

Package: SRTsim (via r-universe)

August 22, 2024

Type Package

Title Simulator for Spatially Resolved Transcriptomics

Version 0.99.7

Date 2024-08-14

Maintainer Jiaqiang Zhu <jiaqiang@umich.edu>

Description An independent, reproducible, and flexible Spatially Resolved Transcriptomics (SRT) simulation framework that can be used to facilitate the development of SRT analytical methods for a wide variety of SRT-specific analyses. It utilizes spatial localization information to simulate SRT expression count data in a reproducible and scalable fashion. Two major simulation schemes are implemented in 'SRTsim': reference-based and reference-free.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports

concaveman,sf,sp,spatstat.geom,parallel,pdist,MASS,S4Vectors,stats,Matrix,Morpho,matrixStats,plotly,shiny, viridis,shiny
ggplot2, ggpubr, DT, spatstat.random, magrittr, FNN, dplyr, magick

Suggests knitr,rmarkdown,BiocStyle,RefManageR,BiocManager,sessioninfo

VignetteBuilder knitr

Depends R (>= 3.5.0), methods

NeedsCompilation no

Collate 'SRTaffine.R' 'SRTcount.R' 'SRTfit.R' 'SRTmodels.R'
'SRTsim_class.R' 'body.R' 'compareSRT.R' 'createSRT.R' 'data.R'
'globals.R' 'reGenCountshiny.R' 'runapp.R' 'server.R'
'shiny2srt.R' 'sidebar.R' 'simSRTLlocs.R' 'subsetSRT.R'
'SRTcci.R' 'ui.R' 'utilies_func.R' 'visualize_gene.R'
'visualize_metrics.R'

Author Jiaqiang Zhu [aut, ctb, cre]
 (<<https://orcid.org/0000-0002-1455-0041>>), Lulu Shang [aut]
 (<<https://orcid.org/0000-0002-2480-3065>>), Xiang Zhou [aut]
 (<<https://orcid.org/0000-0002-4331-7599>>)

Repository CRAN

Date/Publication 2024-08-21 08:10:05 UTC

Contents

compareSRT	3
convert_grid	3
createSRT	4
EstParam	5
exampleLIBD	5
fit_pos_optim	6
get_metrics_pd	6
get_stats_gene	7
get_stats_loc	7
metaParam	8
refcolData	8
refCounts	9
refrowData	10
reGenCountshiny	10
Shiny2SRT	11
simcolData	12
simCounts	12
simNewLocs	13
simrowData	14
srtsim_cci_free	14
srtsim_cci_ref	15
srtsim_count	16
srtsim_count_affine	18
srtsim_fit	19
srtsim_newlocs	20
SRTsim_shiny	21
subsetSRT	21
toyData	22
toyShiny	23
visualize_gene	23
visualize_metrics	24

Index **26**

compareSRT	<i>Summarize metrics for reference data and synthetic data</i>
------------	--

Description

Summarize metrics for reference data and synthetic data

Usage

```
compareSRT(simsrt)
```

Arguments

simsrt A SRTsim object

Value

Returns an object with summarized metrics

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
## Compute metrics
toySRT <- compareSRT(toySRT)
```

convert_grid	<i>Convert continuous coordinate into integer, essential for BayesSpace to determine the neighborhood info</i>
--------------	--

Description

Convert continuous coordinate into integer, essential for BayesSpace to determine the neighborhood info

Usage

```
convert_grid(x)
```

Arguments

x A numeric vector of continuous coordinate

Value

Returns a numeric vector of integer coordinate

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)

## Create New Locations within Profile
toySRT2 <- srtsim_newlocs(toySRT,new_loc_num=1000)

## Convert non-integer x-coordinates into an integer value
newGrid_x <- convert_grid(simcolData(toySRT2)$x)
```

createSRT	<i>Create simSRT object</i>
-----------	-----------------------------

Description

Create simSRT object

Usage

```
createSRT(count_in, loc_in, refID = "ref1")
```

Arguments

count_in	A gene expression count matrix
loc_in	A location dataframe with colnames x,y,label
refID	A character reference sample identifier. Default = ref1.

