

Package: SCEVAN (via r-universe)

May 27, 2026

Type Package

Title Single Cell Variational Aneuploidy aNalysis

Version 1.0.6

Date 2026-01-24

Author Zaoqu Liu [ctb, cre], A. De Falco [aut], M. Ceccarelli [aut]

Maintainer Zaoqu Liu <liuzaoqu@163.com>

Description SCEVAN automatically classifies cells in scRNA-seq data by segregating non-malignant cells of tumor microenvironment from malignant cells. It also infers copy number profiles of malignant cells, identifies subclonal structures and analyzes specific and shared alterations of each subpopulation.

License GPL-2

URL <https://zaoqu-liu.github.io/SCEVAN/>,
<https://github.com/Zaoqu-Liu/SCEVAN>

BugReports <https://github.com/Zaoqu-Liu/SCEVAN/issues>

Imports parallelDist, pheatmap, forcats, parallel, dplyr, magrittr, methods, fgsea, ggplot2, Rtsne, scran, igraph, ape, ggtree, ggrepel, RColorBrewer, grid, stats, utils, grDevices, graphics

LazyLoad yes

LazyData true

NeedsCompilation yes

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Suggests rmarkdown, knitr, Seurat, clue, Biobase, biomaRt

VignetteBuilder knitr

Encoding UTF-8

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev zlib1g-dev

Repository <https://zaoqu-liu.r-universe.dev>

Date/Publication 2026-01-24 06:31:33 UTC

RemoteUrl <https://github.com/Zaoqu-Liu/SCEVAN>

RemoteRef main

RemoteSha 07b7f0383bcfea19fe4bd2aee4488aa4b072da67

Contents

annotateGenes	2
annotateBandOncoHeat	3
classifyCluster	3
classifyTumorCells	4
computeCNAmtx	5
getBreaksVegaMC	6
getConfidentNormalCells	6
getCountMtxFromSeurat	7
multiSampleComparisonClonalCN	8
pipelineCNA	8
plotAllClonalCN	10
plotAllSubclonalCN	10
plotCNA_withAnnotCells	11
preprocessingMtx	12
removeSyntheticBaseline	13
SCEVAN	13
sortData	14
top30classification	15
Index	16

annotateGenes	<i>annotateGenes Annotate genes with genomic coordinates with reference to hg38 using Ensembl based annotation package</i>
---------------	--

Description

annotateGenes Annotate genes with genomic coordinates with reference to hg38 using Ensembl based annotation package

Usage

```
annotateGenes(mtx, organism = "human")
```

Arguments

mtx	Count matrix with genes on row names (Ensemble or Symbol)
organism	Organism to be analysed ("human" or "mouse", default "human")

Value

Annotated matrix

Examples

```
## Not run:
count_mtx_annot <- annotateGenes(count_mtx)

## End(Not run)
```

annotateBandOncoHeat *annotateBandOncoHeat* Annotate with chromosome bands the data frame with difference copy number alterations between subclones

Description

annotateBandOncoHeat Annotate with chromosome bands the data frame with difference copy number alterations between subclones

Usage

```
annotateBandOncoHeat(mtx_annot, diffSub, nSub, organism = "human")
```

Arguments

mtx_annot	Annotation matrix
diffSub	Data frame with difference copy number alterations between subclones
nSub	Number of subclones
organism	Organism to be analysed (default = "human")

classifyCluster *classifyCluster* Classify the two major clusters of CNA matrix on the basis of confident normal cells

Description

classifyCluster Classify the two major clusters of CNA matrix on the basis of confident normal cells

Usage

```
classifyCluster(hcc2, norm_cell_names)
```

Arguments

hcc2 Two clusters from hierarchical clustering
 norm_cell_names Vector of confident normal cells

Value

classification of tumor and normal cells

classifyTumorCells *classifyTumorCells* Classify tumour and normal cells from the raw count matrix, using normal cells in the matrix or by subtracting a synthetic baseline from the matrix if there are no normal cells in the matrix.

Description

classifyTumorCells Classify tumour and normal cells from the raw count matrix, using normal cells in the matrix or by subtracting a synthetic baseline from the matrix if there are no normal cells in the matrix.

