Package 'scpoisson'

August 17, 2022

Title Single Cell Poisson Probability Paradigm

Version 0.0.1

Description Useful to visualize the Poissoneity (an independent Poisson statistical framework, where each RNA measurement for each cell comes from its own independent Poisson distribution) of
Unique Molecular Identifier (UMI) based single cell RNA sequencing (scRNA-seq) data, and explore
cell clustering based on model departure as a novel data representation.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Suggests renv, testthat (>= 3.0.0), vdiffr, rmarkdown, knitr, qpdf

VignetteBuilder knitr

Config/testthat/edition 3

Imports ggplot2, glmpca, Seurat, magrittr, dplyr, tidyr, purrr, Matrix, Rdpack, SeuratObject, WGCNA, broom, stats, methods, matrixStats

RdMacros Rdpack

Depends R (>= 2.10)

NeedsCompilation no

Author Yue Pan [aut, cre],
Justin Landis [aut] (<https://orcid.org/0000-0001-5501-4934>),
Dirk Dittmer [aut],
James S. Marron [aut],
Di Wu [aut]

Maintainer Yue Pan <yuep027@gmail.com>

Repository CRAN

Date/Publication 2022-08-17 06:50:02 UTC

R topics documented:

`adj_CDF_logit` *A novel data representation based on Poisson probability*

Description

This function returns a matrix of a novel data representation with the same dimension as input data matrix.

Usage

```
adj_CDF_logit(data, change = 1e-10, ...)
```

Arguments

`data`	A UMI count data matrix with genes as rows and cells as columns or an S3 object for class 'scppp'.
`change`	A numeric value used to correct for exactly 0 and 1 before logit transformation. Any values below change are set to be change and any values above $1-change$ are set to be $1-change$.
`...`	not used.

Details

This is a function used to calculate model departure as a novel data representation.

Value

A matrix of departure as a novel data representation (matrix as input) or an S3 object for class 'scppp' (scppp object as input; departure result will be stored in object scppp under "representation").

Examples

```
# Matrix as input
test_set <- matrix(rpois(500, 0.5), nrow = 10)
adj_CDF_logit(test_set)
# scppp object as input
adj_CDF_logit(scppp(test_set))
```

`cluster_size` *Cluster size*

Description

This function calculates the number of elements in current cluster

Usage

```
cluster_size(test_dat)
```

Arguments

`test_dat`	a matrix or data frame with cells to cluster as rows

Value

a numeric value with number of cells to cluster

clust_clean *Cluster label clean*

Description

This function removes unwanted characters from cluster label string

Usage

```
clust_clean(clust)
```

Arguments

clust	a string indicates cluster label at each split step

Details

The clust_clean function removes any "-" or "NA" at the end of a string for a given cluster label

Value

a string with unwanted characters removed

diff_gene_list *Differential expression analysis*

Description

This function returns a data frame with differential expression analysis results.

Usage

```
diff_gene_list(
  data,
  final_clust_res = NULL,
  clust1 = "1",
  clust2 = "2",
  t_test = FALSE,
  ...
)
```

Arguments

data	A departure matrix generated from adj_CDF_logit() or an S3 object for class 'scppp'.
final_clust_res	A data frame with clustering results generated from HclustDepart(). It contains two columns: names (cell names) and clusters (cluster label).
clust1	One of the cluster label used to make comparison, default "1".
clust2	The other cluster label used to make comparison, default "2".
t_test	A logical value indicating whether the t-test should be used to make comparison. In general, for large cluster ($n \geq 30$), the t-test should be used. Otherwise, the Wilcoxon test might be more appropriate.
...	not used.

Details

This is a function used to find deferentially expressed genes between two clusters.

Value

A data frame contains genes (ranked by decreasing order of mean difference), and associated statistics (p-values, FDR adjusted p-values, etc.). If the input is an S3 object for class 'scppp', differential expression analysis results will be stored in object scppp under "de_results".

---

fwer_cutoff-shc *get FWER cutoffs for shc object*

Description

get FWER cutoffs for shc object

Usage

```
## S3 method for class 'shc'
fwer_cutoff(obj, alpha, ...)
```

Arguments

obj	shc object
alpha	numeric value specifying level
...	other parameters to be used by the function

Author(s)

Patrick Kimes

get_example_data *get example data*

Description

get example data

Usage

```
get_example_data(x = c("p5", "p56"))
```

Arguments

x	data set to choose

Value

A data set from example data

HclustDepart *Cluster cells in a recursive way*

Description

This function returns a list with clustering results.

