

Package ‘cinaR’

February 4, 2026

Type Package

Title A Computational Pipeline for Bulk 'ATAC-Seq' Profiles

Version 0.2.6

Author Onur Karakaslar [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-9182-910X>>),

Duygu Ucar [aut] (ORCID: <<https://orcid.org/0000-0002-9772-3066>>)

Maintainer Onur Karakaslar <eonurkara@gmail.com>

Description Differential analyses and Enrichment pipeline for bulk 'ATAC-seq' data analyses. This package combines different packages to have an ultimate package for both data analyses and visualization of 'ATAC-seq' data. Methods are described in 'Karakaslar et al.' (2021) <[doi:10.1101/2021.03.05.434143](https://doi.org/10.1101/2021.03.05.434143)>.

License GPL-3

Encoding UTF-8

LazyData true

URL <https://github.com/eonurk/cinaR/>, <https://eonurk.github.io/cinaR/>

BugReports <https://github.com/eonurk/cinaR/issues/>

Depends R (>= 4.0.0)

Imports BiocManager, DESeq2, dplyr, edgeR, fgsea, GenomicRanges, ggplot2, ggrepel, grDevices, limma, methods, utils, pheatmap, preprocessCore, RColorBrewer, sva, writexl

RoxygenNote 7.2.3

Suggests ChIPseeker, knitr, Matrix, Seurat, rmarkdown, markdown, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Mmusculus.UCSC.mm10.knownGene

VignetteBuilder knitr, rmarkdown, markdown

NeedsCompilation no

Repository CRAN

Date/Publication 2026-02-04 14:50:02 UTC

Contents

annotatePeaks	2
bed	3
cinaR	3
color_values	6
differentialAnalyses	6
dot_plot	7
filterConsensus	8
grch37	9
grch38	9
grcm38	10
GSEA	10
heatmap_differential	11
heatmap_var_peaks	12
HPEA	12
normalizeConsensus	13
pca_plot	14
prep_scATAC_cinaR	15
prep_scATAC_seurat	16
run_enrichment	17
scale_rows	18
show_comparisons	19
verboseFn	19
vp2008	20
Index	21

annotatePeaks

annotatePeaks

Description

Runs DA pipeline and makes it ready for enrichment analyses

Usage

```
annotatePeaks(cp, reference.genome, show.annotation.pie = FALSE, verbose)
```

Arguments

cp	bed formatted consensus peak matrix: CHR, START, STOP and raw peak counts (peaks by 3+samples)
reference.genome	genome of interested species. It should be 'hg38', 'hg19' or 'mm10'.
show.annotation.pie	shows the annotation pie chart produced with ChipSeeker
verbose	prints messages through running the pipeline

Value

DApeaks returns DA peaks

bed	<i>Example peaks from bone marrow of B6 mice</i>
-----	--

Description

Example peaks from bone marrow of B6 mice

Usage

```
data(atac_seq_consensus_bm)
```

Format

An object of class `data.frame` with 1000 rows and 25 columns.

Examples

```
data(atac_seq_consensus_bm)
```

cinaR	<i>cinaR</i>
-------	--------------

Description

Runs differential analyses and enrichment pipelines

Usage

```
cinaR(  
  matrix,  
  contrasts,  
  experiment.type = "ATAC-Seq",  
  DA.choice = 1,  
  DA.fdr.threshold = 0.05,  
  DA.lfc.threshold = 0,  
  comparison.scheme = "OVO",  
  save.DA.peaks = FALSE,  
  DA.peaks.path = NULL,  
  norm.method = "cpm",  
  filter.method = "custom",  
  library.threshold = 2,  
  cpm.threshold = 1,  
  TSS.threshold = 50000,
```

```

show.annotation.pie = FALSE,
reference.genome = NULL,
batch.correction = FALSE,
batch.information = NULL,
additional.covariates = NULL,
sv.number = NULL,
run.enrichment = TRUE,
enrichment.method = NULL,
enrichment.FDR.cutoff = 1,
background.genes.size = 20000,
geneset = NULL,
verbose = TRUE
)

