  remove_zero_features = TRUE,
  remove_non_canonical = TRUE,
  remove_chr_M = TRUE,
  verbose = TRUE
)
```

Arguments

`datamatrix`	A matrix or sparseMatrix of raw counts. Features x Cells (rows x columns).
`annot`	A data.frame containing informations on cells. Should have the same number of rows as the number of columns in datamatrix.
`remove_zero_cells`	remove cells with zero counts ? (TRUE)
`remove_zero_features`	remove cells with zero counts ? (TRUE)
`remove_non_canonical`	remove non canonical chromosomes ?(TRUE)
`remove_chr_M`	remove chromosomes M ? (TRUE)
`verbose`	(TRUE)

Value

Returns a SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp
```

`DA_one_vs_rest_fun` *Differential Analysis in 'One vs Rest' mode*

Description

Differential Analysis in 'One vs Rest' mode

Usage

```
DA_one_vs_rest_fun(affectation, nclust, counts, method, feature, block)
```

Arguments

`affectation`	An annotation data.frame with cell_id and cell_cluster columns
`nclust`	Number of clusters
`counts`	Count matrix
`method`	DA method : Wilcoxon or EdgeR
`feature`	Feature tables
`block`	Blocking feature

Value

A list of results, groups compared and references

`DA_pairwise` *Run differential analysis in Pairwise mode*

Description

Run differential analysis in Pairwise mode

Usage

```
DA_pairwise(affectation, nclust, counts, method, feature, block)
```

Arguments

`affectation`	An annotation data.frame with cell_cluster and cell_id columns
`nclust`	Number of clusters
`counts`	Count matrix
`method`	DA method, Wilcoxon or edgeR
`feature`	Feature data.frame
`block`	Blocking feature

Value

A list of results, groups compared and references

`define_feature` *Define the features on which reads will be counted*

Description

Define the features on which reads will be counted

Usage

```
define_feature(ref, peak_file, n_bins, bin_width, geneTSS, aroundTSS)
```

Arguments

`ref`	Reference genome
`peak_file`	A bed file if counting on peaks
`n_bins`	A number of bins if divinding genome into fixed number of bins
`bin_width`	A number of bins if divinding genome into fixed width bins
`geneTSS`	A logical indicating if feature should be counted around genes TSS instead
`aroundTSS`	Region to take in account around genes TSS

Value

A GRanges object

`detect_samples` *Heuristic discovery of samples based on cell labels*

Description

Identify a fixed number of common string (samples) in a set of varying strings (cells). E.g. in the set "Sample1_cell1","Sample1_cell2","Sample2_cell1","Sample2_cell2" and with nb_samples=2, the function returns "Sample1","Sample1","Sample2","Sample2".

Usage

```
detect_samples(barcodes, nb_samples = 1)
```

Arguments

`barcodes`	Vector of cell barcode names (e.g. Sample1_cell1, Sample1_cell2...)
`nb_samples`	Number of samples to find

Value

character vector of sample names the same length as cell labels

Examples

```
barcodes = c(paste0("HBCx22_BC_",seq_len(100)),
paste0("mouse_sample_XX",208:397))
samples = detect_samples(barcodes, nb_samples=2)
```

`differential_analysis_scExp` *Runs differential analysis between cell clusters*

Description

Based on clusters of cell defined previously, runs non-parametric Wilcoxon Rank Sum test to find significantly depleted or enriched features, in 'one_vs_rest' mode or 'pairwise' mode. In pairwise mode, each cluster is compared to all other cluster individually, and then pairwise comparisons between clusters are combined to find overall differential features using combineMarkers function from scran.

Usage

```
differential_analysis_scExp(
  scExp,
  de_type = "one_vs_rest",
  method = "wilcox",
  qval.th = 0.01,
  cdiff.th = 1,
  block = NULL
)
```

Arguments

`scExp`	A SingleCellExperiment object containing consclust with selected number of cluster.
`de_type`	Type of comparisons. Either 'one_vs_rest', to compare each cluster against all others, or 'pairwise' to make 1 to 1 comparisons. ('one_vs_rest')
`method`	Wilcoxon or edgerGLM
`qval.th`	Adjusted p-value threshold. (0.01)
`cdiff.th`	Fold change threshold. (1)
`block`	Use batches as blocking factors ?

Details

This functions takes as input a SingleCellExperiment object with consclust, the type of comparison, either 'one_vs_rest' or 'pairwise', the adjusted p-value threshold (qval.th) and the fold-change threshold (cdiff.th). It outputs a SingleCellExperiment object containing a differential list.

Value

Returns a SingleCellExperiment object containing a differential list.

Examples

```
data("scExp")
scExp_cf = differential_analysis_scExp(scExp)
```

`distPearson` *distPearson*

Description

distPearson

Usage

```
distPearson(m)
```

Arguments

`m`	A matrix

Value

A dist object

`enrichmentTest` *enrichmentTest*

Description

enrichmentTest

Usage

```
enrichmentTest(gene.sets, mylist, possibleIds, sep = ";", silent = FALSE)
```

Arguments

`gene.sets`	A list of reference gene sets
`mylist`	A list of genes to test
`possibleIds`	All existing genes
`sep`	Separator used to collapse genes
`silent`	Silent mode ?

