Package 'scTyper'

July 16, 2020

Type Package

Title scTyper: a comprehensive pipeline for the cell typing analysis of single-cell RNA-seq data

Version 0.1.0

Author Ji-Hye Choi, Hye In Kim, Hyun Goo Woo

Maintainer Hyun Goo Woo <HyunGooWoo@gmail.com>

Description scTyper provides a comprehensive and user-friendly analysis pipeline for the cell typing of scRNA-Seq data with a curated cell marker database, scTyper.db.

License GPL2

Encoding UTF-8

LazyData true

Suggests knitr, BiocStyle

Depends R (>= 3.5)

Imports fastqcr, Seurat, infercnv, gProfileR, rmarkdown, parallel, perm, Biobase, reshape2, limma, GenomicRanges, ggplot2, grid, gridExtra, grDevices, ComplexHeatmap, circlize, png, knitr, kableExtra, pander, colorspace

RoxygenNote 7.1.1

VignetteBuilder knitr

R topics documented:

`cell.typing.seurat` *cell.typing.seurat*

Description

Wrapper function for cell typing of seurat object

Usage

```
cell.typing.seurat(
  seurat,
  marker = "Puram.2017.HNSCC",
  cell.typing.method = c("NTP", "ES", "Average"),
  level = c("cell", "cluster"),
  wd,
  slot = c("scale.data", "count.data", "data"),
  assay = "RNA",
  ntp.dir,
  rda.dir,
  NTP.g.filter.method = c("sd", "mad", "none"),
  NTP.gene.filter.cutoff = 0.3,
  NTP.distance = c("cosine", "correlation"),
  NTP.norm.method = c("none", "row.std"),
  mc.cores = 1
)
```

Arguments

`seurat`	Seurat object
`marker`	Cell markers to use in cell typing, character or List (identifier or StudyName or User defined gene list)
`cell.typing.method`	cell typing method, c("NTP", "ES", "Average"), (default = "NTP")
`wd`	working directory
`slot`	assay data type of seurat object, c("scale.data", "count.data", "data")
`assay`	Assay of seurat object
`ntp.dir`	Output directory of NTP
`rda.dir`	Path of the RData saving directory

NTP.g.filter.method
: Method of gene filtering in NTP c(sd (Default), mad, none)

NTP.gene.filter.cutoff
: Cut-off score of standard deviation in NTP

NTP.distance
: Method of calculating distance in NTP, either c("correlation" or "cosine").

NTP.norm.method
: Method of normalization in NTP, either c("none", "row.std")

mc.cores
: The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

CellrangerCount *CellrangerCount*

Description

A wrapper function to run Cellranger Count process

Usage

```
CellrangerCount(
  cellranger.path,
  fastq.dir,
  cellranger.ref.dir,
  output.dir,
  sample.name,
  run.cmd = TRUE,
  mc.cores = 1
)
```

Arguments

cellranger.path
: Cell Ranger program path

fastq.dir
: FastQC output directory

cellranger.ref.dir
: Directory of Cell Ranger reference file

output.dir
: Output directory

sample.name
: sample name

run.cmd
: Whether to execute the command line (default=TRUE)

mc.cores
: The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

CellrangerCount takes FASTQ files from fastQC and performs alignment, filtering, barcode counting, and UMI counting.

Value

feature-barcode matrices and Secondary analysis (e.g., dimensionality reduction, cell clustering, and differential expression)

References

Massively parallel digital transcriptional profiling of single cells. GXY Zheng. (2017).

See Also

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/using/count

---

`cell_type_NTP` *cell_type_NTP*

Description

A function to run NTP to infer cell type

Usage

```
cell_type_NTP(
  seurat,
  wd,
  markerList,
  assay = "RNA",
  slot = c("scale.data", "count.data", "data"),
  output.dir = "./",
  rda.dir = "./data",
  NTP.g.filter.method = c("sd", "mad", "none"),
  NTP.gene.filter.cutoff = 0.3,
  NTP.distance = c("cosine", "correlation"),
  NTP.norm.method = c("none", "row.std"),
  mc.cores = 1
)
```