```

Arguments

matrix	either bed formatted consensus peak matrix (peaks by 3+samples) CHR, START, STOP and raw peak counts OR count matrix (genes by 1+samples).
contrasts	user-defined contrasts for comparing samples
experiment.type	The type of experiment either set to "ATAC-Seq" or "RNA-Seq"
DA.choice	determines which pipeline to run: (1) edgeR, (2) limma-voom, (3) limma-trend, (4) DEseq2. Note: Use limma-trend if consensus peaks are already normalized, otherwise use other methods.
DA.fdr.threshold	fdr cut-off for differential analyses
DA.lfc.threshold	log-fold change cutoff for differential analyses
comparison.scheme	either one-vs-one (OVO) or one-vs-all (OVA) comparisons.
save.DA.peaks	saves differentially accessible peaks to an excel file
DA.peaks.path	the path which the excel file of the DA peaks will be saved, if not set it will be saved to current directory.
norm.method	normalization method for consensus peaks
filter.method	filtering method for low expressed peaks
library.threshold	number of libraries a peak occurs so that it is not filtered default set to 2
cpm.threshold	count per million threshold for not to filter a peak
TSS.threshold	Distance to transcription start site in base-pairs. Default set to 50,000.
show.annotation.pie	shows the annotation pie chart produced with ChipSeeker
reference.genome	genome of interested species. It should be 'hg38', 'hg19' or 'mm10'.

<code>batch.correction</code>	logical, if set will run unsupervised batch correction via sva (default) or if the batch information is known 'batch.information' argument should be provided by user.
<code>batch.information</code>	character vector, given by user.
<code>additional.covariates</code>	vector or data.frame, this parameter will be directly added to design matrix before running the differential analyses, therefore won't affect the batch corrections but adjust the results in down-stream analyses.
<code>sv.number</code>	number of surrogate variables to be calculated using SVA, best left untouched.
<code>run.enrichment</code>	logical, turns off enrichment pipeline
<code>enrichment.method</code>	There are two methodologies for enrichment analyses, Hyper-geometric p-value (HPEA) or Geneset Enrichment Analyses (GSEA).
<code>enrichment.FDR.cutoff</code>	FDR cut-off for enriched terms, p-values are corrected by Benjamini-Hochberg procedure
<code>background.genes.size</code>	number of background genes for hyper-geometric p-value calculations. Default is 20,000.
<code>geneset</code>	Pathways to be used in enrichment analyses. If not set vp2008 (Chaussabel, 2008) immune modules will be used. This can be set to any geneset using 'read.gmt' function from 'qusage' package. Different modules are available: https://www.gsea-msigdb.org/gsea/downloads.jsp .
<code>verbose</code>	prints messages through running the pipeline

Value

returns differentially accessible peaks

Examples

```
data(atac_seq_consensus_bm) # calls 'bed'

# a vector for comparing the examples
contrasts <- sapply(strsplit(colnames(bed), split = "-", fixed = TRUE),
                    function(x){x[1]}[4:25])

results <- cinaR(bed, contrasts, reference.genome = "mm10")
```

color_values	<i>color values</i>
--------------	---------------------

Description

color values

Usage

color_values

Format

An object of class character of length 8.

differentialAnalyses	<i>Differential Analyses</i>
----------------------	------------------------------

Description

Runs differential analyses pipeline of choice on consensus peaks

Usage

```
differentialAnalyses(  
  final.matrix,  
  contrasts,  
  experiment.type,  
  DA.choice,  
  DA.fdr.threshold,  
  DA.lfc.threshold,  
  comparison.scheme,  
  save.DA.peaks,  
  DA.peaks.path,  
  norm.method,  
  batch.correction,  
  batch.information,  
  additional.covariates,  
  sv.number,  
  verbose  
)
```