Value

A dataframe with the gene sets and their enrichment p.value

exclude_features_scExp *Remove specific features (CNA, repeats)*

Description

Remove specific features (CNA, repeats)

Usage

```
exclude_features_scExp(
  scExp,
  features_to_exclude,
  by = "region",
  verbose = TRUE
)
```

Arguments

scExp	A SingleCellExperiment object.
features_to_exclude	A data.frame containing features to exclude.
by	Type of features. Either 'region' or 'feature_name'. If 'region', will look for genomic coordinates in columns 1-3 (chr,start,stop). If 'feature_name', will look for a genes in first column. ('region')
verbose	(TRUE)

Value

A SingleCellExperiment object without features to exclude.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
features_to_exclude = data.frame(chr=c("chr4","chr7","chr17"),
start=c(50000,8000000,2000000),
end=c(100000,16000000,2500000))

scExp
scExp = exclude_features_scExp(scExp,features_to_exclude)
scExp
```

feature_annotation_scExp *Add gene annotations to features*

Description

Add gene annotations to features

Usage

```
feature_annotation_scExp(scExp, ref = "hg38", reference_annotation = NULL)
```

Arguments

scExp	A SingleCellExperiment object.
ref	Reference genome. Either 'hg38' or 'mm10'. ('hg38')
reference_annotation	A data.frame containing gene (or else) annotation with genomic coordinates.

Value

A SingleCellExperiment object with annotated rowData.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = feature_annotation_scExp(scExp)
head(SummarizedExperiment::rowRanges(scExp))

# Mouse
scExp = create_scExp(create_scDataset_raw(ref="mm10")$mat,
  create_scDataset_raw(ref="mm10")$annot)
scExp = feature_annotation_scExp(scExp,ref="mm10")
head(SummarizedExperiment::rowRanges(scExp))
```

filter_correlated_cell_scExp *Filter lowly correlated cells*

Description

Remove cells that have a correlation score lower than what would be expected by chance with other cells.

Usage

```
filter_correlated_cell_scExp(
  scExp,
  random_iter = 50,
  corr_threshold = 99,
  percent_correlation = 1,
  run_tsne = FALSE,
  verbose = FALSE
)
```

Arguments

scExp	A SingleCellExperiment object containing 'Cor', a correlation matrix, in reducedDims.
random_iter	Number of random matrices to create to calculate random correlation scores. (50)
corr_threshold	Quantile of random correlation score above which a cell is considered to be 'correlated' with another cell. (99)
percent_correlation	Percentage of the cells that any cell must be 'correlated' to in order to not be filtered. (1)
run_tsne	Re-run tsne ? (FALSE)
verbose	(TRUE)

Details

This functions takes as input a SingleCellExperiment object that must have correlation matrix calculated and outputs a SingleCellExperiment object without lowly correlated cells. TSNE is recalculated.

Value

Returns a SingleCellExperiment object without lowly correlated cells. The calculated correlation score limit threshold is saved in metadata.

Examples

```
data("scExp")
dim(scExp)
scExp_cf = filter_correlated_cell_scExp(scExp,
corr_threshold = 99, percent_correlation = 1)
dim(scExp_cf)
```

---

filter_genes_with_refined_peak_annotation

Filter genes based on peak calling refined annotation

Description

Filter genes based on peak calling refined annotation

Usage

```
filter_genes_with_refined_peak_annotation(
  refined_annotation,
  peak_distance,
  signific,
  over,
  under
)
```

Arguments

refined_annotation	A data.frame containing each gene distance to real peak
peak_distance	Minimum distance to an existing peak to accept a given gene
signific	Indexes of all significantly differential genes
over	Indexes of all significantly overexpressed genes
under	Indexes of all significantly underexpressed genes

Value

List of significantly differential, overexpressed and underexpressed genes close enough to existing peaks

---

filter_scExp *Filter cells and features*

Description

Function to filter out cells & features from SingleCellExperiment based on total count per cell, number of cells 'ON' in features and top covered cells that might be doublets.

Usage

```
filter_scExp(
  scExp,
  min_cov_cell = 1600,
  quant_removal = 95,
  percentMin = 1,
  bin_min_count = 2,
  verbose = TRUE
)
```

Arguments

scExp	A SingleCellExperiment object.
min_cov_cell	Minimum counts for each cell. (1600)
quant_removal	Centile of cell counts above which cells are removed. (95)
percentMin	Minimum percent of cells 'ON' in feature. (1)
bin_min_count	Minimum number of counts to define if cell is 'ON'. (2)
verbose	(TRUE)

Value

Returns a filtered SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp. = filter_scExp(scExp)

# No feature filtering (all features are valuable)
scExp. = filter_scExp(scExp,percentMin=0)

# No cell filtering (all features are valuable)
scExp. = filter_scExp(scExp,min_cov_cell=0,quant_removal=100)
```

---

generate_count_matrix *Generate count matrix*

Description

Generate count matrix

Usage

```
generate_count_matrix(cells, features, sparse, cell_names, feature_names)
```

Arguments

cells	Number of cells
features	Number of features
sparse	Is matrix sparse ?
cell_names	Cell names
feature_names	Feature names

Value

A matrix or a sparse matrix

generate_feature_names *Generate feature names*

Description

Generate feature names

Usage

```
generate_feature_names(featureType, ref, features)
```

Arguments

featureType	Type of feature
ref	Reference genome
features	Number of features to generate

Value

A character vector of feature names

gene_set_enrichment_analysis_scExp *Runs Gene Set Enrichment Analysis on genes associated with differential features*

Description

This function takes previously calculated differential features and runs hypergeometric test to look for enriched gene sets in the genes associated with differential features, for each cell cluster. This functions takes as input a SingleCellExperiment object with consclust, the type of comparison, either 'one_vs_rest' or 'pairwise', the adjusted p-value threshold (qval.th) and the fold-change threshold (cdiff.th). It outputs a SingleCellExperiment object containing a differential list.

Usage

```
gene_set_enrichment_analysis_scExp(
  scExp,
  enrichment_qval = 0.1,
  ref = "hg38",
  GeneSets = NULL,
  GeneSetsDf = NULL,
  GenePool = NULL,
  qval.th = 0.01,
  cdiff.th = 1,
  peak_distance = 1000,
  use_peaks = FALSE,
 GeneSetClasses = c("c1_positional", "c2_curated", "c3_motif", "c4_computational",
    "c5_GO", "c6_oncogenic", "c7_immunologic", "hallmark")
)
```

Arguments

scExp	A SingleCellExperiment object containing list of differential features.
enrichment_qval	Adjusted p-value threshold for gene set enrichment. (0.1)
ref	A reference annotation. ('hg38')
GeneSets	A named list of gene sets. If NULL will automatically load MSigDB list of gene sets for specified reference genome. (NULL)
GeneSetsDf	A dataframe containing gene sets & class of gene sets. If NULL will automatically load MSigDB dataframe of gene sets for specified reference genome. (NULL)
GenePool	The pool of genes to run enrichment in. If NULL will automatically load Gencode list of genes fro specified reference genome. (NULL)
qval.th	Adjusted p-value threshold to define differential features. (0.01)
cdiff.th	Fold change threshold to define differential features. (1)
peak_distance	Maximum distanceToTSS of feature to gene TSS to consider associated, in bp. (1000)
use_peaks	Use peak calling method (must be calculated beforehand). (FALSE)
GeneSetClasses	Which classes of MSIGdb to look for.

Value

Returns a SingleCellExperiment object containing list of enriched Gene Sets for each cluster, either in depleted features, enriched features or simply differential features (both).

Examples