Value

Returns a spatialExperiment-based object

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)

## Explore the object
toySRT
```

EstParam	<i>Access Model Fitting Parameters</i>
----------	--

Description

Access Model Fitting Parameters

Usage

```
EstParam(x)
```

Arguments

x SRTsim object

Value

Returns a list of estimated parameters by fitting models

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
EstParam(toySRT)
```

exampleLIBD	<i>Data used for creating vignettes</i>
-------------	---

Description

A data list containing the a gene expression matrix and a location dataframe

Usage

```
exampleLIBD
```

Format

A data list

example_count A sparse matrix with 80 rows and 3611 columns.

example_loc A data frame with 3611 rows and 6 columns.

Source

created based on a SpatialLIBD data (SampleID: 151673) to serve as an example

Examples

```
data(exampleLIBD)      #Lazy loading. Data becomes visible as soon as called
```

```
fit_pos_optim          fitting data with poisson through optim function
```

Description

fitting data with poisson through optim function

Usage

```
fit_pos_optim(x, maxiter = 500)
```

Arguments

x	A vector of count values to be fitted
maxiter	number of iteration

Value

Returns a vector with mean paramter lambda, loglikelihood value llk, convergence

```
get_metrics_pd          Extracted summarized metrics for reference data and synthetic data
```

Description

Extracted summarized metrics for reference data and synthetic data

Usage

```
get_metrics_pd(simsrt, metric = "GeneMean")
```

Arguments

simsrt	A SRTsim object
metric	Specification of metrics to be plotted.

Value

Returns a dataframe for ggplot

get_stats_gene	<i>Summarize gene-wise summary metrics</i>
----------------	--

Description

Summarize gene-wise summary metrics

Usage

```
get_stats_gene(mat, group, log_trans = TRUE)
```

Arguments

mat	A count matrix
group	A group label
log_trans	A logical constant indicating whether to log transform the gene mean and variance

Value

Returns a n by 5 dataframe with location metrics

get_stats_loc	<i>Summarize location-wise summary metrics</i>
---------------	--

Description

Summarize location-wise summary metrics

Usage

```
get_stats_loc(mat, group, log_trans = TRUE)
```

Arguments

mat	A count matrix
group	A group label
log_trans	A logical constant indicating whether to log transform the libsize

Value

Returns a n by 3 dataframe with location metrics

metaParam	<i>Access User-Specified Parameters</i>
-----------	---

Description

Access User-Specified Parameters

Usage

```
metaParam(x)
```

Arguments

x SRTsim object

Value

Returns a list of user-specified parameters

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
metaParam(toySRT)
```

refcolData	<i>Access reference colData</i>
------------	---------------------------------

Description

Access reference colData

Usage

```
refcolData(x)
```

Arguments

x SRTsim object

Value

Returns the colData of reference data

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
refcolData(toySRT)
```

refCounts	<i>Access reference count matrix</i>
-----------	--------------------------------------

Description

Access reference count matrix

Usage

```
refCounts(x)
```

Arguments

x SRTsim object

Value

Returns a reference count matrix

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
refCounts(toySRT)[1:3,1:3]
```

refrowData	<i>Access reference rowData</i>
------------	---------------------------------

Description

Access reference rowData

Usage

```
refrowData(x)
```

Arguments

x SRTsim object

Value

Returns the rowData of reference data

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
refrowData(toySRT)
```

reGenCountshiny	<i>ReSimulate Count Data with Parameters Specification from Shiny</i>
-----------------	---

Description

ReSimulate Count Data with Parameters Specification from Shiny

Usage

```
reGenCountshiny(shinyOutput, NewSeed = NULL)
```

Arguments

shinyOutput A list of Shiny Output. Including a simCount, simInfo,simcountParam,simLocParam
NewSeed A new seed for data generation. Useful when multiple replicates are needed.

Value

Returns a Count DataFrame

Examples

```
## Re-generate Count Data based on ShinyOutput Parameters, should be same as simCount in ShinyOutput
cMat <- reGenCountshiny(toyShiny)

## Generate Count Data with A New Seed based on ShinyOutput Parameters
cMat2 <- reGenCountshiny(toyShiny,NewSeed=2)

## Comparison across the output
toyShiny$simCount[1:3,1:3]
cMat[1:3,1:3]
cMat2[1:3,1:3]
```

Shiny2SRT

Create a SRTsim object from reference-free shinyoutput

Description

Create a SRTsim object from reference-free shinyoutput

Usage

```
Shiny2SRT(shinyOutput)
```

Arguments

shinyOutput A list of Shiny Output. Including a simCount, simInfo,simcountParam,simLocParam

Value

Returns a SRTsim object with user-specified parameters stored in metaParam slot.