Usage

```
classifyTumorCells(
  count_mtx,
  annot_mtx,
  sample = "",
  distance = "euclidean",
  par_cores = 20,
  ground_truth = NULL,
  norm_cell_names = NULL,
  SEGMENTATION_CLASS = TRUE,
  SMOOTH = TRUE,
  beta_vega = 0.5,
  FIXED_NORMAL_CELLS = FALSE,
  output_dir = "./output"
)
```

Arguments

count_mtx raw count matrix
 annot_mtx matrix containing the annotations of the genes (rows: genes, columns: chr start end)
 sample sample name (optional)
 distance distance used in hierarchical clustering (default euclidean)
 par_cores number of cores (default 20)

norm_cell_names	confident normal cells (optional)
SEGMENTATION_CLASS	Boolean value to perform segmentation before classification (default TRUE)
SMOOTH	Boolean value to perform smoothing (default TRUE)
beta_vega	specifies beta parameter for segmentation, higher beta for more coarse-grained segmentation. (default 0.5)
FIXED_NORMAL_CELLS	TRUE if vector of norm_cell to be used as reference fixed, if you are interested only in clonal structure e non nella classificazione normal/tumor (default FALSE)
gr_truth	ground truth of classification (optional)

computeCNAmtx	<i>computeCNAmtx computed the CNA matrix using the break points obtained from segmentation</i>
---------------	--

Description

computeCNAmtx computed the CNA matrix using the break points obtained from segmentation

Usage

```
computeCNAmtx(count_mtx, breaks, par_cores = 20, segmAlt)
```

Arguments

count_mtx	count matrix
par_cores	number of cores for parallel computing (optional)
breaksbreak	points obtained from segmentation

Value

CNA matrix

getBreaksVegaMC *getBreaksVegaMC* Get SCEVAN segmentation of the matrix.

Description

getBreaksVegaMC Get SCEVAN segmentation of the matrix.

Usage

```
getBreaksVegaMC(  
  mtx,  
  chr_vect,  
  sample = "",  
  beta_vega = 0.5,  
  output_dir = "./output"  
)
```

Arguments

mtx	count matrix
chr_vect	Vector specifying for each gene the chromosome where it is located
sample	sample name (optional)
beta_vega	specifies beta parameter for segmentation, higher beta for more coarse-grained segmentation. (default 0.5)

Value

breakpoints

getConfidentNormalCells
getConfidentNormalCells Get at most top 30 confident normal cells from count matrix.

Description

getConfidentNormalCells Get at most top 30 confident normal cells from count matrix.

Usage

```
getConfidentNormalCells(
  mtx,
  sample = "",
  par_cores = 20,
  AdditionalGeneSets = NULL,
  SCEVANSignatures = TRUE,
  organism = "human",
  output_dir = "./output"
)
```

Arguments

mtx	count matrix
sample	sample name (optional)
par_cores	number of cores (default 20)
AdditionalGeneSets	list of additional signatures of normal cell types (optional)
SCEVANSignatures	FALSE if you only want to use only the signatures specified in AdditionalGeneSets (default TRUE)

getCountMtxFromSeurat *getCountMtxFromSeurat Extract count matrix from Seurat object (V4 and V5 compatible)*

Description

This function extracts the raw count matrix from a Seurat object, supporting both Seurat V4 and V5 data structures. It prioritizes V4 format.

Usage

```
getCountMtxFromSeurat(seurat_obj, assay = "RNA", layer = "counts")
```

Arguments

seurat_obj	A Seurat object
assay	Assay name to use (default "RNA")
layer	Layer name for Seurat V5 (default "counts")

Value

Raw count matrix with genes on rows and cells on columns

Examples

```
## Not run:
count_mtx <- getCountMtxFromSeurat(seurat_obj)
results <- pipelineCNA(count_mtx)

## End(Not run)
```

multiSampleComparisonClonalCN

multiSampleComparisonClonalCN Compare the clonal Copy Number of multiple samples.

Description

multiSampleComparisonClonalCN Compare the clonal Copy Number of multiple samples.

Usage

```
multiSampleComparisonClonalCN(
  listCountMtx,
  listNormCells = NULL,
  analysisName = "all",
  organism = "human",
  par_cores = 20,
  plotTree = TRUE,
  output_dir = "./output"
)
```

Arguments

listCountMtx	Named list of raw count matrix of samples
analysisName	Name of the analysis (default "all")
organism	Organism to be analysed (optional - "mouse" or "human" - default "human")
par_cores	number of cores (default 20)

pipelineCNA

pipelineCNA Executes the entire SCEVAN pipeline that classifies tumour and normal cells from the raw count matrix, infer the clonal profile of cancer cells and looks for possible sub-clones in the tumour cell matrix automatically analysing the specific and shared alterations of each subclone and a differential analysis of pathways and genes expressed in each subclone.

Description

pipelineCNA Executes the entire SCEVAN pipeline that classifies tumour and normal cells from the raw count matrix, infer the clonal profile of cancer cells and looks for possible sub-clones in the tumour cell matrix automatically analysing the specific and shared alterations of each subclone and a differential analysis of pathways and genes expressed in each subclone.