```
data("scExp")

#Usually recommanding qval.th = 0.01 & cdiff.th = 1 or 2
## Not run: scExp_cf = gene_set_enrichment_analysis_scExp(scExp,
 qval.th = 0.4, cdiff.th = 0.3)
## End(Not run)
```

---

get_color_dataframe_from_input

Get color dataframe from shiny::colorInput

Description

Get color dataframe from shiny::colorInput

Usage

```
get_color_dataframe_from_input(
  input,
  levels_selected,
  color_by = c("sample_id", "total_counts"),
  input_id_prefix = "color_"
)
```

Arguments

input	Shiny input object
levels_selected	Names of the features
color_by	Which feature color to retrieve
input_id_prefix	Prefix in front of the feature names

Value

A data.frame with the feature levels and the colors of each level of this feature.

---

get_genomic_coordinates *Get SingleCellExperiment's genomic coordinates*

Description

Get SingleCellExperiment's genomic coordinates

Usage

```
get_genomic_coordinates(scExp)
```

Arguments

scExp	A SingleCellExperiment object.

Value

A GRanges object of genomic coordinates.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
feature_GRanges = get_genomic_coordinates(scExp)
```

gg_fill_hue *gg_fill_hue*

Description

gg_fill_hue

Usage

```
gg_fill_hue(n)
```

Arguments

n	num hues

Value

A color in HEX format

groupMat *groupMat*

Description

groupMat

Usage

```
groupMat(mat = NA, margin = 1, groups = NA, method = "mean")
```

Arguments

mat	A matrix
margin	By row or columns ?
groups	Groups
method	Method to group

Value

A grouped matrix

H1proportion *H1proportion*

Description

H1proportion

Usage

```
H1proportion(pv = NA, lambda = 0.5)
```

Arguments

pv	P.value vector
lambda	Lambda value

Value

H1 proportion value

---

has_genomic_coordinates *Does SingleCellExperiment has genomic coordinates in features ?*

Description

Does SingleCellExperiment has genomic coordinates in features ?

Usage

```
has_genomic_coordinates(scExp)
```

Arguments

scExp	A SingleCellExperiment object

Value

TRUE or FALSE

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
has_genomic_coordinates(scExp)
scExp_gene = create_scExp(create_scDataset_raw(featureType="gene")$mat,
  create_scDataset_raw(featureType="gene")$annot)
has_genomic_coordinates(scExp_gene)
```

`hclustAnnotHeatmapPlot` *hclustAnnotHeatmapPlot*

Description

hclustAnnotHeatmapPlot

Usage

```
hclustAnnotHeatmapPlot(
  x = NULL,
  hc = NULL,
  hmColors = NULL,
  anocol = NULL,
  xpos = c(0.1, 0.9, 0.114, 0.885),
  ypos = c(0.1, 0.5, 0.5, 0.6, 0.62, 0.95),
  dendro.cex = 1,
  xlab.cex = 0.8,
  hmRowNames = FALSE,
  hmRowNames.cex = 0.5
)
```

Arguments

`x`	A correlation matrix
`hc`	An hclust object
`hmColors`	A color palette
`anocol`	A matrix of colors
`xpos`	Xpos
`ypos`	Ypos
`dendro.cex`	Size of denro names
`xlab.cex`	Size of x label
`hmRowNames`	Write rownames ?
`hmRowNames.cex`	Size of rownames ?

Value

A heatmap

hg38.chromosomes *Data.frame of chromosome length - hg38*

Description

This data frame provides the length of each "canonical" chromosomes of Homo Sapiens genome build hg38.

Usage

```
data("hg38.chromosomes")
```

Format

hg38.chromosomes - a data frame with 24 rows and 3 variables:

chr Chromosome - character

start Start of the chromosome (bp) - integer

end End of the chromosome (bp) - integer

hg38.GeneTSS *Data.frame of gene TSS - hg38*

Description

This dataframe was extracted from Gencode v25 and report the Transcription Start Site of each gene in the Homo Sapiens genome build hg38.

Usage

```
data("hg38.GeneTSS")
```

Format

hg38.GeneTSS - a data frame with 24 rows and 3 variables:

chr Chromosome - character

start Start of the gene (TSS) - integer

end End of the gene - integer

gene Gene symbol - character

imageCol *imageCol*

Description

imageCol

Usage

```
imageCol(
  matcol = NULL,
  strat = NULL,
  xlab.cex = 0.5,
  ylab.cex = 0.5,
  drawLines = c("none", "h", "v", "b")[1],
  ...
)
```

Arguments

matcol	A matrix of colors
strat	Strat
xlab.cex	X label size
ylab.cex	Y label size
drawLines	Draw lines ?
...	Additional parameters

Value

A rectangular image

import_count_input_files *Import and count input files depending on their format*

Description

Import and count input files depending on their format

Usage

```
import_count_input_files(
  files_dir,
  file_type,
  which,
  ref,
  peak_file_2,
  barcode_file,
  index_file,
  verbose
)
```

Arguments

files_dir	Path to the input files
file_type	Input file type
which	A GRanges object of features
ref	Reference genome
peak_file_2	A bed file for peak annotation
barcode_file	A file containing barcode names
index_file	A file containing indexes of non zero entries
verbose	Print ?

Value

A list with a GRanges object of feature types (which), the feature indexes data.frame containing non-zeroes entries in the count matrix and the cell names

---

import_scExp *Read single-cell matrix(ces) into scExp*

Description

Combine one or multiple matrices together to create a sparse matrix and cell annotation data.frame.

Usage