Usage

```
HclustDepart(data, maxSplit = 10, minSize = 10, sim = 100, ...)
```

Arguments

data	A UMI count matrix with genes as rows and cells as columns or an S3 object for class 'scppp'.
maxSplit	A numeric value specifying the maximum allowable number of splitting steps (default 10).
minSize	A numeric value specifying the minimal allowable cluster size (the number of cells for the smallest cluster, default 10).
sim	A numeric value specifying the number of simulations during the Monte Carlo simulation procedure for statistical significance test, i.e. n_sim argument when apply sigclust2 (default = 100).
...	not used.

Details

This is a function used to get cell clustering results in a recursive way. At each step, the two-way approximation is re-calculated again within each subcluster, and the potential for further splitting is calculated using sigclust2. A non significant result suggests cells are reasonably homogeneous and may come from the same cell type. In addition, to avoid over splitting, the maximum allowable number of splitting steps `maxSplit` (default is 10, which leads to at most $2^{10} = 1024$ total number of clusters) and minimal allowable cluster size `minSize` (the number of cells in a cluster allowed for further splitting, default is 10) may be set beforehand. Thus the process is stopped when any of the conditions is satisfied: (1) the split is no longer statistically significant; (2) the maximum allowable number of splitting steps is reached; (3) any current cluster has less than 10 cells.

Value

A list with the following elements:

- `res2`: a data frame contains two columns: names (cell names) and clusters (cluster label)
- `sigclust_p`: a matrix with cells to cluster as rows, split index as columns, the entry in row `i` and column `j` denoting the p-value for the cell `i` at split step `j`
- `sigclust_z`: a matrix with cells to cluster as rows, split index as columns, the entry in row `i` and column `j` denoting the z-score for the cell `i` at split step `j`

If the input is an S3 object for class 'scppp', clustering result will be stored in object scppp under "clust_results".

Examples

```
test_set <- matrix(rpois(500, 0.5), nrow = 10)
HclustDepart(test_set)
```

`interpolate` *Linear interpolation for one sample given reference sample*

Description

This function returns a data frame with interpolated data points.

Usage

```
interpolate(df, reference, sample_id)
```

Arguments

`df`	The object data frame requires interpolation.
`reference`	The reference data frame to make comparison.
`sample_id`	A character to denote the object data frame.

Details

This is a function developed to do linear interpolation for corresponding probability from empirical cumulative distribution function (CDF) and corresponding quantiles. Given a reference data frame and a data frame needed to do interpolation, if there are any CDF values in reference but not in object data frame, do the linear interpolation and insert both CDF values and respective quantiles to the original object data frame.

Value

A data frame contains CDF, the sample name, and the corresponding quantiles.

`logit` *Logit transformation*

Description

This function applies logit transformation for a given probability

Usage

```
logit(p)
```

Arguments

`p`	a numeric value of probability, ranges between 0 and 1, exactly 0 and 1 not allowed

Details

The logit function transforms a probability within the range of 0 and 1 to the real line

Value

a numeric value transformed to the real line

`LouvainDepart` *Louvain clustering using departure as data representation*

Description

This function returns a list with elements useful to check and compare cell clustering.

Usage

```
LouvainDepart(
  data,
  pdat = NULL,
  PCA = TRUE,
  N = 15,
  pres = 0.8,
  tsne = FALSE,
  umap = FALSE,
  ...
)
```

Arguments

`data`	A UMI count matrix with genes as rows and cells as columns or an S3 object for class 'scppp'.
`pdat`	A matrix used as input for cell clustering. If not specify, the departure matrix will be calculated within the function.
`PCA`	A logic value specifying whether apply PCA before Louvain clustering, default is `TRUE`.
`N`	A numeric value specifying the number of principal components included for further clustering (default 15).
`pres`	A numeric value specifying the resolution parameter in Louvain clustering (default 0.8)
`tsne`	A logic value specifying whether t-SNE dimension reduction should be applied for visualization.
`umap`	A logic value specifying whether UMAP dimension reduction should be applied for visualization.
`...`	not used.

Details

This is a function used to get cell clustering using Louvain clustering algorithm implemented in the Seurat package.

Value

A list with the following elements:

- `sdata`: a Seurat object
- `tsne_data`: a matrix containing t-SNE dimension reduction results, with cells as rows, and first two t-SNE dimensions as columns; NULL if `tsne = FALSE`.
- `umap_data`: a matrix containing UMAP dimension reduction results, with cells as rows, and first two UMAP dimensions as columns; NULL if `tsne = FALSE`.
- `res_clust`: a data frame contains two columns: names (cell names) and clusters (cluster label)

References

Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck III WM, Hao Y, Stoeckius M, Smibert P, Satija R (2019). "Comprehensive Integration of Single-Cell Data." *Cell*, **177**, 1888-1902. doi:10.1016/j.cell.2019.05.031.