Arguments

<code>final.matrix</code>	Annotated Consensus peaks
<code>contrasts</code>	user-defined contrasts for comparing samples
<code>experiment.type</code>	The type of experiment either set to "ATAC-Seq" or "RNA-Seq"
<code>DA.choice</code>	determines which pipeline to run: (1) edgeR, (2) limma-voom, (3) limma-trend, (4) DEseq2
<code>DA.fdr.threshold</code>	fdr cut-off for differential analyses
<code>DA.lfc.threshold</code>	log-fold change cutoff for differential analyses
<code>comparison.scheme</code>	either one-vs-one (OVO) or one-vs-all (OVA) comparisons.
<code>save.DA.peaks</code>	logical, saves differentially accessible peaks to an excel file
<code>DA.peaks.path</code>	the path which the excel file of the DA peaks will be saved, if not set it will be saved to current directory.
<code>norm.method</code>	normalization method for consensus peaks
<code>batch.correction</code>	logical, if set will run unsupervised batch correction via sva (default) or if the batch information is known 'batch.information' argument should be provided by user.
<code>batch.information</code>	character vector, given by user.
<code>additional.covariates</code>	vector or data.frame, this parameter will be directly added to design matrix before running the differential analyses, therefore won't affect the batch corrections but adjust the results in down-stream analyses.
<code>sv.number</code>	number of surrogate variables to be calculated using SVA, best left untouched.
<code>verbose</code>	prints messages through running the pipeline

Value

returns consensus peaks (batch corrected version if enabled) and DA peaks

<code>dot_plot</code>	<i>dot_plot</i>
-----------------------	-----------------

Description

Given the results from 'cinaR' it produces dot plots for enrichment analyses.

Usage

```
dot_plot(results, fdr.cutoff = 0.1, filter.pathways = FALSE)
```

Arguments

`results` `cinaR` result object

`fdr.cutoff` Pathways with smaller fdr values than the cut-off will be shown as dots.

`filter.pathways` logical, it will filter the pathways from dot plot with fdr values less than 'fdr.cutoff'.

Value

ggplot object

Examples

```
library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'

# a vector for comparing the examples
contrasts <- sapply(strsplit(colnames(bed), split = "-", fixed = TRUE),
                    function(x){x[1]}[4:25])

results <- cinaR(bed, contrasts, reference.genome = "mm10")

dot_plot(results)
```

filterConsensus	<i>filterConsensus</i>
-----------------	------------------------

Description

Filters lowly expressed peaks from down-stream analyses

Usage

```
filterConsensus(
  cp,
  filter.method = "custom",
  library.threshold = 2,
  cpm.threshold = 1
)
```

Arguments

`cp` consensus peak matrix, with unique ids at rownames.

`filter.method` filtering method for low expressed peaks

`library.threshold` number of libraries a peak occurs so that it is not filtered default set to 2

`cpm.threshold` count per million threshold for not to filter a peak

Value

returns differentially accessible peaks

Examples

```
set.seed(123)
cp <- matrix(rexp(200, rate=.1), ncol=20)

## using cpm function from `edgeR` package
cp.filtered <- filterConsensus(cp)
```

grch37

Grch37

Description

Grch37

Usage

data(grch37)

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 66978 rows and 3 columns.

grch38

Grch38

Description

Grch38

Usage

data(grch38)

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 67495 rows and 3 columns.

`grcm38`*Grcm38*

Description`Grcm38`**Usage**`data(grcm38)`**Format**

An object of class `data.frame` with 25350 rows and 4 columns.

`GSEA`*GSEA*

Description

Gene set enrichment analyses, runs 'fgsea' package implementation with preset values.

Usage`GSEA(genes, geneset)`**Arguments**

<code>genes</code>	DA gene names to be checked if they are over-represented or not.
<code>geneset</code>	Pathways to be used in enrichment analyses. If not set <code>vp2008</code> (Chaussabel, 2008) immune modules will be used. This can be set to any geneset using 'read.gmt' function from 'qusage' package. Different modules are available: https://www.gsea-msigdb.org/gsea/downloads.jsp .

Value

`data.frame`, list of pathways and their enrichment (adjusted) p-values.