```
import_scExp(file_names, path_to_matrix = NULL)
```

Arguments

file_names	A character vector of file names towards single cell epigenomic matrices (features x cells) (must be .txt / .tsv)
path_to_matrix	In case matrices are stored in temporary folder, a character vector of path towards temporary files. (NULL)

Value

A list containing:

- datamatrix: a sparseMatrix of features x cells
- annot_raw: an annotation of cells as data.frame

Examples

```
mat1 = mat2 = create_scDataset_raw()$mat
tmp1 = tempfile(fileext = ".tsv")
tmp2 = tempfile(fileext = ".tsv")
write.table(as.matrix(mat1),file=tmp1,sep = "\t",
row.names = TRUE,col.names = TRUE,quote = FALSE)
write.table(as.matrix(mat2),file=tmp2, sep = "\t",
row.names = TRUE,col.names = TRUE,quote = FALSE)
file_names = c(tmp1,tmp2)
out = import_scExp(file_names)
```

index_peaks_barcodes_to_matrix_indexes

Read index-peaks-barcodes trio files on interval to create count indexes

Description

Read index-peaks-barcodes trio files on interval to create count indexes

Usage

```
index_peaks_barcodes_to_matrix_indexes(
  peak_file,
  index_file,
  name_cells,
  binarize = FALSE,
  ref = "hg38"
)
```

Arguments

peak_file	A file containing the peak genomic locations
index_file	A file containing the indexes of non-zeroes values and their value (respectively i,j,x,see sparseMatrix)
name_cells	A vector with cell names
binarize	Binarize matrix ?
ref	Reference genome

Value

A list containing a "feature index" data.frame and a region GenomicRange object both used to form the sparse matrix

launchApp

Launch ChromSCape

Description

Main function to launch ChromSCape in your favorite browser. You can pass additional parameters that you would pass to shiny::runApp (runApp)

Usage

```
launchApp(launch.browser = TRUE, ...)
```

Arguments

launch.browser	Wether to launch browser or not
...	Additional parameters passed to runApp

Value

Launches the shiny application

Examples

```
launchApp()
```

`load_MSIGdb` *Load and format MSIGdb pathways using msigdbr package*

Description

Load and format MSIGdb pathways using msigdbr package

Usage

```
load_MSIGdb(ref, GeneSetClasses)
```

Arguments

`ref`	Reference genome, either mm10 or hg38
`GeneSetClasses`	Which classes of MSIGdb to load

Value

A list containing the GeneSet (list), GeneSetDf (data.frame) and GenePool character vector of all possible genes

---

`merge_MACS2_peaks` *Merge peak files from MACS2 peak caller*

Description

Merge peak files from MACS2 peak caller

Usage

```
merge_MACS2_peaks(odir, class, peak_distance_to_merge, ref)
```

Arguments

`odir`	Output directory
`class`	Cell cluster
`peak_distance_to_merge`	Maximum distance to merge two peaks
`ref`	Reference genome

Value

Peaks as GRanges

mm10.chromosomes *Data.frame of chromosome length - mm10*

Description

This data frame provides the length of each "canonical" chromosomes of Mus Musculus (Mouse) genome build mm10.

Usage

```
data("mm10.chromosomes")
```

Format

mm10.chromosomes - a data frame with 24 rows and 3 variables:

chr Chromosome - character

start Start of the chromosome (bp) - integer

end End of the chromosome (bp) - integer

mm10.GeneTSS *Data.frame of gene TSS - mm10*

Description

This dataframe was extracted from Gencode v25 and report the Transcription Start Site of each gene in the Mus Musculus genome build mm10 (Mouse).

Usage

```
data("mm10.GeneTSS")
```

Format

mm10.GeneTSS - a data frame with 24 rows and 3 variables:

chr Chromosome name - character

start Start of the gene (TSS) - integer

end End of the gene - integer

gene Gene symbol - character

normalize_scExp *Normalize counts*

Description

Normalize counts

Usage

```
normalize_scExp(scExp, type = c("RPKM", "CPM", "TPM", "feature_size_only"))
```

Arguments

scExp	A SingleCellExperiment object.
type	Which normalization to apply. Either 'RPKM', 'CPM', 'TPM' or 'feature_size_only'. Note that for all normalization by size (RPKM, TPM, feature_size_only), the features must have defined genomic coordinates.

Value

A SingleCellExperiment object containing normalized counts. (See ?normcounts())

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = normalize_scExp(scExp)
head(SingleCellExperiment::normcounts(scExp))
```

num_cell_after_cor_filt_scExp *Number of cells before & after correlation filtering*

Description

Number of cells before & after correlation filtering

Usage

```
num_cell_after_cor_filt_scExp(scExp, scExp_cf)
```

Arguments

scExp	SingleCellExperiment object before correlation filtering.
scExp_cf	SingleCellExperiment object atfer correlation filtering.

Value

A colored kable with the number of cells per sample before and after filtering for display

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = filter_correlated_cell_scExp(scExp_cf,
corr_threshold = 99, percent_correlation = 1)
## Not run: num_cell_after_cor_filt_scExp(scExp,scExp_cf)
```

---

num_cell_after_QC_filt_scExp

Table of cells before / after QC

Description

Table of cells before / after QC

Usage

```
num_cell_after_QC_filt_scExp(scExp, annot)
```

Arguments

scExp	A SingleCellExperiment object.
annot	A raw annotation data.frame of cells before filtering.

Value

A formatted kable in HTML.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp_filtered = filter_scExp(scExp)
## Not run:  num_cell_after_QC_filt_scExp(
scExp_filtered,SingleCellExperiment::colData(scExp))
## End(Not run)
```

---

num_cell_before_cor_filt_scExp

Table of number of cells before correlation filtering

Description

Table of number of cells before correlation filtering

Usage

```
num_cell_before_cor_filt_scExp(scExp)
```

Arguments

scExp	A SingleCellExperiment Object

Value

A colored kable with the number of cells per sample for display

Examples

```
data("scExp")
## Not run: num_cell_before_cor_filt_scExp(scExp)
```

---

num_cell_in_cluster_scExp

Number of cells in each cluster

Description

Number of cells in each cluster

Usage

```
num_cell_in_cluster_scExp(scExp)
```

Arguments

scExp	A SingleCellExperiment object containing chromatin groups.