Arguments

`seurat`	Seurat object
`wd`	working directory
`markerList`	List of cell type marker
`assay`	Assay to use
`slot`	seurat object expression data, c("scale.data", "count.data", "data")()
`output.dir`	output directory

rda.dir	Path of the RData saving directory
NTP.g.filter.method	Method of gene filtering in NTP c(sd (Default), mad, none)
NTP.gene.filter.cutoff	Cut-off score of standard deviation in NTP
NTP.distance	Method of calculating distance in NTP, either c("correlation" or "cosine").
NTP.norm.method	Method of normalization in NTP, either c("none", "row.std")
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

cell type annotation using NTP

Value

Seurat object

`cnv.distribution` *cnv.distribution*

Description

Visualize cnv distribution of scTyper

Usage

```
cnv.distribution(
  seurat,
  wd,
  marker = "Puram.2017.HNSCC.TME",
  slot = c("scale.data", "count.data", "data")
)
```

Arguments

seurat	seurat object
wd	working directory
marker	cell type marker
slot	assay data type of seurat object, c("scale.data", "count.data", "data")

draw.heatmap *draw.heatmap*

Description

Visualize heatmap of scTyper

Usage

```
draw.heatmap(
  seurat,
  wd,
  run.inferCNV = TRUE,
  slot = c("scale.data", "count.data", "data"),
  marker = "Puram.2017.HNSCC.TME"
)
```

Arguments

seurat	seurat object
wd	working directory
run.inferCNV	whether run inferCNV (default=TRUE)
slot	assay data type of seurat object, c("scale.data", "count.data", "data")
marker	cell type marker

fastqc *fastqc*

Description

A wrapper function to run fastQC

Usage

```
fastqc(
  fastqc.path,
  fastq.dir,
  sample.name,
  fq1.idx = "_R1_001.fastq",
  fq2.idx = "_R2_001.fastq",
  output.dir,
  run.cmd = TRUE,
  mc.cores = 1
)
```

Arguments

fastqc.path	FastQC program path
fastq.dir	FastQC output directory
sample.name	sample name
fq1.idx	Index of the FASTQ file (Read 1)
fq2.idx	Index of the FASTQ file (Read 2)
output.dir	Output directory
run.cmd	Whether to execute the command line (default=TRUE)
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

FastQC aims to provide a QC report that detects problems originating from either the sequencer or the starting library material.

Value

Quality check report for sequence data. (e.g., .html)

References

FastQC: a quality control tool for high throughput sequence data. Andrews S. (2010).

See Also

http://www.bioinformatics.babraham.ac.uk/projects/fastqc

---

get.markerList *get.markerList*

Description

get markerList from scTyper database

Usage

```
get.markerList(marker)
```

Arguments

marker	Signature_list or Signature name of scTyper db or User-defined list of marker genes

Value

marker list

loadTestData *loadTestData*

Description

load test data in 'extdata' directory

Usage

```
loadTestData()
```

Value

test.seurat

make.seurat.fdata *make.seurat.fdata*

Description

A function to make fdata using seurat object

Usage

```
make.seurat.fdata(seurat, gene.ref.gtf, rda.dir)
```

Arguments

seurat	Seurat object
gene.ref.gtf	gene reference gtf file
rda.dir	rData directory

Details

make feature data using seurat

Value

feature data

`malignant.cellTyper` *malignant.cellTyper*

Description

A function to malignant cell typing

Usage

```
malignant.cellTyper(
  seurat,
  rda.dir = "./data",
  malignant.cell.type = "Epithelial",
  feature.to.test = c("cell.type", "tissue.type"),
  cells.test_reference = "immune"
)
```

Arguments

`seurat`	Seurat object
`rda.dir`	rData directory
`malignant.cell.type`	Cell type to assign malignant cell
`feature.to.test`	features to test as reference
`cells.test_reference`	cells to test as reference

Details

classification of malignant and non malignant seurat object.

Value

Seurat object

`report` *report*

Description

A function to create a report

Usage

```
report(envList, qc.dir, output.dir)
```

Arguments

envList	R environment list
qc.dir	qc directory
output.dir	output directory

Details

Provides a report that summarizes the processing steps and visualized tables and plots for the processed results. The report file is automatically generated recording the workflows of the data processing steps, the options used in the processing, and the outcome results.