Examples

```
shinySRT <- Shiny2SRT(toyShiny)

## Explore the new SRT object
shinySRT@metaParam
shinySRT@simCounts[1:3,1:3]
shinySRT@simcolData
```

simcolData	<i>Access synthetic colData</i>
------------	---------------------------------

Description

Access synthetic colData

Usage

```
simcolData(x)
```

Arguments

x SRTsim object

Value

Returns the colData of synthetic data

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount, loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT, sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
simcolData(toySRT)
```

simCounts	<i>Access synthetic count matrix</i>
-----------	--------------------------------------

Description

Access synthetic count matrix

Usage

```
simCounts(x)
```

Arguments

x SRTsim object

Value

Returns a synthetic count matrix

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
simCounts(toySRT)[1:3,1:3]
```

simNewLocs

Fit the marginal distributions for single gene

Description

Fit the marginal distributions for single gene

Usage

```
simNewLocs(newN, lay_out = c("grid", "random"), preLoc)
```

Arguments

newN	A integer specifying the number of spatial locations in the synthetic data
lay_out	A character string specifying arrangement of new generated spatial locations. Default is "grid"
preLoc	A data frame of shape n by 3 that x, y coordinates and domain label

Value

Returns a n by 2 dataframe with newly generated spatial locations

simrowData	<i>Access synthetic rowData</i>
------------	---------------------------------

Description

Access synthetic rowData

Usage

```
simrowData(x)
```

Arguments

x SRTsim object

Value

Returns the rowData of synthetic data

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount, loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT, sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)
simrowData(toySRT)
```

srtsim_cci_free	<i>Generate Data with Cell-Cell Interaction Under Reference-Free Mode</i>
-----------------	---

Description

Generate Data with Cell-Cell Interaction Under Reference-Free Mode

Usage

```
srtsim_cci_free(
  zero_prop_in = 0,
  disper_in = Inf,
  mu_in = 1,
  numGene = 1000,
  location_in,
```

```

    region_cell_map,
    fc = 3,
    LR_in,
    sim_seed = 1,
    numKNN = 4,
    numSingleCellType = 2000
)

```

Arguments

zero_prop_in	A number specifying zero proportion for the count model, default is 0
disper_in	A number specifying dispersion for the count model, default is Inf. Same as the size parameter in rnbinom.
mu_in	A number specifying mean for background count model, default is 1
numGene	An integer specifying the number of genes in the synthetic data, default is 1000
location_in	A dataframe with x, y, and region_label
region_cell_map	A dataframe specifying the cell type proportion in each region. Row: region, Column: cell type.
fc	A number specifying effect size for ligand-receptor pairs that mediate the cell-cell communication, default is 3
LR_in	A dataframe specifying ligand and receptor pairs, containing four columns: protein_a, protein_b, celltypeA, and celltype B
sim_seed	A number for reproducible purpose
numKNN	A number specifying number of nearest neighbors with elevated gene expression levels, default is 4
numSingleCellType	A number specifying number of spots in the background pool. Gene expression count are then sampled from this background pool.

Value

Returns a SRTsim object with a newly generated count matrix and corresponding parameters

srtsim_cci_ref	<i>Generate Data with Cell-Cell Interaction Under Reference-Based Mode</i>
----------------	--

Description

Generate Data with Cell-Cell Interaction Under Reference-Based Mode

Usage

```

srtsim_cci_ref(
  EstParam = NULL,
  numGene = 1000,
  location_in,
  region_cell_map,
  fc = 3,
  LR_in,
  sim_seed = 1,
  numKNN = 4,
  numSingleCellType = 2000
)

```

Arguments

EstParam	A list of estimated parameters from srtsim_fit function, EstParam slot if the simSRT object.
numGene	An integer specifying the number of genes in the synthetic data, default is 1000
location_in	A dataframe with x, y, and region_label
region_cell_map	A dataframe specifying the cell type proportion in each region. Row: region, Column: cell type.
fc	A number specifying effect size for ligand-receptor pairs that mediate the cell-cell communication, default is 3
LR_in	A dataframe specifying ligand and receptor pairs, containing four columns: protein_a, protein_b, celltypeA, and celltype B
sim_seed	A number for reproducible purpose
numKNN	A number specifying number of nearest neighbors with elevated gene expression levels, default is 4
numSingleCellType	A number specifying number of spots in the background pool. Gene expression count are then sampled from this background pool.

Value

Returns a SRTsim object with a newly generated count matrix and corresponding parameters

srtsim_count

Generate Data with Estimated Parameters

Description

Generate Data with Estimated Parameters

Usage

