

# Package ‘scCNAutils’

November 5, 2018

**Title** Functions to analyze copy number aberrations in single-cell data

**Version** 0.0.0.9000

**Description** Functions to analyze copy number aberrations in single-cell data. A bunch of scripts and workflows to read and analyze scRNA-seq data and look at CNA-oriented signal.

**Depends** R (≥ 3.4.4)

**License** MIT License + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** Matrix,  
dplyr,  
magrittr,  
tidyr,  
rlang,  
parallel,  
ggplot2,  
data.table,  
Rtsne,  
FNN,  
igraph,  
RcppHMM,  
GenomicRanges,  
IRanges

**Suggests** testthat

**RoxygenNote** 6.1.0

## R topics documented:

auto_cna_call	2
auto_cna_signal	3
bin_genes	4
call_cna	5
convert_to_coord	5

define_cycling_cells . . . . .	6
find_communities . . . . .	7
make_metacells . . . . .	7
merge_samples . . . . .	8
norm_ge . . . . .	9
plot_cna . . . . .	9
plot_communities . . . . .	10
plot_qc_cells . . . . .	10
plot_tsne . . . . .	11
qc_cells . . . . .	11
qc_filter . . . . .	12
read_mtx . . . . .	13
run_pca . . . . .	13
run_tsne . . . . .	14
smooth_movingw . . . . .	15
zscore . . . . .	16

<b>Index</b>	<b>17</b>
--------------	-----------

---

auto_cna_call	<i>Automated pipeline to call CNA</i>
---------------	---------------------------------------

---

## Description

Automated pipeline to call CNA

## Usage

```
auto_cna_call(ge_df, comm_df, nb_metacells = 10, metacell_size = 3,
  trans_prob = 1e-04, baseline_cells = NULL,
  baseline_communities = NULL, prefix = "scCNAutils_out",
  nb_cores = 1, chrs = c(1:22, "X", "Y"), bin_mean_exp = 3,
  z_wins_th = 3, smooth_wnsize = 3)
```

## Arguments

ge_df	normalized gene expression of all cells (e.g. output from <a href="#">norm_ge</a> ).
comm_df	a data.frame with community information, output from <a href="#">find_communities</a> .
nb_metacells	the number of metacells per community.
metacell_size	the number of cells in a metacell.
trans_prob	the transition probability for the HMM.
baseline_cells	cells to use as baseline.
baseline_communities	communities to use as baseline. Used if baseline.cells is NULL.
prefix	the prefix to use for the files created by this function (e.g. graphs).
nb_cores	the number of processors to use.

chrs	the chromosome names to keep. NULL to include all the chromosomes.
bin_mean_exp	the desired minimum mean expression in the bin.
z_wins_th	the threshold to winsorize Z-score. Default is 3
smooth_wnsize	the window size for smoothing. Default is 3.

**Value**

a data.frame with CNAs

**Author(s)**

Jean Monlong

---

auto_cna_signal	<i>Automated pipeline to compute CNA signal from scRNA expression</i>
-----------------	-----------------------------------------------------------------------

---

**Description**

Goes from reading raw gene counts to CNA-level signal, tSNE and community detection.

**Usage**

```
auto_cna_signal(data, genes_coord, prefix = "scCNAutils_out",
  nb_cores = 1, pause_after_qc = FALSE, sample_names = NULL,
  max_mito_prop = 0.2, min_total_exp = 0, chrs = c(1:22, "X", "Y"),
  cell_cycle = NULL, bin_mean_exp = 3, z_wins_th = 3,
  smooth_wnsize = 3, cc_sd_th = 3, nb_pcs = 10, comm_k = 100)
```

**Arguments**

data	a data.frame with gene expression or the path to the folder with the 'matrix.mtx', 'genes.tsv' and 'barcodes.tsv' files. A list if multiple samples.
genes_coord	either a file name or a data.frame with coordinates and gene names.
prefix	the prefix to use for the files created by this function (e.g. graphs).
nb_cores	the number of processors to use.
pause_after_qc	pause after the QC to pick custom QC thresholds.
sample_names	the names of each sample. If NULL, tries to use data's names.
max_mito_prop	the maximum proportion of mitochondrial RNA.
min_total_exp	the minimum total cell expression
chrs	the chromosome names to keep. NULL to include all the chromosomes.
cell_cycle	if non-null, either a file or data.frame to compute cell cycle scores. See details.
bin_mean_exp	the desired minimum mean expression in the bin.

z_wins_th	the threshold to winsorize Z-score. Default is 3
smooth_wnsize	the window size for smoothing. Default is 3.
cc_sd_th	the number of SD used for the thresholds when defining cycling cells.
nb_pcs	the number of PCs used in the community detection or tSNE.
comm_k	the number of nearest neighbor for the KNN graph. Default 100.