Value

pdf file include data processing result information

---

run.inferCNV *run.inferCNV*

Description

A wrapper function to run inferCNV

Usage

```
run.inferCNV(
  seurat,
  assay = "RNA",
  output.dir = "./",
  rda.dir = "./data",
  fdata,
  pheno_info = pheno.df,
  feature.to.test = c("tissue.type", "cell.type"),
  cells.test_reference = "Normal",
  cells.test_excluded = c("Epithelial"),
  fc.cutoff = 0.05,
  cutoff.gene.cluster = 0.05,
  min_mean_expr_cutoff = 0.1,
  window_length = 101,
  smooth_ends = TRUE,
  recenter_method = "median",
  ordered = FALSE,
  inv_log = TRUE,
  sd_amplifier = 1.5,
  bp = 1000000,
  sd.cut = 0.3,
  mc.cores = 1
)
```

Arguments

seurat	Seurat object
assay	Name of assay to pull data from seurat object
output.dir	output directory
rda.dir	rData directory
fdata	feature information
pheno_info	phenotype information
feature.to.test	featuest to test either "tissue.type" or "cell.type"
cells.test_reference	a vector containing the classifications of the reference (normal) cells to use for infering cnv
cells.test_excluded	cell type to exclude functional enrichment analysis
fc.cutoff	fold change cutoff
cutoff.gene.cluster	A cutoff P-value for filtering out the gene clusters (calculated from GO analysis)
min_mean_expr_cutoff	the minimum mean value allowed for a gene to be retained in the expression matrix.
window_length	length of window (number of genes) for the moving average
smooth_ends	perform smoothing at the ends of the chromosomes (default:TRUE)
recenter_method	method to select the center of the cell expression value. (default:'mean', option:'mean', 'median')
ordered	order bool, default FALSE. Sort descending vs. ascending
inv_log	mean values will be determined based on (2^x -1)
sd_amplifier	multiplicative factor applied to the standard deviation to alter the noise range (default: 1.5)
bp	base pair
sd.cut	standard deviation cutoff
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

CNV inference

Value

Seurat object

`run.seurat.process` *run.seurat.process*

Description

A wrapper function to run seurat normalization process

Usage

```
run.seurat.process(
  count.dir = count.dir,
  rda.dir = rda.dir,
  project = proj.name,
  metrics_summary,
  sample.name = sample.name,
  percent.min.cells = 0.01,
  min.features = 200,
  scale.factor = 10000,
  vars.to.regress = c("nCount_RNA", "percent.mt"),
  selection.method = "vst",
  more_nFeature_RNA = 200,
  Less_nFeature_RNA = 8000,
  percent.mt = 10,
  normalize = TRUE,
  assay = "RNA",
  dims = 1:10,
  resolution = seq(0.6, 2, 0.2),
  random.seed = 1234
)
```

Arguments

`count.dir`	Ouput directory of cellragner count
`rda.dir`	Path of the RData saving directory
`project`	project name
`metrics_summary`	cellranger summary metrics
`sample.name`	Sample name
`percent.min.cells`	Include genes with detected expression in at least this many cells. Will subset the raw.data matrix as well. To reintroduce excluded genes, create a new object with a lower cutoff.
`min.features`	Include cells where at least this many genes are detected.
`scale.factor`	Sets the scale factor for cell-level normalization(10,000 by default)

`vars.to.regress`
: Variables to regress out (previously latent.vars in RegressOut). For example, nUMI, or percent.mito.

`selection.method`
: How to choose top variable features. Choose one of 'vst', 'mean.var.plot', 'dispersion'

`more_nFeature_RNA`
: High cutoffs for filtering cells that have unique feature counts (default is 200)

`Less_nFeature_RNA`
: low cutoffs for filtering cells that have unique feature counts (default is 8000)

`percent.mt`
: low cutoffs for filtering cells that have >n percent mitochondrial counts (default is 10)

`normalize`
: use log normalization

`assay`
: Assay to use

`dims`
: A vector of the dimensions to use in construction of the SNN grouph.

`resolution`
: Value of the resolution parameter, use a value above (below) 1.0 if you want to obtain a larger (smaller) number of communities.

`random.seed`
: Seed of the random number generator.

Details

CellrangerCount takes FASTQ files from fastQC and performs alignment, filtering, barcode counting, and UMI counting.

Value

feature-barcode matrices and Secondary analysis (e.g., dimensionality reduction, cell clustering, and differential expression)

References

Massively parallel digital transcriptional profiling of single cells. GXY Zheng. (2017).