Usage

```
pipelineCNA(
  count_mtx,
  sample = "",
  par_cores = 20,
  norm_cell = NULL,
  SUBCLONES = TRUE,
  beta_vega = 0.5,
  ClonalCN = TRUE,
  plotTree = TRUE,
  AdditionalGeneSets = NULL,
  SCEVANsignatures = TRUE,
  organism = "human",
  ngenes_chr = 5,
  perc_genes = 10,
  FIXED_NORMAL_CELLS = FALSE,
  output_dir = "./output"
)
```

Arguments

count_mtx	Raw count matrix with genes on rows (both Gene Symbol or Ensembl ID are allowed) and cells on columns.
sample	Sample name to save results (optional)
par_cores	Number of cores to run the pipeline (optional - default 20)
norm_cell	Vector of possible known normal cells to be used as confident normal cells (optional)
SUBCLONES	Boolean value TRUE if you are interested in analysing the clonal structure and FALSE if you are only interested in the classification of malignant and non-malignant cells (optional - default TRUE)
beta_vega	Specifies beta parameter for segmentation, higher beta for more coarse-grained segmentation. (optional - default 0.5)
ClonalCN	Get clonal CN profile inference from all tumour cells (optional)
plotTree	Plot Phylogenetic tree (optional - default FALSE)
AdditionalGeneSets	list of additional signatures of normal cell types (optional)
SCEVANsignatures	FALSE if you only want to use only the signatures specified in AdditionalGeneSets (default TRUE)

organism	Organism to be analysed (optional - "mouse" or "human" - default "human")
ngenes_chr	Minimum number of genes expressed on chromosome (optional - default 5)
perc_genes	Minimum percentage gene expressed in each cell (optional - default 10)
FIXED_NORMAL_CELLS	TRUE if norm_cell vector to be used as fixed reference, if you are only interested in clonal structure and not normal/tumor classification (default FALSE)

Examples

```
## Not run:
res_pip <- pipelineCNA(count_mtx)

## End(Not run)
```

plotAllClonalCN	<i>Title plotAllClonalCN</i>
-----------------	------------------------------

Description

Title plotAllClonalCN

Usage

```
plotAllClonalCN(samples, name)
```

Arguments

samples	Vector with sample names to be plotted
name	Analysis name

plotAllSubclonalCN	<i>plotAllSubclonalCN Plot the copy number of each subclone of a sample.</i>
--------------------	--

Description

plotAllSubclonalCN Plot the copy number of each subclone of a sample.

Usage

```
plotAllSubclonalCN(sample, pathOutput = "./output/")
```

Arguments

sample	Name of the sample.
pathOutput	Path to the output folder containing the output of pipelineCNA.

```
plotCNA_withAnnotCells
```

plotCNA_withAnnotCells allows generating a heatmap of the copy number profile of each cell, adding cell annotations as tracks on the heatmap.

Description

plotCNA_withAnnotCells allows generating a heatmap of the copy number profile of each cell, adding cell annotations as tracks on the heatmap.

Usage

```
plotCNA_withAnnotCells(
  SampleName,
  metadata,
  COLUMNS_TO_PLOT,
  outputPATH = "./output/",
  SUBCLONE = FALSE,
  hcc = NULL,
  plotNAME = "heatmap_with_annotation.png",
  par_cores = 20
)
```

Arguments

SampleName	Sample name used in pipelineCNA
metadata	data.frame cells (rownames) and annotations (columns)
COLUMNS_TO_PLOT	columns of metadata to be added as tracks in the heatmap
outputPATH	output folder of pipelineCNA (optional)
SUBCLONE	Boolean value TRUE if you are interested in CNA matrix of subclone and FALSE if you are interested in CNA matrix of all cells.
hcc	if you have previously computed clustering for the heatmap (optional - default 0.5)
plotNAME	name file to save the figure (optional)
par_cores	number of cores used for clustering (optional - default 20)

Examples

```
## Not run:
plotCNA_withAnnotCells(SampleName, metadata, c("CellType","Tissue","Cluster"))

## End(Not run)
```

preprocessingMtx	<i>preprocessingMtx</i> Pre-processing steps: Cells with less than 200 genes and the genes expressed in less than 1 according to genomic coordinates. Highly confident normal cells are sought in the matrix. Genes involved in the cell cycle pathway are removed. Log-Freeman–Tukey transformation to stabilize variance and a polynomial dynamic linear modeling (DLM) to smooth out the outliers.
------------------	---

Description

preprocessingMtx Pre-processing steps: Cells with less than 200 genes and the genes expressed in less than 1 according to genomic coordinates. Highly confident normal cells are sought in the matrix. Genes involved in the cell cycle pathway are removed. Log-Freeman–Tukey transformation to stabilize variance and a polynomial dynamic linear modeling (DLM) to smooth out the outliers.