Examples

```
set.seed(1234)
test_set <- matrix(rpois(500, 2), nrow = 20)
rownames(test_set) <- paste0("gene", 1:nrow(test_set))
colnames(test_set) <- paste0("cell", 1:ncol(test_set))
LouvainDepart(test_set)
```

`nboot_small` *Random sample generation function to generate sets of samples from theoretical Poisson distribution.*

Description

This function returns a data frame with generated sets of samples and simulation index.

Usage

```
nboot_small(x, lambda, R)
```

Arguments

`x`	a numeric vector of sampled data points to compare with theoretical Poisson.
`lambda`	a numeric value for mean of theoretical Poisson.
`R`	a numeric value for mean of theoretical Poisson.

Details

This is a function used to simulate a given number sets of samples from a theoretical Poisson distribution that match input samples on sample size and sample mean (or theoretical Poisson parameter). Plotting these as envelopes in Q-Q plot shows the variability in shapes we can expect when sampling from the theoretical Poisson distribution.

Value

A data frame contains simulated data and corresponding simulation index. Random sample generation function to generate sets of samples from theoretical Poisson distribution.

nboot_small returns a data frame with generate sets of samples and simulation index.

This is a function used to simulate a given number sets of samples from a theoretical Poisson distribution that match input samples on sample size and sample mean (or theoretical Poisson parameter). Plotting these as envelopes in Q-Q plot shows the variability in shapes we can expect when sampling from the theoretical Poisson distribution.

a numeric vector of number of simulation sets that match input samples on sample size and sample mean (or theoretical Poisson parameter).

---

new_quantile *A more "continuous" approximation of quantiles of samples with a few integer case*

Description

This function returns a data frame including data points and corresponding quantile.

Usage

```
new_quantile(data, sample)
```

Arguments

data	A numeric vector of sampled data points.
sample	A character string denotes which sample data points come from.

Details

This is a function developed to get quantile for samples with only a few integer values. Define both $p_{-1} = 0$ and $q_{-1} = 0$. Replace the point mass at each integer z by a bar on the interval $[z \breve{} \frac{1}{2}, z + \frac{1}{2}]$ with height $P(X = z)$. This is a more "continuous" approximation of quantiles in this case.

Value

A data frame contains the corresponding probability from cumulative distribution function (CDF), sample name, and corresponding respective quantiles.

new_quantile_pois — *A more "continuous" approximation of quantiles from the theoretical Poisson distribution.*

Description

This function returns a data frame including the probability from cumulative distribution function (CDF) and corresponding quantiles.

Usage

```
new_quantile_pois(data, lambda)
```

Arguments

data	A numeric vector of sampled data points to compare with theoretical Poisson.
lambda	A numeric value for theoretical Poisson distribution parameter (equal to mean).

Details

This is a function developed to get corresponding quantiles from theoretical Poisson distribution. The data points ranges from 0 to maximum value of sampled data used to compare with the theoretical Poisson distribution.

Value

A data frame contains CDF probability and corresponding quantiles from the theoretical Poisson distribution.

para_est_new — *Parameter estimates based on two-way approximation*

Description

This function returns a vector consists of parameter estimates for overall offset, cell effect, and gene effect.

Usage

```
para_est_new(test_set)
```

Arguments

test_set	A UMI count data matrix with genes as rows and cells as columns

Details

This is a function used to calculate parameter estimates based on $\lambda_{gc} = e^{\mu+\alpha_g+\beta_c}$, where μ is the overall offset, α is a vector with the same length as the number of genes, and β is a vector with the same length as the number of cells. The order of elements in vectors α or β is the same as rows (genes) or cells (columns) from input data. Be sure to remove cells/genes with all zeros.

Value

A numeric vector containing parameter estimates from overall offset (first element), gene effect (same order as rows) and cell effect (same order as columns).

Examples

```
# Matrix as input
test_set <- matrix(rpois(500, 0.5), nrow = 10)
para_est_new(test_set)
```

---

qqplot_env_pois *Q-Q plot comparing samples with a theoretical Poisson distribution*

Description

This function returns a Q-Q plot with envelope using a more "continuous" approximation of quantiles.

Usage

```
qqplot_env_pois(sample_data, lambda, envelope_size = 100, ...)
```

Arguments

sample_data	A numeric vector of sample data points or an S3 object for class 'scppp'.
lambda	A numeric value specifying the theoretical Poisson parameter.
envelope_size	A numeric value specifying the size of envelope on Q-Q plot (default 100).
...	not used.