See Also

https://satijalab.org/seurat/

`scTyper` *scTyper*

Description

The main scTyper function

Usage

```
scTyper(
  seurat.object = NULL,
  marker = "Puram.2017.HNSCC.TME",
  wd = getwd(),
  output.name = "test.result",
  pheno.fn,
  qc = FALSE,
  run.cellranger = FALSE,
  norm.seurat = FALSE,
  cell.typing.method = c("NTP", "ES", "Average"),
  level = c("cell", "cluster"),
  run.inferCNV = TRUE,
  proj.name = "scTyper",
  fastqc.path = NULL,
  fastq.dir = NULL,
  fq1.idx = "_R1_001.fastq",
  fq2.idx = "_R2_001.fastq",
  cellranger.path = NULL,
  cellranger.ref.dir = NULL,
  percent.min.cells = 0.1,
  min.features = 200,
  percent.mt = 10,
  vars.to.regress = c("nCount_RNA", "percent.mt"),
  dims = 1:10,
  resolution = 2,
  slot = c("scale.data", "count.data", "data"),
  assay = "RNA",
  NTP.g.filter.method = c("sd", "mad", "none"),
  NTP.gene.filter.cutoff = 0.3,
  NTP.distance = c("cosine", "correlation"),
  NTP.norm.method = c("none", "row.std"),
  gene.ref.gtf = NULL,
  feature.to.test = c("tissue.type", "cell.type"),
  cells.test_excluded = "Epithelial",
  cells.test_reference = "immune",
  fc.cutoff = 0.05,
  cutoff.gene.cluster = 0.05,
  malignant.cell.type = "Epithelial",
  report.mode = TRUE,
  mc.cores = 1
)
```

Arguments

seurat.object	Seurat object, if users have pre-processed seurat object, user have to insert seurat object as input
marker	Cell markers to use in cell typing, character or List (#identifier or #StudyName

	or User defined gene list)
wd	Working directory
output.name	Output directory name
pheno.fn	Phenotype file path
qc	Whether to execute FASTQC (default=FALSE)
run.cellranger	whether to excute cellranger count (default=FALSE)
norm.seurat	whether to normalize seurat object (default=FALSE)
cell.typing.method	cell typing method, c("NTP", "ES", "Average"), (default = "NTP")
level	Indicate the cell assignment level (cell or cluster)
run.inferCNV	Indicate whether 'malignant cell typing by inferCNV process run
proj.name	Project name
fastqc.path	FastQC program path
fastq.dir	FastQC output directory
fq1.idx	Index of the FASTQ file (Read 1)
fq2.idx	Index of the FASTQ file (Read 2)
cellranger.path	Cell Ranger program path
cellranger.ref.dir	Directory of Cell Ranger reference file
percent.min.cells	Cutoff to filter features containing minimum percent of cells
min.features	Cutoff to filter cells containing minimum number of features
percent.mt	Cutoff for filtering cells that have >n percent mitochondrial counts
vars.to.regress	Variables to regress out
dims	A vector of the dimensions to use in construction of the SNN graph.
resolution	Value of the resolution parameter, use a value above (below) 1.0 if you want to obtain a larger (smaller) number of communities.
slot	Data type of Seurat object, c("scale.data", "count.data", "data")
assay	Assay of Seurat object
NTP.g.filter.method	Method to filter genes in NTP
NTP.gene.filter.cutoff	Cutoff to filter genes of in NTP
NTP.distance	NTP distance method, a character, either c("correlation" or "cosine").
NTP.norm.method	NTP normalization method, either c("none", "row.std")
gene.ref.gtf	Path of GTF file including genomic location for genes

`feature.to.test`
: Column header name of the meta data in Seurat object (select the cell groups for T.test) either "tissue.type" or "cell.type"

`cells.test_excluded`
: A value indicates the cells to be excluded in T.test

`cells.test_reference`
: A value indicates the cells to use as be excluded in T.test

`fc.cutoff`
: Cutoff of fold change

`cutoff.gene.cluster`
: A cutoff P-value for filtering out the gene clusters (calculated from GO analysis)

`malignant.cell.type`
: Cell type to assign malignant cell

`report.mode`
: Generate report file

`mc.cores`
: The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Value

Seurat object

`update.sig.db` *update.sig.db*

Description

Update sig.db

Usage

```
update.sigTyper.db(sig.db.path, db.name=c("CellMarker", "sigTyper.db"), output.dir=system.file("/dat
```

Arguments

`sig.db.path`
: Path of sig.db.txt

`db.name`
: database name to update. either c("sigTyper.db", "CellMarker")

`output.dir`
: storage path of sig.db marker

Index