**Value**

a data.frame with QC, community and tSNE for each cell.

**Author(s)**

Jean Monlong

---

bin_genes	<i>Merge consecutive genes into expressed bins</i>
-----------	----------------------------------------------------

---

**Description**

Merge consecutive genes into expressed bins

**Usage**

```
bin_genes(ge_df, mean_exp = 3, nb_cores = 1)
```

**Arguments**

ge_df	the input gene expression with coordinate columns (chr, start, end) and then one column per cell.
mean_exp	the desired minimum mean expression in the bin.
nb_cores	the number of processors to use.

**Value**

a data.frame with bin expression.

**Author(s)**

Jean Monlong

---

call_cna	<i>Call CNA</i>
----------	-----------------

---

**Description**

Call CNA

**Usage**

```
call_cna(z_df, trans_prob = 1e-04, nb_cores = 1, mc_info = NULL)
```

**Arguments**

**z\_df** the Z-scores, from [zscore](#).  
**trans\_prob** the transition probability for the HMM.  
**nb\_cores** the number of processor to use.  
**mc\_info** the information about the metacells, if relevant. Default is NULL.

**Value**

a data.frame with the CNA calls.

**Author(s)**

Jean Monlong

---

convert_to_coord	<i>Convert gene symbols to coordinates</i>
------------------	--------------------------------------------

---

**Description**

If *genes\_coord* is a filename, the file is expected to be a tab-delimited file with four columns: 'chr', 'start', 'end', 'symbol'. The order of the columns is not important.

**Usage**

```
convert_to_coord(ge_df, genes_coord, chrs = c(1:22, "X", "Y"),  
  rm_dup = TRUE)
```

**Arguments**

**ge\_df** the data.frame with gene expression and one column 'symbol' with gene names.  
**genes\_coord** either a file name or a data.frame with coordinates and gene names.  
**chrs** the chromosome names to keep. NULL to include all the chromosomes.  
**rm\_dup** remove duplicated coordinates? Default is TRUE.

**Details**

The gene names in column 'symbol' should match the gene names in the input *ge\_df*.

**Value**

a data.frame with columns 'chr', 'start', 'end' columns with genes coordinates (and still one column per barcode).

**Author(s)**

Jean Monlong

---

define_cycling_cells	<i>Define cycling cells</i>
----------------------	-----------------------------

---

**Description**

Define cycling cells

**Usage**

```
define_cycling_cells(qc_df, sd_th = 3)
```

**Arguments**

qc_df	the output data.frame from <a href="#">qc_cells</a> (ran with a non-null <i>cell_cycle</i> parameter)
sd_th	the number of SD used for the thresholds.

**Value**

a list with	
cells.noc	a vector with the names of non-cycling cells
graphs	a list of ggplot2 graphs

**Author(s)**

Jean Monlong

---

find_communities	<i>Community detection</i>
------------------	----------------------------

---

**Description**

Community detection

**Usage**

```
find_communities(pca_o, nb_pcs = 10, k = 100)
```

**Arguments**

pca_o	the output of <a href="#">run_pca</a>
nb_pcs	the number of PCs to use. Default 10.
k	the number of nearest neighbor for the KNN graph. Default 100.

**Value**

a data.frame with two columns: 'cell' and 'community'.

**Author(s)**

Jean Monlong

---

make_metacells	<i>Make metacells</i>
----------------	-----------------------

---

**Description**

Make metacells

**Usage**

```
make_metacells(ge_df, comm_df, nb_metacells = 10, metacell_size = 3,  
  baseline_cells = NULL, nb_cores = 1)
```

**Arguments**

ge_df	normalized gene expression of all cells (e.g. output from <a href="#">norm_ge</a> .
comm_df	a data.frame with community information, output from <a href="#">find_communities</a> .
nb_metacells	the number of metacells per community.
metacell_size	the number of cells in a metacell.
baseline_cells	the cells to use for baseline communities.
nb_cores	the number of processor to use.

**Value**

a list with

ge	a data.frame with coordinates and gene expression for each metacell.
info	information about which metacell correspond to which community.