```
srtsim_count(
  simsrt,
  breaktie = "random",
  total_count_new = NULL,
  total_count_old = NULL,
  rrr = NULL,
  nn_num = 5,
  nn_func = c("mean", "median", "ransam"),
  numCores = 1,
  verbose = FALSE
)
```

Arguments

simsrt	A object with estimated parameters from fitting step
breaktie	A character string specifying how ties are treated. Same as the "tie.method" in rank function
total_count_new	The (expected) total number of reads or UMIs in the simulated count matrix.
total_count_old	The total number of reads or UMIs in the original count matrix.
rrr	The ratio applies to the gene-specific mean estimate, used for the fixing average sequencing depth simulation. Default is null. Its specification will override the specification of total_count_new and total_count_old.
nn_num	A integer of nearest neighbors, default is 5.
nn_func	A character string specifying how the psedo-count to be generated. options include 'mean','median' and 'ransam'.
numCores	The number of cores to use
verbose	Whether to show running information for srtsim_count

Value

Returns a SRTsim object with a newly generated count matrix

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")
## Generate synthetic data with estimated parameters
toySRT <- srtsim_count(toySRT)

## Explore the synthetic count matrix
simCounts(toySRT)[1:3,1:3]
```

srtsim_count_affine *Generate Data with Estimated Parameters For A New Designed Pattern*

Description

Generate Data with Estimated Parameters For A New Designed Pattern

Usage

```
srtsim_count_affine(
  simsrt,
  rellabel,
  targetlabel,
  breaktie = "random",
  nn_func = c("mean", "median", "ransam"),
  nn_num = 5,
  local_sid = NULL,
  numCores = 1
)
```

Arguments

simsrt	A SRTsim object with estimated parameters from fitting step
rellabel	A character vector specifying labels for reference regions
targetlabel	A character vector specifying labels for target regions
breaktie	A character string specifying how ties are treated. Same as the "tie.method" in rank function
nn_func	A character string specifying how the psedo-count to be generated. options include 'mean','median' and 'ransam'.
nn_num	A integer of nearest neighbors, default is 5.
local_sid	A numeric seed used locally for the affine transformation. Default is NULL.
numCores	A number of cores to use

Value

Returns a SRTsim object with a newly generated count matrix

Examples

```
## Prepare Data From LIBD Sample
subinfo <- exampleLIBD$info[,c("imagecol", "imagerow", "layer")]
colnames(subinfo) <- c("x", "y", "label")
gns <- c("ENSG00000168314", "ENSG00000183036", "ENSG00000132639" )

## Create a simSRT Object with Three Genes For a Fast Example
```

```

simSRT1 <- createSRT(count_in= exampleLIBD$count[gns,],loc_in =subinfo)

## Estimate model parameters for data generation: domain-specific
simSRT1 <- srtsim_fit(simSRT1,sim_schem="domain")

## Define New Layer Structures
simSRT1@refcolData$target_label <- "NL1"
simSRT1@refcolData$target_label[simSRT1@refcolData$label %in% paste0("Layer",4:5)] <- "NL2"
simSRT1@refcolData$target_label[simSRT1@refcolData$label %in% c("Layer6","WM")] <- "NL3"

## Perform Data Generation for New Defined Layer Structures
## Reference: WM --> NL3, Layer5--> NL2, Layer3 --> NL1
simSRT1 <- srtsim_count_affine(simSRT1,
  rellabel=c("Layer3","Layer5","WM"),
  targetlabel=c("NL1","NL2","NL3"),
  nn_func="ransam"
)

## Visualize the Expression Pattern for Gene of Interest
visualize_gene(simsrt=simSRT1,plotgn = "ENSG00000168314",rev_y=TRUE,ptsizeCount=1)