Value

A formatted kable of cell assignation to each cluster.

Examples

```
data("scExp")
scExp_cf = correlation_and_hierarchical_clust_scExp(scExp)
scExp_cf = choose_cluster_scExp(scExp_cf,nclust=3,consensus=FALSE)
## Not run: num_cell_in_cluster_scExp(scExp_cf)
```

num_cell_scExp *Table of cells*

Description

Table of cells

Usage

```
num_cell_scExp(annot)
```

Arguments

annot	An annotation of cells. Can be obtain through 'colData(scExp)'.

Value

A formatted kable in HTML.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
## Not run: num_cell_scExp(SingleCellExperiment::colData(scExp))
```

pca_irlba_for_sparseMatrix *Run sparse PCA using irlba SVD*

Description

Run sparse PCA using irlba SVD

Usage

```
pca_irlba_for_sparseMatrix(x, n_comp)
```

Arguments

x	A sparse normalized matrix (features x cells)
n_comp	The number of principal components to keep

Value

The rotated data, e.g. the cells x PC column in case of sc data.

peaks_to_bins *Transforms a peaks x cells count matrix into a bins x cells count matrix.*

Description

This functions is best used to re-count large number of small peaks (e.g. <= 5000bp) into equal or larger bins. The genome is either cut in fixed bins (e.g. 50,000bp) or into an user defined number of bins. Bins are calculated based on the canconical chromosomes. Note that if peaks are larger than bins, or if peaks are overlapping multiple bins, the signal is added to each bin. Users can increase the minimum overlap to consider peaks overlapping bins (by default 150bp, size of a nucleosome) to disminish the number of peaks overlapping multiple region. Any peak smaller than the minimum overlapp threshold will be dismissed. Therefore, library size might be slightly different from peaks to bins if signal was duplicated into multiple bins or ommitted due to peaks smaller than minimum overlap.

Usage

```
peaks_to_bins(
  mat,
  bin_width = 50000,
  n_bins = NULL,
  minoverlap = 150,
  verbose = TRUE,
  ref = "hg38"
)
```

Arguments

mat	A matrix of peaks x cells
bin_width	width of bins to produce in base pairs (minimum 500) (50000)
n_bins	number of bins (exclusive with bin_width)
minoverlap	Minimum overlap between a peak and a bin to consider the peak as overlapping the bin (150).
verbose	Verbose
ref	reference genome to use (hg38)

Value

A sparse matrix of bins instead of peaks

Examples

```
mat = create_scDataset_raw()$mat
binned_mat = peaks_to_bins(mat,bin_width = 10e6)
dim(binned_mat)
```

plot_cluster_consensus_scExp

Plot cluster consensus

Description

Plot cluster consensus score for each k as a bargraph.

Usage

```
plot_cluster_consensus_scExp(scExp)
```

Arguments

scExp	A SingleCellExperiment

Value

The consensus score for each cluster for each k as a barplot

Examples

```
data("scExp")
plot_cluster_consensus_scExp(scExp)
```

plot_differential_H1_scExp

Differential H1 distribution plot

Description

Differential H1 distribution plot

Usage

```
plot_differential_H1_scExp(scExp_cf, cell_cluster = "C1")
```

Arguments

scExp_cf	A SingleCellExperiment object
cell_cluster	Which cluster to plot

Value

A barplot of H1 distribution

Examples

```
data("scExp")
plot_differential_H1_scExp(scExp)
```

plot_differential_summary_scExp

Differential summary barplot

Description

Differential summary barplot

Usage

```
plot_differential_summary_scExp(scExp_cf)
```

Arguments

scExp_cf	A SingleCellExperiment object

Value

A barplot summary of differential analysis

Examples

```
data("scExp")
plot_differential_summary_scExp(scExp)
```

plot_differential_volcano_scExp

Volcano plot of differential features

Description

Volcano plot of differential features

Usage

```
plot_differential_volcano_scExp(
  scExp_cf,
  cell_cluster = "C1",
  cdiff.th = 1,
  qval.th = 0.01
)
```

Arguments

scExp_cf	A SingleCellExperiment object
cell_cluster	Which cluster to plot
cdiff.th	Fold change threshold
qval.th	Adjusted p.value threshold

Value

A volcano plot of differential analysis of a specific cluster

Examples

```
data("scExp")
plot_differential_volcano_scExp(scExp,"C1")
```

`plot_distribution_scExp` *Plotting distribution of signal*

Description

Plotting distribution of signal

Usage

```
plot_distribution_scExp(
  scExp,
  raw = TRUE,
  log10 = FALSE,
  pseudo_counts = 1,
  bins = 150
)
```

Arguments

`scExp`	A SingleCellExperiment Object
`raw`	Use raw counts ?
`log10`	Transform using log10 ?
`pseudo_counts`	Pseudo-count to add if using log10
`bins`	Number of bins in the histogram

Value

A ggplot histogram representing the distribution of count per cell

Examples

```
data("scExp")
plot_distribution_scExp(scExp)
```

plot_heatmap_scExp *Plot cell correlation heatmap with annotations*

Description

Plot cell correlation heatmap with annotations

Usage

```
plot_heatmap_scExp(
  scExp,
  name_hc = "hc_cor",
 corColors = (grDevices::colorRampPalette(c("royalblue", "white", "indianred1")))(256),
  color_by = NULL
)
```

Arguments

scExp	A SingleCellExperiment Object
name_hc	Name of the hclust contained in the SingleCellExperiment object
corColors	A palette of colors for the heatmap
color_by	Which features to add as additional bands on top of plot

Value

A heatmap of cell to cell correlation, grouping cells by hierarchical clustering.

Examples