References

G. Korotkevich, V. Sukhov, A. Sergushichev. Fast gene set enrichment analysis. bioRxiv (2019), doi:10.1101/060012

Examples

```
library(cinaR)
library(fgsea)
data(examplePathways)
data(exampleRanks)
GSEA(exampleRanks, examplePathways)
```

```
heatmap_differential heatmap_differential
```

Description

plot differentially accessible peaks for a given comparison

Usage

```
heatmap_differential(results, comparison = NULL, ...)
```

Arguments

results	cinaR result object
comparison	these are created by cinaR from ‘contrasts’ user provided. If not selected the first comparison will be shown!
...	additional arguments for heatmap function, for more info ‘?pheatmap’

Value

ggplot object

Examples

```
library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'

# a vector for comparing the examples
contrasts <- sapply(strsplit(colnames(bed), split = "-", fixed = TRUE),
                    function(x){x[1]}[4:25])

results <- cinaR(bed, contrasts, reference.genome = "mm10")

heatmap_differential(results)
```

heatmap_var_peaks	<i>heatmap_var_peaks</i>
-------------------	--------------------------

Description

plot most variable k peaks (default k = 100) among all samples

Usage

```
heatmap_var_peaks(results, heatmap.peak.count = 100, ...)
```

Arguments

- results cinaR result object
- heatmap.peak.count number of peaks to be plotted. If number of peaks are less than k then all peaks will be used.
- ... additional arguments for heatmap function, for more info ‘?pheatmap’

Value

ggplot object

Examples

```
library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'

# creating dummy results
results <- NULL
results[["cp"]] <- bed[,c(4:25)]

heatmap_var_peaks(results)
```

HPEA	<i>HPEA</i>
------	-------------

Description

Hyper-geometric p-value enrichment analyses, looking for over-representation of a set of genes on given pathways.

Usage

```
HPEA(genes, geneset, background.genes.size)
```

Arguments

<code>genes</code>	DA gene names to be checked if they are over-represented or not.
<code>geneset</code>	Pathways to be used in enrichment analyses. If not set <code>vp2008</code> (Chaussabel, 2008) immune modules will be used. This can be set to any geneset using ‘ <code>read.gmt</code> ’ function from ‘ <code>qusage</code> ’ package. Different modules are available: https://www.gsea-msigdb.org/gsea/downloads.jsp .
<code>background.genes.size</code>	number of background genes for hyper-geometric p-value calculations. Default is 20,000.

Value

data.frame, list of pathways and their enrichment (adjusted) p-values.

Examples

```
library(cinaR)

data("VP2008")
genes.to.test <- vp2008[[1]][1:10]
HPEA(genes.to.test, vp2008, background.genes.size = 20e3)
```

<code>normalizeConsensus</code>	<i>normalizeConsensus</i>
---------------------------------	---------------------------

Description

Normalizes consensus peak using different methods

Usage

```
normalizeConsensus(cp, norm.method = "cpm", log.option = FALSE)
```

Arguments

<code>cp</code>	bed formatted consensus peak matrix: CHR, START, STOP and raw peak counts (peaks by 3+samples)
<code>norm.method</code>	normalization method for consensus peaks
<code>log.option</code>	logical, log option for cpm function in edgeR

Value

Normalized consensus peaks

Examples

```

set.seed(123)
cp <- matrix(rexp(200, rate=.1), ncol=20)

## using cpm function from `edgeR` package
cp.normalized <- normalizeConsensus(cp)

## quantile normalization option
cp.normalized <- normalizeConsensus(cp, norm.method = "quantile")

```

*pca_plot**pca_plot*

Description*pca_plot***Usage**

```
pca_plot(results, overlaid.info, sample.names = NULL, show.names = TRUE)
```

Arguments

<i>results</i>	cinaR result object
<i>overlaid.info</i>	overlaid information onto the samples
<i>sample.names</i>	names of the samples shown on pca plot
<i>show.names</i>	logical, if set FALSE sample names will be hidden