**Author(s)**

Jean Monlong

---

merge_samples	<i>Merge expression of multiple samples</i>
---------------	---------------------------------------------

---

**Description**

Merge expression of multiple samples

**Usage**

```
merge_samples(ge_list, sample_names = NULL)
```

**Arguments**

ge_list	a list of ge_df (e.g. read from <a href="#">read_mtx</a> ).
sample_names	the names of each sample. If NULL, tries to use ge_list's names.

**Value**

a list with

ge	the merged gene expression data.frame
info	a data.frame with new and original cell names, and corresponding sample name

**Author(s)**

Jean Monlong



---

norm_ge	<i>Normalize gene expression</i>
---------	----------------------------------

---

**Description**

Normalize gene expression

**Usage**

```
norm_ge(ge_df, method = c("tmm", "total"), nb_cores = 1)
```

**Arguments**

ge_df	the input gene expression
method	the normalization method
nb_cores	the number of processors to use.

**Value**

a data.frame with the normalized expression.

**Author(s)**

Jean Monlong

---

plot_cna	<i>Heatmap of CNA</i>
----------	-----------------------

---

**Description**

Heatmap of CNA

**Usage**

```
plot_cna(cna_df, chrs_order = c(1:22, "X", "Y"))
```

**Arguments**

cna_df	CNA from <a href="#">call_cna</a> .
chrs_order	order of the chromosomes in the graph.

**Value**

a ggplot2 graph

**Author(s)**

Jean Monlong

---

plot_communities	<i>Community graphs</i>
------------------	-------------------------

---

**Description**

Community graphs

**Usage**

```
plot_communities(comm_df, qc_df = NULL, info_df = NULL)
```

**Arguments**

comm_df	the output data.frame from <a href="#">find_communities</a>
qc_df	a data.frame with QC metrics (output from <a href="#">qc_cells</a> ). Default is NULL (i.e. not used)
info_df	a data.frame with sample merge info (output from <a href="#">merge_samples</a> ). Default is NULL (i.e. not used)

**Value**

a list of ggplot2 graphs.

**Author(s)**

Jean Monlong

---

plot_qc_cells	<i>QC graphs</i>
---------------	------------------

---

**Description**

QC graphs

**Usage**

```
plot_qc_cells(qc_df)
```

**Arguments**

qc_df	the output data.frame from <a href="#">qc_cells</a>
-------	-----------------------------------------------------

**Value**

a list of ggplots

**Author(s)**

Jean Monlong

---

plot_tsne	<i>tSNE graphs</i>
-----------	--------------------

---

**Description**

tSNE graphs

**Usage**

```
plot_tsne(tsne_df, qc_df = NULL, comm_df = NULL, info_df = NULL)
```

**Arguments**

tsne_df	the output data.frame from <a href="#">run_tsne</a> (columns: cell, tsne1, tsne2)
qc_df	a data.frame with QC metrics (output from <a href="#">qc_cells</a> ). Default is NULL (i.e. not used)
comm_df	a data.frame with communities (output from <a href="#">find_communities</a> ). Default is NULL (i.e. not used)
info_df	a data.frame with sample merge info (output from <a href="#">merge_samples</a> ).

**Value**

a list of ggplot objects

**Author(s)**

Jean Monlong

---

qc_cells	<i>Compute quality control metrics for each cell</i>
----------	------------------------------------------------------

---

**Description**

If cell\_cycle is provided it should be a data.frame (or a tsv file) with two columns: 'symbol' with gene names, and 'phase' with the cell cycle phase (e.g. either 'G1.S' or 'G2.M').

**Usage**

```
qc_cells(ge_df, cell_cycle = NULL)
```

**Arguments**

ge_df	the input gene expression with a 'symbol' column and then one column per cell.
cell_cycle	if non-null, either a file or data.frame to compute cell cycle scores. See details.

**Value**

a data.frame with qc metrics per cell.

**Author(s)**

Jean Monlong

---

qc_filter	<i>Filter cells based on QC results</i>
-----------	-----------------------------------------

---

**Description**

Filter cells based on QC results

**Usage**

```
qc_filter(ge_df, qc_df, max_mito_prop = 0.2, min_total_exp = 0)
```

**Arguments**

ge_df	the input gene expression with a 'symbol' column and then one column per cell.
qc_df	the output data.frame from qc.cells
max_mito_prop	the maximum proportion of mitochondrial RNA.
min_total_exp	the minimum total cell expression

**Value**

*ge\_df* with only the cells that passed the filters

**Author(s)**

Jean Monlong

---

read_mtx	<i>Read a trio of genes, barcodes and mtx files.</i>
----------	------------------------------------------------------

---

### Description

Read a trio of genes, barcodes and mtx files.