```
data("scExp")
plot_heatmap_scExp(scExp)
```

plot_reduced_dim_scExp *Plot reduced dimensions (PCA, TSNE, UMAP)*

Description

Plot reduced dimensions (PCA, TSNE, UMAP)

Usage

```
plot_reduced_dim_scExp(
  scExp,
  color_by = "sample_id",
  reduced_dim = c("PCA", "TSNE", "UMAP"),
  select_x = "Component_1",
  select_y = "Component_2"
)
```

Arguments

scExp	A SingleCellExperiment Object
color_by	Feature used for coloration
reduced_dim	Reduced Dimension used for plotting
select_x	Which variable to select for x axis
select_y	Which variable to select for y axis

Value

A ggplot geom_point plot of reduced dimension 2D reprensentation

Examples

```
data("scExp")
plot_reduced_dim_scExp(scExp, color_by = "sample_id")
plot_reduced_dim_scExp(scExp, color_by = "total_counts")
plot_reduced_dim_scExp(scExp, reduced_dim = "UMAP")
```

---

preprocess_CPM *Preprocess scExp - Counts Per Million (CPM)*

Description

Preprocess scExp - Counts Per Million (CPM)

Usage

```
preprocess_CPM(scExp)
```

Arguments

scExp	A SingleCellExperiment Object

Value

A SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = preprocess_CPM(scExp)
head(SingleCellExperiment::normcounts(scExp))
```

preprocess_feature_size_only *Preprocess scExp - size only*

Description

Preprocess scExp - size only

Usage

```
preprocess_feature_size_only(scExp)
```

Arguments

scExp	A SingleCellExperiment Object

Value

A SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = preprocess_feature_size_only(scExp)
head(SingleCellExperiment::normcounts(scExp))
```

preprocess_RPKM *Preprocess scExp - Read per Kilobase Per Million (RPKM)*

Description

Preprocess scExp - Read per Kilobase Per Million (RPKM)

Usage

```
preprocess_RPKM(scExp)
```

Arguments

scExp	A SingleCellExperiment Object

Value

A SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = preprocess_RPKM(scExp)
head(SingleCellExperiment::normcounts(scExp))
```

preprocess_TPM *Preprocess scExp - Transcripts per Million (TPM)*

Description

Preprocess scExp - Transcripts per Million (TPM)

Usage

```
preprocess_TPM(scExp)
```

Arguments

scExp	A SingleCellExperiment Object

Value

A SingleCellExperiment object.

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = preprocess_TPM(scExp)
head(SingleCellExperiment::normcounts(scExp))
```

raw_counts_to_feature_count_files *Create a sparse count matrix from various format of input data.*

Description

This function takes three different type of single-cell input: - Single cell BAM files (sorted) - Single cell BED files (gzipped) - A combination of an index file, a peak file and cell barcode file (The index file is composed of three column: index i, index j and value x for the non zeroes entries in the sparse matrix.)

Usage

```
raw_counts_to_feature_count_files(
  files_dir,
  file_type = c("BAM", "BED", "Index_Peak_Barcode"),
  peak_file = NULL,
  n_bins = NULL,
  bin_width = NULL,
  geneTSS = NULL,
  aroundTSS = 2500,
  verbose = TRUE,
  ref = "hg38"
)
```

Arguments

files_dir	The directory containing the files
file_type	Input file(s) type(s) ('BAM')
peak_file	A file containing genomic location of peaks (NULL)
n_bins	The number of bins to tile the genome (NULL)
bin_width	The size of bins to tile the genome (NULL)
geneTSS	Use geneTSS regions for annotation ? (NULL)
aroundTSS	Space up and downstream of TSS to use (2500)
verbose	Verbose (TRUE)
ref	reference genome to use (hg38)

Details

This functions re-counts signal on either fixed genomic bins, a set of user-defined peaks or around the TSS of genes.

Value

A sparse matrix of features x cells

---

read_count_mat_with_separated_chr_start_end

Read a count matrix with three first columns (chr,start,end)

Description

Read a count matrix with three first columns (chr,start,end)

Usage

```
read_count_mat_with_separated_chr_start_end(
  path_to_matrix,
  format_test,
  separator
)
```

Arguments

path_to_matrix	Path to the count matrix
format_test	Sample of the read.table
separator	Separator character

Value

A sparseMatrix with rownames in the form "chr1:1222-55555"

`reduce_dims_scExp` *Reduce dimensions (PCA, TSNE, UMAP)*

Description

Reduce dimensions (PCA, TSNE, UMAP)

Usage

```
reduce_dims_scExp(
  scExp,
  dimension_reductions = c("PCA", "TSNE", "UMAP"),
  n = 50,
  batch_correction = FALSE,
  batch_list = NULL,
  verbose = TRUE
)
```

Arguments

`scExp`	A SingleCellExperiment object.
`dimension_reductions`	A character vector of methods to apply. (c('PCA','TSNE','UMAP'))
`n`	Numbers of dimensions to keep for PCA. (50)
`batch_correction`	Do batch correction ? (FALSE)
`batch_list`	List of characters. Names are batch names, characters are sample names.
`verbose`	(TRUE)

Value

A SingleCellExperiment object containing feature spaces. See ?reduceDims().

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp = reduce_dims_scExp(scExp,dimension_reductions=c("PCA","UMAP"))
scExp = normalize_scExp(scExp)
scExp = reduce_dims_scExp(scExp,dimension_reductions=c("PCA","UMAP"))
```

`reduce_dim_batch_correction` *Reduce dimension with batch corrections*

Description

Reduce dimension with batch corrections

Usage

```
reduce_dim_batch_correction(scExp, mat, batch_list, n)
```

Arguments

`scExp`	SingleCellExperiment
`mat`	The normalized count matrix
`batch_list`	List of batches
`n`	Number of PCs to keep

Value

A list containing the SingleCellExperiment with batch info and the corrected pca

`remove_chr_M_fun` *Remove chromosome M from scExprownames*

Description

Remove chromosome M from scExprownames

Usage

```
remove_chr_M_fun(scExp, verbose)
```

Arguments

`scExp`	A SingleCellExperiment
`verbose`	Print ?