Value

ggplot object

Examples

```

#' library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'

# creating dummy results
results <- NULL
results[["cp"]] <- bed[,c(4:25)]

# a vector for comparing the examples
contrasts <- sapply(strsplit(colnames(bed), split = "-", fixed = TRUE),
                    function(x){x[1]})[4:25]

## overlays the contrasts info onto PCA plots
pca_plot(results, contrasts)

```

```
## you can overlay other information as well,
## as long as it is the same length with the
## number of samples.

sample.info <- c(rep("Group A", 11), rep("Group B", 11))
pca_plot(results, sample.info, show.names = FALSE)
```

```
prep_scATAC_cinaR      prep_scATAC_cinaR
```

Description

Prepare 10x scATAC peak-by-cell matrices for cinaR by pseudobulking per sample.

Usage

```
prep_scATAC_cinaR(
  counts,
  cell.meta,
  sample.col,
  group.col,
  cluster.col = NULL,
  peak.bed = NULL,
  min.cells = 20,
  verbose = TRUE
)
```

Arguments

<code>counts</code>	peak-by-cell count matrix (dense matrix or sparse dgCMatrx).
<code>cell.meta</code>	data.frame with rownames as cell barcodes.
<code>sample.col</code>	column name in cell.meta indicating biological replicate.
<code>group.col</code>	column name in cell.meta indicating condition/group.
<code>cluster.col</code>	optional column name for cell type/cluster. If provided, output is a named list per cluster.
<code>peak.bed</code>	optional data.frame with CHR/START/STOP columns for peaks. If not provided, rownames(counts) are parsed as "chr:start-end" or "chr_start_end".
<code>min.cells</code>	minimum number of cells required per sample (and per cluster if used).
<code>verbose</code>	logical, prints informative messages.

Value

list with elements 'bed', 'contrasts', and 'group.info', or a named list of such lists when cluster.col is provided.

Examples

```
counts <- matrix(c(1, 0, 2, 1, 0, 1, 3, 0, 0, 2, 1, 0),
                 nrow = 2, byrow = TRUE)
rownames(counts) <- c("chr1:1-100", "chr1:101-200")
colnames(counts) <- paste0("cell", 1:6)
meta <- data.frame(sample = c("S1", "S1", "S2", "S2", "S3", "S3"),
                  group = c("A", "A", "B", "B", "B", "B"),
                  row.names = colnames(counts))
prep <- prep_scATAC_cinaR(counts, meta, sample.col = "sample", group.col = "group",
                        min.cells = 2)
```

```
prep_scATAC_seurat      prep_scATAC_seurat
```

Description

Prepare 10x scATAC data from a Seurat/Signac object for cinaR.

Usage

```
prep_scATAC_seurat(
  object,
  assay = NULL,
  slot = "counts",
  sample.col,
  group.col,
  cluster.col = NULL,
  peak.bed = NULL,
  min.cells = 20,
  verbose = TRUE
)
```

Arguments

<code>object</code>	Seurat object containing an ATAC assay (typically "peaks").
<code>assay</code>	assay name to use; defaults to <code>Seurat::DefaultAssay(object)</code> .
<code>slot</code>	assay slot to pull counts from (default "counts").
<code>sample.col</code>	column name in <code>object@meta.data</code> indicating biological replicate.
<code>group.col</code>	column name in <code>object@meta.data</code> indicating condition/group.
<code>cluster.col</code>	optional column name for cell type/cluster.
<code>peak.bed</code>	optional data.frame with CHR/START/STOP columns for peaks.
<code>min.cells</code>	minimum number of cells required per sample (and per cluster if used).
<code>verbose</code>	logical, prints informative messages.

Value

list with elements 'bed', 'contrasts', and 'group.info', or a named list of such lists when cluster.col is provided.

Examples

```
## Not run:
prep <- prep_scATAC_seurat(seurat_obj,
                           sample.col = "sample",
                           group.col = "group",
                           assay = "peaks")

## End(Not run)
```

run_enrichment	<i>run_enrichment</i>
----------------	-----------------------

Description

This function is run, if the enrichment pipeline wants to be called afterwards. Setting reference genome to the same genome which cinaR was run should be given to this function!