Usage

```
preprocessingMtx(
  count_mtx,
  sample,
  ngenes_chr = 5,
  perc_genes = 0.1,
  par_cores = 20,
  findConfident = TRUE,
  AdditionalGeneSets = NULL,
  SCEVANsignatures = TRUE,
  organism = "human",
  output_dir = "./output"
)
```

Arguments

count_mtx	raw count matrix
ngenes_chr	minimum number of genes per chromosome (optional)
perc_genes	percentage of cells in which each gene is to be expressed (optional)
par_cores	number of cores (optional)
findConfident	Boolean value to search for normal cells (default TRUE)
AdditionalGeneSets	List of additional signatures to be used to search for normal cells (optional)
SCEVANsignatures	Boolean value TRUE to use internal SCEVAN signatures for normal cells or FALSE to use only signatures specified in AdditionalGeneSets (default TRUE)
SMOOTH	Boolean value to perform smoothing (optional)

Value

count_mtx_smooth processed and smoothed matrix count_mtx_annot annotated matrix

Examples

```
## Not run:
res <- preprocessingMtx(count_mtx, sample = "test")

## End(Not run)
```

```
removeSyntheticBaseline
```

removeSyntheticBaseline Removes a synthetic baseline from a tumour pure matrix

Description

removeSyntheticBaseline Removes a synthetic baseline from a tumour pure matrix

Usage

```
removeSyntheticBaseline(count_mtx, par_cores = 20)
```

Arguments

count_mtx	count matrix
par_cores	number of cores for parallel computing.

Value

relative matrix

SCEVAN

SCEVAN: R package that automatically classifies the cells in the scRNA data by segregating non-malignant cells of tumor microenvironment from the malignant cells. It also infers the copy number profile of malignant cells, identifies subclonal structures and analyses the specific and shared alterations of each subpopulation.

Description

The SCEVAN package

SCEVAN functions

The SCEVAN functions ...

See Also

Useful links:

- <https://github.com/Zaoqu-Liu/SCEVAN>
- Report bugs at <https://github.com/Zaoqu-Liu/SCEVAN/issues>

sortData

This function sorts a dataset file by the genomic position of the probes.

Description

This function sorts a dataset file by the genomic position of the probes. This function makes very easy the integration of VegaMC with the output of PennCNV tool.

Usage

```
sortData(dataset, output_file_name = "")
```

Arguments

dataset Dataset file.
output_file_name Name of the file in which sorted data are stored.

Value

This function returns the input matrix ordered by the genomic position of the probes.

Note

This function allows to sort a dataset by the genomic position. The input file must have the chromosome and the position in column two and three respectively. This format follows the standard output of PennCNV. An example of file can be found in inst/example folder.

Author(s)

Sandro Morganella

References

Morganella S., and Ceccarelli M. VegaMC: a R/bioconductor package for fast downstream analysis of large array comparative genomic hybridization datasets. *Bioinformatics*, 28(19):2512-4 (2012).

Examples

```
## Not run:
## Copy the example dataset in current folder
file.copy(system.file("example/breast_Affy500K.txt", package="VegaMC"),
          ".")

## Sort data and save results in sorted.txt file
sortData("breast_Affy500K.txt", "sorted.txt")

## End(Not run)
```

top30classification *Get at most top 30 confident normal cells*

Description

Get at most top 30 confident normal cells

Usage

```
top30classification(
  NES,
  pValue,
  FDR,
  pval_filter,
  fdr_filter,
  pval_cutoff,
  nes_cutoff,
  nNES
)
```

Arguments

NES
pValue
FDR
pval_filter
fdr_filter
pval_cutoff
nes_cutoff
nNES

Index

annotateGenes, [2](#)
annotateBandOncoHeat, [3](#)

classifyCluster, [3](#)
classifyTumorCells, [4](#)
computeCNAMtx, [5](#)

getBreaksVegaMC, [6](#)
getConfidentNormalCells, [6](#)
getCountMtxFromSeurat, [7](#)

multiSampleComparisonClonalCN, [8](#)

pipelineCNA, [8](#)
plotAllClonalCN, [10](#)
plotAllSubclonalCN, [10](#)
plotCNA_withAnnotCells, [11](#)
preprocessingMtx, [12](#)

removeSyntheticBaseline, [13](#)

SCEVAN, [13](#)
SCEVAN-package (SCEVAN), [13](#)
sortData, [14](#)

top30classification, [15](#)