Package 'scTyper'

June 25, 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 sscTyper 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 7.1.0

VignetteBuilder knitr

R topics documented:

`cell.filter.seurat` *cell.filter.seurat*

Description

A wrapper function to cell.filter.seurat

Usage

```
cell.filter.seurat(seurat, sample.name, metrics_summary, more_nFeature_RNA = 200, Less_nFeature_RNA =
```

Arguments

seurat	Seurat object
sample.name	single cell RNA sequening sample name
metrics_summary	summary metrics
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 2500)
percent.mt	low cutoffs for filtering cells that have >n percent mitochondrial counts (default is 5)

Details

Creates a Seurat object containing only a subset of the cells in the original object.

Value

a Seurat object containing only the relevant subset of cells

---

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

Description

A wrapper function to cell.typing.seurat

Usage

```
cell.typing.seurat(seurat, marker="Puram.2017.HNSCC", cell.typing.method, level, wd, slot, assay='RNA
```

Arguments

seurat	Seurat object
marker	Which markers to use cell typing
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 CellrangerCount

Usage

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

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 wrapper function to cell_type_NTP

Usage

```
cell_type_NTP(seurat, wd, markerList, assay="RNA", slot=c("scale.data", "count.data", "data"), output.
```

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 result 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")

`cts.geneSetCluster` *cts.geneSetCluster*

Description

A wrapper function to cts.geneSetCluster

Usage

```
cts.geneSetCluster(cset, rda.dir, perm.t.resList, fc=0.05, bp = 1000000)
```

Arguments

`cset`	cnv Set
`rda.dir`	rData directory
`perm.t.resList`	list of t-test result
`bp`	base pair
`fc`	fold change

Details

get gene cluster

Value

cell type specific geneClustList

cts.GO *cts.GO*

Description

A wrapper function to cts.GO

Usage

```
cts.GO(cell.type.set, rda.dir,cts.geneClustList)
```

Arguments

cell.type.set	cell type set
rda.dir	rData directory
cts.geneClustList	cell type specific geneClustList

Details

Interface to the g:Profiler tool for finding enrichments in gene lists.

Value

gprofiler result List

df2gr *df2gr*

Description

A wrapper function to make dataframe to GRange

Usage

```
df2gr(df, seqnames, start, end, strand)
```

Arguments

df	dataframe
seqnames	A character vector of recognized names for the column in df that contains the chromosome name (sequence name) associated with each genomic range.
start	A character vector of recognized names for the column in df that contains the start positions of the genomic ranges.

end	A character vector of recognized names for the column in df that contains the end positions of the genomic ranges.
strand	A character vector of recognized names for the column in df that contains the strand associated with each genomic range.

Details

the workhorse behind the coercion method from data.frame to GRanges.

Value

GRanges object

`draw.heatmap` *draw.heatmap*

Description

Visualize result heatmap of scTyper

Usage

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

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.
```

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

---

fastqc.summary *fastqc.summary*

Description

A wrapper function to make fastqc dataframe

Usage

```
fastqc.summary(fastqc.df, fq1.idx="_R1_001.fastq", fq2.idx="_R2_001.fastq")
```

Arguments

fastqc.df	fastqc output directory
fq1.idx	Index of the FASTQ file (Read 1)
fq2.idx	Index of the FASTQ file (Read 2)

Details

make table of fastqc using qc outputs.

Value

fastqc dataframe

---

fastqc.table *fastqc.table*

Description

A wrapper function to make fastqc dataframe

Usage

```
fastqc.table(qc.dir)
```

Arguments

qc.dir	fastqc output directory

Details

make table of fastqc using qc outputs.

Value

fastqc dataframe

---

fil.infercnv_obj *fil.infercnv_obj*

Description

A wrapper function to fil.infercnv_obj

Usage

```
fil.infercnv_obj(infercnv_obj, cset, rda.dir, gprofiler.resList, cell.type.set, cts.geneClustList)
```

Arguments

infercnv_obj	infercnv object
cset	cnv Set
rda.dir	rData directory
gprofiler.resList	gprofiler result List
cell.type.set	cell type set
cts.geneClustList	cell type specific geneClustList

Details

get gene cluster

Value

infercnv object

`get.geneClust` *get.geneClust*

Description

A wrapper function to get.geneClust

Usage

```
get.geneClust(gr, bp=1000000)
```

Arguments

gr	GRanges object
bp	base pair

Details

get gene cluster

Value

gene clust List

`get.markerList` *get.markerList*

Description

get markerList from scTyper databse

Usage

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

Arguments

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

Value

marker list

`get.qc.report` *get.qc.report*

Description

QC Reports

Usage

```
get.qc.report(qc.dir)
```

Arguments

`qc.dir`	qc directory

Details

Provides FASTQC report that summarizes the QC processing steps

GSEA.EnrichmentScore2 *GSEA.EnrichmentScore2*

Description

Run GSEA(Gene Set Enrichment Analysis)

Usage

```
GSEA.EnrichmentScore2(gene.list, gene.set, weighted.score.type = 1, correl.vector = NULL)
```

Arguments

gene.list	gene signature list
gene.set	gene set
weighted.score.type	Type of weighted score
correl.vector	correlation vector

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

infercnv2cset *infercnv2cset*

Description

A wrapper function to infercnv2cset

Usage

```
infercnv2cset(infercnv_obj, pdata)
```

Arguments

infercnv_obj	an infercnv object
pdata	phenotype data

Details

Creation of cset using infercnv object.

Value

cset

invalid *invalids*

Description

Test if a value is missing, empty, or contains only NA or NULL values.

Usage

```
invalid(x)
```

Arguments

x	value to be tested

Value

Bool

`list2matrix` *list2matrix*

Description

convert list to matrix

Usage