Usage

```
run_enrichment(
  results,
  geneset = NULL,
  experiment.type = "ATAC-Seq",
  reference.genome = NULL,
  enrichment.method = NULL,
  enrichment.FDR.cutoff = 1,
  background.genes.size = 20000,
  verbose = TRUE
)
```

Arguments

results	list, DA peaks list for different contrasts
geneset	Pathways to be used in enrichment analyses. If not set vp2008 (Chaussabel, 2008) immune modules will be used. This can be set to any geneset using 'read.gmt' function from 'qusage' package. Different modules are available: https://www.gsea-msigdb.org/gsea/downloads.jsp .
experiment.type	The type of experiment either set to "ATAC-Seq" or "RNA-Seq"
reference.genome	genome of interested species. It should be 'hg38', 'hg19' or 'mm10'.

`enrichment.method`
 There are two methodologies for enrichment analyses, Hyper-geometric p-value (HPEA) or Geneset Enrichment Analyses (GSEA).

`enrichment.FDR.cutoff`
 FDR cut-off for enriched terms, p-values are corrected by Benjamini-Hochberg procedure

`background.genes.size`
 number of background genes for hyper-geometric p-value calculations. Default is 20,000.

`verbose`
 prints messages through running the pipeline

Value

list, enrichment analyses results along with corresponding differential analyses outcomes

Examples

```
library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'

# a vector for comparing the examples
contrasts <- sapply(strsplit(colnames(bed), split = "-", fixed = TRUE),
  function(x){x[1]}[4:25])

results <- cinaR(bed, contrasts, reference.genome = "mm10", run.enrichment = FALSE)

results_with_enrichment <- run_enrichment(results, reference.genome = "mm10")
```

scale_rows

scale_rows

Description

Normalize (z-score) rows of a matrix

Usage

```
scale_rows(x)
```

Arguments

`x` a matrix, possibly containing gene by samples

Value

Row-normalized matrix

Examples

```
library(cinaR)
data(atac_seq_consensus_bm) # calls 'bed'
bed.row.normalized <- scale_rows(bed[,c(4:25)])
head(bed.row.normalized)
```

show_comparisons	<i>show_comparisons</i>
------------------	-------------------------

Description

returns the names of the created comparisons

Usage

```
show_comparisons(results)
```

Arguments

results output of the cinaR

Value

comparisons created

verboseFn	<i>verboseFn</i>
-----------	------------------

Description

returns a printing function to be used with in the script

Usage

```
verboseFn(verbose)
```

Arguments

verbose boolean, determines whether the output going be printed or not

Value

print function

vp2008

*Immune modules***Description**

Immune modules

Usage

`data(VP2008)`

Format

An object of class GMT; see `read.gmt` from qusage package.

References

Chaussabel et al. (2008) Immunity 29:150-164 ([PubMed](#))

Index

- * **datasets**
 - bed, [3](#)
 - color_values, [6](#)
 - grch37, [9](#)
 - grch38, [9](#)
 - grcm38, [10](#)
 - vp2008, [20](#)
- annotatePeaks, [2](#)
- bed, [3](#)
- cinaR, [3](#)
- color_values, [6](#)
- differentialAnalyses, [6](#)
- dot_plot, [7](#)
- filterConsensus, [8](#)
- grch37, [9](#)
- grch38, [9](#)
- grcm38, [10](#)
- GSEA, [10](#)
- heatmap_differential, [11](#)
- heatmap_var_peaks, [12](#)
- HPEA, [12](#)
- normalizeConsensus, [13](#)
- pca_plot, [14](#)
- prep_scATAC_cinaR, [15](#)
- prep_scATAC_seurat, [16](#)
- run_enrichment, [17](#)
- scale_rows, [18](#)
- show_comparisons, [19](#)
- verboseFn, [19](#)
- vp2008, [20](#)