```

srtsim_fit

Fit the marginal distributions for each row of a count matrix

Description

Fit the marginal distributions for each row of a count matrix

Usage

```

srtsim_fit(
  simsrt,
  marginal = c("auto_choose", "zinb", "nb", "poisson", "zip"),
  sim_scheme = c("tissue", "domain"),
  min_nonzero_num = 2,
  maxiter = 500
)

```

Arguments

simsrt	A SRTsim object
marginal	Specification of the types of marginal distribution. Default value is 'auto_choose' which chooses between ZINB, NB, ZIP, and Poisson by a likelihood ratio test (lrt), AIC and whether there is underdispersion. 'zinb' will fit the ZINB model. If there is underdispersion, it will fit the Poisson model. If there is no zero at all or an error occurs, it will fit an NB model instead. 'nb' fits the NB model and chooses between NB and Poisson depending on whether there is underdispersion. 'poisson' simply fits the Poisson model. 'zip' fits the ZIP model and chooses between ZIP and Poisson by a likelihood ratio test

sim_scheme a character string specifying simulation scheme. "tissue" stands for tissue-based simulation; "domain" stands for domain-specific simulation. Default is "tissue".
min_nonzero_num The minimum number of non-zero values required for a gene to be fitted. Default is 2.
maxiter The number of iterations for the model-fitting. Default is 500.

Value

Returns an object with estimated parameters

Examples

```

## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Estimate model parameters for data generation
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")

```

srtsim_newlocs *Fit the marginal distributions for each row of a count matrix*

Description

Fit the marginal distributions for each row of a count matrix

Usage

```

srtsim_newlocs(
  simsrt,
  new_loc_num = NULL,
  loc_lay_out = c("grid", "random"),
  voting_nn = 3
)

```

Arguments

simsrt A SRTsim object
new_loc_num A integer specifying the number of spatial locations in the synthetic data
loc_lay_out a character string specifying arrangement of new generated spatial locations. Default is "grid"
voting_nn A integer of nearest neighbors used in label assignment for new generated locations. Default is 3.

Value

Returns a object with estimated parameters

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount, loc_in = toyData$toyInfo)
set.seed(1)

## Create New Locations within Profile
toySRT2 <- srtsim_newlocs(toySRT, new_loc_num=1000)

## Explore New Generated Locations
simcolData(toySRT2)
```

SRTsim_shiny*Run the SRTsim Shiny Application*

Description

Run the SRTsim Shiny Application

Usage

```
SRTsim_shiny()
```

Value

A list that contains a count matrix, a location dataframe, and all parameter specifications.

Examples

```
## Not run:
## Will Load an Interactive Session
shinyOutput <- SRTsim_shiny()

## End(Not run)
```

subsetSRT*Subset SRT object based on domain labels of interest*

Description

Subset SRT object based on domain labels of interest

Usage

```
subsetSRT(simsrt, sel_label = NULL)
```

Arguments

`simsrt` A SRTsim object
`sel_label` A vector of selected domain labels used for the data generation

Value

Returns a `spatialExperiment`-based object

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount, loc_in = toyData$toyInfo)
set.seed(1)
## Only Keep the Spatial Locations labelled as "A" in the reference data
subtoySRT <- subsetSRT(toySRT, sel_label = "A")
```

<code>toyData</code>	<i>A toyExample to showcase reference-based simulations</i>
----------------------	---

Description

A data list containing the a gene expression matrix and a location dataframe

Usage

`toyData`

Format

A data list

toyCount A sparse matrix with 100 rows and 251 columns.

toyInfo A data frame with 251 rows and 3 columns.

Source

created based on a ST Human Breast Cancer data to serve as an example

Examples

```
data(toyData)            #Lazy loading. Data becomes visible as soon as called
```

toyShiny	<i>A toyExample to showcase reference-free simulations</i>
----------	--

Description

A list of shiny output

Usage

```
toyShiny
```

Format

A list of shiny output

simCount A data frame with 150 rows and 980 columns.

simInfo A data frame with 980 rows and 4 columns: x, y, group, foldchange

simcountParam A list of user-specified parameters for count generation

simLocParam A list of user-specified parameters for pattern design

Source

created based using the SRTsim_shiny()

Examples

```
data(toyShiny)      #Lazy loading. Data becomes visible as soon as called
```

visualize_gene	<i>Visualize expression pattern for the gene of interest in reference data and synthetic data</i>
----------------	---