```
list2matrix(List)
```

Arguments

`List`	list

`make.color.set` *make.color.set*

Description

Get random colors.

Usage

```
make.color.sample(n, col.label)
```

Arguments

`n`	the number of color sample, integer
`col.label`	color label, character

`make.eset` *make.eset*

Description

For the expression data are transformed to a file with extension .eSet

Usage

```
make.eset(expr, pdata=NULL, fdata=NULL, verbose=TRUE)
```

Arguments

expr	expression data
pdata	phenotype data
fdata	feature data
verbose	default TRUE, Class to writing verbose messages to a connection or file.

Details

store expression data in ExpressionSet format for convenient analysis

Value

expression set

---

make.seurat *make.seurat*

Description

A wrapper function to make.seurat

Usage

```
make.seurat(count.dir, sample.name = sample.name, project = "SeuratProject", min.cells=0, min.features
```

Arguments

count.dir	Path of the cellranger count directory
sample.name	single cell RNA sequening sample name
project	project name(string)
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.

Details

Initializes the Seurat object and some optional filtering

Value

Seurat object serves as a container that contains both data (like the count matrix) and analysis (like PCA, or clustering results) for a single-cell dataset.

`make.seurat.eset` *make.seurat.eset*

Description

A wrapper function to make.seurat.eset

Usage

```
make.seurat.eset(seurat, slot=c("scale.data", "count.data", "count.data"), fdata, output.dir=NULL, sa
```

Arguments

`seurat`	Seurat object
`slot`	assay data type of seurat object, c("scale.data", "count.data", "data")
`fdata`	feature data
`output.dir`	output directory
`save`	whether save

Details

make expression set using seurat

Value

expression set

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

Description

A wrapper function to make.seurat.fdata

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

`make.stat_summary` *make.stat_summary*

Description

A wrapper function to run make.stat_summary

Usage

```
make.stat_summary(count.dir, sample.name, output.dir, pheno.df)
```

Arguments

`count.dir`	cellragner count ouput directory
`sample.name`	cell sample name
`output.dir`	Output directory
`pheno.df`	phenotype dataframe(reference an instruction manual)

Details

make data summary file

Value

data summary file

`malignant.cellTyper` *malignant.cellTyper*

Description

A wrapper function to malignant.cellTyper

Usage

```
malignant.cellType(seurat, rda.dir, malignant.cell.type, feature.to.test, cells.test_reference)
```

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

`NTP` *NTP*

Description

A wrapper function to NTP

Usage

```
NTP(eset, sigList, out.dir, output.name, dist.selection, norm.method, nresmpl, rnd.seed, mc.cores)
```

Arguments

eset	expressio
sigList	gene signiture list
out.dir	output directory
dist.selection	calculating distance, a character, either c("correlation" or "cosine").
norm.method	normalization method, either c("none", "row.std")
nresmpl	an integer, number of permutations for p-value
rnd.seed	Seed of the random number generator.
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Nearest Template Prediction (NTP) based on predefined class templates.

Value

Nearest Template Prediction (NTP) result

References

Hoshida, Y. (2010). Nearest Template Prediction: A Single-Sample-Based Flexible Class Prediction with Confidence Assessment. PLoS ONE 5, e15543.

---

perm.subcset.t *perm.subcset.t*

Description

A wrapper function to perm.subcset.t

Usage

```
perm.subcset.t(cset, cell.type.set, rda.dir, ordered= FALSE, levels= c(1,0), mc.cores=5)
```

Arguments

cset	cnv Set
cell.type.set	cell type set
rda.dir	rData directory
ordered	order bool, default FALSE. Sort descending vs. ascending
levels	confidence level of the interval.
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

Details

Performs t-tests on cnv subset

Value

list of t-test result

`perm.ttest` *perm.ttest*

Description

Run permutation T-test

Usage

```
perm.ttest(eset, g.st, level=NULL, t.test=F, permp=T, permp.exact=NULL, ordered=T, mc.cores=1,...)
```

Arguments

`eset`	expression set
`g.st`	group subset
`permp`	bool, defalut TRUE. Calculating permuted T test p-values or not
`ordered`	order bool, default FALSE. Sort descending vs. ascending
`mc.cores`	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.
`levels`	confidence level of the interval.

Details

Performs sample t-tests on vectors of data.

Value

t test result

preRanked.GSEA *preRanked.GSEA*

Description

a universal gene set enrichment analysis tools

Usage

```
preRanked.GSEA(expr, SIGDB, weighted.score.type = 0, correl.vector = NULL, n.cutoff=1,mc.cores=1)
```

Arguments

expr	expression metrix
SIGDB	gene signature list
weighted.score.type	Type of weight score
correl.vector	correlation vector
n.cutoff	number of cutoff
mc.cores	The number of cores to use. Must be at least one(default=1), and parallelization requires at least two cores.

report *report*

Description

Reports the result of using scTyper()

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, output.dir, rda.dir, fdata, pheno_info = pheno.df, feature.to.test, cells.
```

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 run.seurat.process

Usage

```
run.seurat.process(count.dir = count.dir, rda.dir = rda.dir, project = proj.name, metrics_summary, samp
```

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).

`scTyper` *scTyper*

Description

Run scTyper

Usage

```
scTyper(seurat.object, marker, wd, output.name, pheno.fn, qc = FALSE, run.cellranger=FALSE, norm.seura
```

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.seqnames` *update.seqnames*

Description

updata the column in df that contains the chromosome name (sequence name)

Usage

```
update.seqnames(seqnames)
```

Arguments

`seqnames`
: A character vector of recognized names for the column in df that contains the chromosome name (sequence name) associated with each genomic range.

Value

sequence name

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