Value

A SingleCellExperiment without chromosome M (mitochondrial chr)

`remove_non_canonical_fun` *Remove non canonical chromosomes from scExp*

Description

Remove non canonical chromosomes from scExp

Usage

```
remove_non_canonical_fun(scExp, verbose)
```

Arguments

`scExp`	A SingleCellExperiment
`verbose`	Print ?

Value

A SingleCellExperiment without non canonical chromosomes (random,unknown, contigs etc...)

`results_enrichmentTest` *Resutls of hypergeometric gene set enrichment test*

Description

Run hypergeometric enrichment test and combine significant pathways into a data.frame

Usage

```
results_enrichmentTest(
  differentialGenes,
  enrichment_qval,
  GeneSets,
  GeneSetsDf,
  GenePool
)
```

Arguments

`differentialGenes`	Genes significantly over / under expressed
`enrichment_qval`	Adusted p-value threshold above which a pathway is considered significative
`GeneSets`	List of pathways
`GeneSetsDf`	Data.frame of pathways
`GenePool`	Pool of possible genes for testing

Value

A data.frame with pathways passing q.value threshold

`run_pairwise_tests` *Run pairwise tests*

Description

Run pairwise tests

Usage

```
run_pairwise_tests(affectation, nclust, counts, feature, method)
```

Arguments

`affectation`	An annotation data.frame with cell_cluster and cell_id columins
`nclust`	Number of clusters
`counts`	Count matrix
`feature`	Feature data.frame
`method`	DA method, Wilcoxon or edgeR

Value

A list containing objects for DA function

`run_tsne_scExp` *Run tsne on single cell experiment*

Description

Run tsne on single cell experiment

Usage

```
run_tsne_scExp(scExp, verbose = FALSE)
```

Arguments

`scExp`	A SingleCellExperiment Object
`verbose`	Print ?

Value

A colored kable with the number of cells per sample for display

scExp *A SingleCellExperiment outputed by ChromSCape*

Description

Data from a single-cell ChIP-seq experiment against H3K4me3 active mark from two cell lines, Jurkat B cells and Ramos T cells from Grosselin et al., 2019. The count matrices, on 5kbp bins, were given to ChromSCape and the filtering parameter was set to 3% of cells active in regions and subsampled down to 150 cells per sample. After correlation filtering, the experiment is composed of respectively 51 and 55 cells from Jurkat & Ramos and 5499 5kbp-genomic bins where signal is located.

Usage

```
data("scExp")
```

Format

scExp - a SingleCellExperiment with 106 cells and 5499 features (genomic bins) in hg38:

chr A SingleCellExperiment

Details

The scExp is composed of :

- counts and normcounts assays, PCA, UMAP, and Correlation matrix in reducedDims(scExp)
- Assignation of genes to genomic bins in rowRanges(scExp)
- Cluster information in colData(scExp) correlation
- Hierarchical clustering dengogram in metadata$hc_cor
- Consensus clustering raw data in metadata$consclust
- Consensus clustering cluster-consensus and item consensus dataframes in metadata$icl
- Differential analysis in metadata$diff
- Gene Set Analysis in metadata$enr

Examples

```
data("scExp")
plot_reduced_dim_scExp(scExp)
plot_reduced_dim_scExp(scExp,color_by = "cell_cluster")
plot_heatmap_scExp(scExp)
plot_differential_volcano_scExp(scExp,cell_cluster = "C1")
plot_differential_summary_scExp(scExp)
```

separate_BAM_into_clusters *Separate BAM files into cell cluster BAM files*

Description

Separate BAM files into cell cluster BAM files

Usage

```
separate_BAM_into_clusters(affectation, odir, merged_bam)
```

Arguments

affectation	An annotation data.frame containing cell_id and cell_cluster columns
odir	A valid output directory path
merged_bam	A list of merged bam file paths @importFrom Rsamtools filterBam ScanBamParam

Value

Create one BAM per cluster from one BAM per condition

separator_count_mat *Determine Count matrix separator ("tab" or ",")*

Description

Determine Count matrix separator ("tab" or ",")

Usage

```
separator_count_mat(path_to_matrix)
```

Arguments

path_to_matrix	A path towards the count matrix to check

Value

A character separator

subsample_scExp *Subsample scExp*

Description

Randomly sample x cells from each sample in a SingleCellExperiment to return a subsampled SingleCellExperiment with all samples having maximum n cells. If n is higher than the number of cell in a sample, this sample will not be subsampled.

Usage

```
subsample_scExp(scExp, n_cells = 500)
```

Arguments

scExp	A SingleCellExperiment
n_cells	An integer number of cells to subsample for each sample (500)

Value

A subsampled SingleCellExperiment

Examples

```
scExp = create_scExp(create_scDataset_raw()$mat,create_scDataset_raw()$annot)
scExp_sub = subsample_scExp(scExp,50)
## Not run: num_cell_scExp(scExp_sub)
```

subset_bam_call_peaks *Peak calling on cell clusters*

Description

This functions does peak calling on each cell population in order to refine gene annotation for large bins. For instance, a 50000bp bin might containt the TSS of several genes, while in reality only one or two of these genes are overlapping the signal (peak). To do so, first in-silico cell sorting is applied based on previously defined clusters contained in the SingleCellExperiment. Taking BAM files of each sample as input, samtools pools then splits reads from each cell barcode into 1 BAM file per cell cluster (pseudo-bulk). Then MACS2 calls peaks on each cluster. The peaks are aggregated and merged if closer to a certain distance defined by user (10000bp). Then,

Usage

```
subset_bam_call_peaks(
  scExp,
  odir,
  inputBam,
  p.value = 0.05,
  ref = "hg38",
  peak_distance_to_merge = 10000,
  geneTSS_annotation = NULL
)
```

Arguments

scExp	A SingleCellExperiment object
odir	Output directory where to write temporary files and each cluster's BAM file
inputBam	A character vector of file paths to each sample's BAM file, containing cell barcode information as tags. BAM files can be paired-end or single-end.
p.value	a p-value to use for MACS2 to determine significant peaks. (0.05)
ref	A reference genome, either hg38 or mm10. ('hg38')
peak_distance_to_merge	Maximal distance to merge peaks together after peak calling , in bp. (10000)
geneTSS_annotation	A data.frame annotation of genes TSS. If NULL will automatically load Gencode list of genes fro specified reference genome.