### Usage

```
read_mtx(mtx_file = "matrix.mtx", genes_file = "genes.tsv",
         barcodes_file = "barcodes.tsv", path = ".", rm_dup = TRUE,
         genes_col = 2)
```

### Arguments

mtx_file	the path to the mtx file
genes_file	the path to the genes file.
barcodes_file	the path to the barcodes file
path	the path to the folder containing the files
rm_dup	remove duplicated gene names? Default is TRUE.
genes_col	the column to use in genes_file. Default is 2.

### Value

a data.frame with a 'symbol' column with gene names and one column per barcode.

### Author(s)

Jean Monlong

---

run_pca	<i>Run PCA</i>
---------	----------------

---

### Description

Cells in `core_cells` are used to build the principal components to which all cells are then projected to. Usually used to reduce the effect of cell cycle in the PCA, by using only cells that don't cycle (see [qc.cells](#)) as *core\_cells*.

### Usage

```
run_pca(z_df, core_cells = NULL, out_pcs = 100)
```

**Arguments**

<code>z_df</code>	a data.frame with z-scores for each cell
<code>core_cells</code>	if non-NULL, a vector with the names of the cells to use as core cells. See details. Default is NULL.
<code>out_pcs</code>	the number of top PCs to report. Default is 100.

**Details**

The graph (*sdev.graph*) shows the standard deviation for the top 50 PCs. To show more/less PCs, add `xlim(1,N)` to the *sdev.graph*. See examples.

**Value**

a list with	
<code>x</code>	the PC matrix
<code>sdev</code>	the standard deviations of the PCs
<code>sdev.graph</code>	a ggplot graph of the sdev

**Author(s)**

Jean Monlong

**Examples**

```
## Not run:
pca.o = run_pca(z)

## Zoom in to the top 20 PCs
pca.o$sdev.graph + xlim(1,20)

## End(Not run)
```

---

run\_tsne

*Run tSNE*


---

**Description**

Run tSNE

**Usage**

```
run_tsne(pca_o, nb_pcs = 10, nb_it = 1000)
```

**Arguments**

<code>pca_o</code>	the output of <a href="#">run_pca</a>
<code>nb_pcs</code>	the number of PCs to use. Default 10.
<code>nb_it</code>	the number of iterations. Default 1000.

**Value**

a data.frame with columns: cell, tsne1, tsne2

**Author(s)**

Jean Monlong

---

smooth_movingw	<i>Moving-window smoothing</i>
----------------	--------------------------------

---

**Description**

Moving-window smoothing

**Usage**

```
smooth_movingw(df, wsize = 3, nb_cores = 1, FUN = stats::median)
```

**Arguments**

df	the input data.frame with coordinate columns (chr, start, end) and then one column per cell
wsize	the window size. Default is 3.
nb_cores	the number of processors to use.
FUN	the function to apply to each window. Default is median.

**Value**

a data.frame with smoothed signal.

**Author(s)**

Jean Monlong

---

zscore	<i>Compute Z-score</i>
--------	------------------------

---

**Description**

Compute Z-score

**Usage**

```
zscore(ge_df, wins_th = 3, method = c("z", "norm"), normals = NULL)
```

**Arguments**

ge_df	the input expression data.frame
wins_th	the threshold to winsorize Z-score. Default is 3
method	the normalization method. Either 'z' or 'norm'.
normals	the cells to use as normals. If NULL (default) all cells are used as normals

**Value**

a data.frame with Z-scores.

**Author(s)**

Jean Monlong



# Index

auto\_cna\_call, [2](#)  
auto\_cna\_signal, [3](#)  
  
bin\_genes, [4](#)  
  
call\_cna, [5](#), [9](#)  
convert\_to\_coord, [5](#)  
  
define\_cycling\_cells, [6](#)  
  
find\_communities, [2](#), [7](#), [7](#), [10](#), [11](#)  
  
make\_metacells, [7](#)  
merge\_samples, [8](#), [10](#), [11](#)  
  
norm\_ge, [2](#), [7](#), [9](#)  
  
plot\_cna, [9](#)  
plot\_communities, [10](#)  
plot\_qc\_cells, [10](#)  
plot\_tsne, [11](#)  
  
qc\_cells, [6](#), [10](#), [11](#), [11](#), [13](#)  
qc\_filter, [12](#)  
  
read\_mtx, [8](#), [13](#)  
run\_pca, [7](#), [13](#), [14](#)  
run\_tsne, [11](#), [14](#)  
  
smooth\_movingw, [15](#)  
  
zscore, [5](#), [16](#)