Description

Visualize expression pattern for the gene of interest in reference data and synthetic data

Usage

```
visualize_gene(  
  simsrt,  
  plotgn = NULL,  
  ptsizeCount = 2,  
  textsizeCount = 12,  
  rev_y = FALSE,  
  virOption = "D",  
  virDirection = -1  
)
```

Arguments

simsrt	A SRTsim object
plotgn	A gene name selected for visualization
ptsizeCount	Specification of point size. Default is 2.
textsizeCount	Specification of axis font size. Default is 12.
rev_y	Logical indicating whether to reverse the y axis. Default is FALSE. Useful for Visualize the LIBD data.
virOption	Specification of option in the scale_color_viridis. Default is "D". User can choose a letter from 'A' to 'H'.
virDirection	Specification of direction in the scale_color_viridis. Default is "-1". User can choose '1' or '-1'.

Value

Returns two expression plots for the gene of interest

Examples

```
## Create a simSRT object
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)
set.seed(1)
## Create New Locations within Profile
toySRT2 <- srtsim_newlocs(toySRT,new_loc_num=1000)

## Estimate model parameters for data generation
toySRT2 <- srtsim_fit(toySRT2,sim_schem="tissue")

## Generate synthetic data with estimated parameters
toySRT2 <- srtsim_count(toySRT2,rrr=1)

## compare the expression pattern of HLA-B in synthetic data and reference data
visualize_gene(simsrt=toySRT2,plotgn = "HLA-B")
```

visualize_metrics	<i>Visualize summarized metrics for reference data and synthetic data</i>
-------------------	---

Description

Visualize summarized metrics for reference data and synthetic data

Usage

```
visualize_metrics(
  simsrt,
  metric_type = c("all", "genewise", "locwise", "GeneMean", "GeneVar", "GeneCV",
    "GeneZeroProp", "LocZeroProp", "LocLibSize"),
```



```
    colorpalette = "Set3",  
    axistextsize = 12  
  )
```

Arguments

<code>simsrt</code>	A SRTsim object
<code>metric_type</code>	Specification of metrics to be plotted. Default value is 'all', which will plot all six metrics: including four gene-wise metrics and two location-wise metrics. "genewise" will produce violin plots for all four gene-wise metrics; "location-wise" will produce violin plots for all two location-wise metrics; "GeneMean", "GeneVar", "GeneCV", "GeneZeroProp", "LocZeroProp", and "LocLibSize" will produce single violin plot for the corresponding metric.
<code>colorpalette</code>	Specification of color palette to be passed to <code>palette</code> in the <code>scale_fill_brewer</code> . Default is "Set3"
<code>axistextsize</code>	Specification of axis font size. Default is 12.

Value

Returns a list of ggplots

Examples

```
## Create a simSRT object  
toySRT <- createSRT(count_in=toyData$toyCount,loc_in = toyData$toyInfo)  
set.seed(1)  
  
## Estimate model parameters for data generation  
toySRT <- srtsim_fit(toySRT,sim_schem="tissue")  
  
## Generate synthetic data with estimated parameters  
toySRT <- srtsim_count(toySRT)  
  
## Compute metrics  
toySRT <- compareSRT(toySRT)  
  
## Visualize Metrics  
visualize_metrics(toySRT)
```

Index

* datasets

- exampleLIBD, 5
- toyData, 22
- toyShiny, 23

- compareSRT, 3
- convert_grid, 3
- createSRT, 4

- EstParam, 5
- exampleLIBD, 5

- fit_pos_optim, 6

- get_metrics_pd, 6
- get_stats_gene, 7
- get_stats_loc, 7

- metaParam, 8

- refcolData, 8
- refCounts, 9
- refrowData, 10
- reGenCountshiny, 10

- Shiny2SRT, 11
- simcolData, 12
- simCounts, 12
- simNewLocs, 13
- simrowData, 14
- srtsim_cci_free, 14
- srtsim_cci_ref, 15
- srtsim_count, 16
- srtsim_count_affine, 18
- srtsim_fit, 19
- srtsim_newlocs, 20
- SRTsim_shiny, 21
- subsetSRT, 21

- toyData, 22
- toyShiny, 23

- visualize_gene, 23
- visualize_metrics, 24