Details

This is a function for Q-Q envelope plot used to compare whether given sample data points come from the theoretical Poisson distribution. By simulating repeated samples of the same size from the candidate theoretical distribution, and overlaying the envelope on the same figure, it provides a feeling of understanding the natural variation from the theoretical distribution.

If an S3 object for class 'scppp' is used as input and the stored result under "data" is a matrix, The GLM-PCA algorithm will be applied to estimate the Poisson parameter for each matrix entry. Then a specific number of entries will be selected as sample data points to compare with the theoretical Poisson distribution.

Value

A ggplot object.

References

Townes FW, Street K (2020). *glmpca: Dimension Reduction of Non-Normally Distributed Data*. R package version 0.2.0, https://CRAN.R-project.org/package=glmpca.

`qqplot_small_test` *Q-Q plot comparing two samples with small discrete counts*

Description

This function returns a ggplot object used to visualize quantiles comparing distributions of two samples.

Usage

```
qqplot_small_test(P, Q, sample1, sample2)
```

Arguments

`P`	A numeric vector from one sample.
`Q`	A numeric vector from the other sample.
`sample1`	A character to denote sample name of one distribution `P` generated from.
`sample2`	A character to denote sample name of the other distribution `Q` generated from.

Details

This is a function for quantile-quantile plot comparing comparing samples from two discrete distributions after *continuity correction* and linear interpolation

Value

A ggplot object. Q-Q plot with continuity correction. Quantiles from one sample on the horizontal axis and corresponding quantiles from the other sample on the vertical axis.

qq_interpolation *Paired quantile after interpolation between two samples*

Description

This function returns a data frame with paired quantiles in two samples after interpolation.

Usage

```
qq_interpolation(dfp, dfq, sample1, sample2)
```

Arguments

dfp	A data frame generated from function new_quantile() based on a specific distribution.
dfq	Another data frame generated from function new_quantile() based on a specific distribution.
sample1	A character to denote sample name of distribution used to generate dfp.
sample2	A character to denote sample name of distribution used to generate dfq.

Details

This is a function for quantile interpolation of two samples. For each unique quantile value that has original data point in one sample but no corresponding original data point in another sample, apply a linear interpolation. So the common quantile values after interpolation should have unique points the same as unique quantile points from either sample.

Value

A data frame contains corresponding probability from cumulative distribution function (CDF), corresponding quantiles from the first sample (dfp), and corresponding quantiles from the second sample (dfq).

scppp *Generate New scppp object*

Description

Define S3 class that stores scRNA-seq data and associated information (e.g. model departure representation, cell clustering results) if corresponding functions are called.

Usage

```
scppp(data, sample = c("columns", "rows"))
```

Arguments

data	input data - Usually a matrix of counts
sample	by rows or columns

Value

S3 object for class 'scppp'.

---

sigp *Significance for first split using sigclust2*

Description

This function returns a list with elements mainly generated from sigclust2.

Usage

```
sigp(test_dat, minSize = 10, sim = 100)
```

Arguments

test_dat	A UMI count data matrix with samples to cluster as rows and features as columns.
minSize	A numeric value specifying the minimal allowable cluster size (the number of cells for the smallest cluster, default 10).
sim	A numeric value specifying the number of simulations during the Monte Carlo simulation procedure (default 100).

Details

This is a function used to calculate the significance level of the first split from hierarchical clustering based on euclidean distance and Ward's linkage.

Value

A list with the following elements:

- p: p-value for the first split
- z: z-score for the first split
- shc_result: a shc S3-object as defined in sigclust2 package
- clust2: a vector with group index for each cell
- clust_dat: a matrix of data representation used as input for hierarchical clustering

References

Kimes PK, Liu Y, Neil Hayes D, Marron JS (2017). "Statistical significance for hierarchical clustering." *Biometrics*, **73**(3), 811–821. Michael Linderman (2019). *Rclusterpp: Linkable C++ Clustering*. R package version 0.2.5, https://github.com/nolanlab/Rclusterpp.

theme_dirk *Dirk theme ggplots*

Description

This function generates ggplot object with theme elements that Dirk appreciates on his ggplots

Usage

```
theme_dirk(
  base_size = 22,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22,
  time_stamp = FALSE
)
```

Arguments

base_size	base font size, given in pts.
base_family	base font family
base_line_size	base size for line elements
base_rect_size	base size for rect elements
time_stamp	Logical value to indicate if the current time should be added as a caption to the plot. Helpful for versioning of plots.

Value

list that can be added to a ggplot object

Index