Details

This function takes as input a SingleCellExperiment, that must contain a 'cell_cluster' column in it's colData, an output directory where to store temporary files, the list of BAM files corresponding to each sample and containing the cell barcode information as a tag (for instance tag CB:Z:xxx, XB:Z:xxx or else...), the p.value used by MACS2 to distinguish significant peaks, the reference genome (either hg38 or mm10), the maximal merging distance in bp and a data.frame containing gene TSS genomic cooridnates of corresponding genome (if set to NULL, will automatically load geneTSS). The output is a SingleCellExperiment with GRanges object containing ranges of each merged peaks that falls within genomic bins of the SingleCellExperiment, saving the bin range as additional column (window_chr, window_start, window_end), as well as the closests genes and their distance relative to the peak. The peaks may be present in several rows if multiple genes are close / overlap to the peaks.

Note that the user must have MACS2 installed and available in the PATH. Users can open command terminal and type 'which macs2' to verify the availability of these programs. Will only work on unix operating system. Check operating system with 'print(.Platform)'.

Value

A SingleCellExperiment with refinded annotation

Examples

```
## Not run:
data("scExp")
subset_bam_call_peaks(scExp, "path/to/out/", list("sample1" =
```

```
   "path/to/BAM/sample1.bam", "sample2" = "path/to/BAM/sample2.bam"),
   p.value = 0.05, ref = "hg38", peak_distance_to_merge = 10000,
   geneTSS_annotation = NULL)

## End(Not run)
```

`table_enriched_genes_scExp` *Creates table of enriched genes sets*

Description

Creates table of enriched genes sets

Usage

```
table_enriched_genes_scExp(
  scExp,
  set = "Both",
  cell_cluster = "C1",
 enr_class_sel = c("c1_positional", "c2_curated", "c3_motif", "c4_computational",
    "c5_GO", "c6_oncogenic", "c7_immunologic", "hallmark")
)
```

Arguments

`scExp`	A SingleCellExperiment object containing list of enriched gene sets.
`set`	A character vector, either 'Both', 'Overexpressed' or 'Underexpressed'. ('Both')
`cell_cluster`	Cell cluster. ('C1')
`enr_class_sel`	Which classes of gene sets to show. (c('c1_positional', 'c2_curated', ...))

Value

A DT::data.table of enriched gene sets.

Examples

```
data("scExp")
## Not run: table_enriched_genes_scExp(scExp)
```

`warning_DA` *Warning for differential_analysis_scExp*

Description

Warning for differential_analysis_scExp

Usage

```
warning_DA(scExp, de_type, method, qval.th, cdiff.th, block)
```

Arguments

`scExp`	A SingleCellExperiment object containing consclust with selected number of cluster.
`de_type`	Type of comparisons. Either 'one_vs_rest', to compare each cluster against all others, or 'pairwise' to make 1 to 1 comparisons. ('one_vs_rest')
`method`	Wilcoxon or edgerGLM
`qval.th`	Adjusted p-value threshold. (0.01)
`cdiff.th`	Fold change threshold. (1)
`block`	Use batches as blocking factors ?

Value

Warnings or Errors if the input are not correct

`warning_filter_correlated_cell_scExp` *warning_filter_correlated_cell_scExp*

Description

warning_filter_correlated_cell_scExp

Usage

```
warning_filter_correlated_cell_scExp(
  scExp,
  random_iter,
  corr_threshold,
  percent_correlation,
  run_tsne,
  verbose
)
```

Arguments

scExp	A SingleCellExperiment object containing 'Cor', a correlation matrix, in reducedDims.
random_iter	Number of random matrices to create to calculate random correlation scores. (50)
corr_threshold	Quantile of random correlation score above which a cell is considered to be 'correlated' with another cell. (99)
percent_correlation	Percentage of the cells that any cell must be 'correlated' to in order to not be filtered. (1)
run_tsne	Re-run tsne ? (FALSE)
verbose	(TRUE)

Value

Warnings or Errors if the input are not correct

---

warning_plot_reduced_dim_scExp *A warning helper for plot_reduced_dim_scExp*

Description

A warning helper for plot_reduced_dim_scExp

Usage

```
warning_plot_reduced_dim_scExp(
  scExp,
  color_by,
  reduced_dim,
  select_x,
  select_y
)
```

Arguments

scExp	A SingleCellExperiment Object
color_by	Feature used for coloration
reduced_dim	Reduced Dimension used for plotting
select_x	Which variable to select for x axis
select_y	Which variable to select for y axis

Value

Warning or errors if the inputs are not correct

`warning_raw_counts_to_feature_count_files` *Warning for _raw_counts_to_feature_count_files*

Description

Warning for _raw_counts_to_feature_count_files

Usage

```
warning_raw_counts_to_feature_count_files(
  files_dir,
  file_type = c("BAM", "BED", "Index_Peak_Barcode"),
  peak_file = NULL,
  n_bins = NULL,
  bin_width = NULL,
  geneTSS = NULL,
  aroundTSS = 2500,
  verbose = TRUE,
  ref = "hg38"
)
```

Arguments

`files_dir`	The directory containing the files
`file_type`	Input file(s) type(s) ('BAM')
`peak_file`	A file containing genomic location of peaks (NULL)
`n_bins`	The number of bins to tile the genome (NULL)
`bin_width`	The size of bins to tile the genome (NULL)
`geneTSS`	Use geneTSS regions for annotation ? (NULL)
`aroundTSS`	Space up and downstream of TSS to use (2500)
`verbose`	Verbose (TRUE)
`ref`	reference genome to use (hg38)

Value

Error or warnings if the input are not correct

Index