Package ‘tricycle’

May 2, 2024

Type Package

Title tricycle: Transferable Representation and Inference of cell cycle

Version 1.12.0

Description The package contains functions to infer and visualize cell cycle process using Single Cell RNASeq data. It exploits the idea of transfer learning, projecting new data to the previous learned biologically interpretable space. We provide a pre-learned cell cycle space, which could be used to infer cell cycle time of human and mouse single cell samples. In addition, we also offer functions to visualize cell cycle time on different embeddings and functions to build new reference.

Depends R (>= 4.0), SingleCellExperiment

Imports methods, circular, ggplot2, ggnewscale, AnnotationDbi, scater, GenomicRanges, IRanges, S4Vectors, scattermore, dplyr, RColorBrewer, grDevices, stats, SummarizedExperiment, utils

Suggests testthat (>= 3.0.0), BiocStyle, knitr, rmarkdown, CircStats, cowplot, htmltools, Seurat, org.Hs.eg.db, org.Mm.eg.db

License GPL-3

VignetteBuilder knitr

Encoding UTF-8

LazyData FALSE

RoxygenNote 7.1.2

biocViews SingleCell, Software, Transcriptomics, RNASeq, Transcription, BiologicalQuestion, DimensionReduction, ImmunoOncology

URL https://github.com/hansenlab/tricycle

BugReports https://github.com/hansenlab/tricycle/issues

git_url https://git.bioconductor.org/packages/tricycle

git_branch RELEASE_3_19

git_last_commit b8b8f3b
circle_scale_legend

Get the cyclic legend

Description

This function is a helper function to create the cyclic ggplot color legend.

Usage

```r
circle_scale_legend(hue.colors = c("#2E22EA", "#9E3DFB", "#F86BE2", "#FCCE7B", "#C4E416", "#4BBA0F", "#447D87", "#2C24E9"),
hue.n = 500, alpha = 0.6, y.inner = 1.5, y.outer = 3, y.text = 3.8, ymax = 4.5, text.size = 3, addStageLabel = FALSE,
G1.pos = 0, S.pos = 2.2, G2M.pos = 3.9)
```

Arguments

- **hue.colors**: The string vector gives the cyclic colors. The first color should look very similar to the last one. Default: c("#2E22EA", "#9E3DFB", "#F86BE2", "#FCCE7B", "#C4E416", "#4BBA0F", "#447D87", "#2C24E9")
- **hue.n**: The number of breaks of color scheme. Default: 500
- **alpha**: The alpha value (transparency). Default: 0.6
- **y.inner**: The radius of inner circle of the donut. Default: 1.5
- **y.outer**: The radius of outer circle of the donut. Default: 3


**Diagnose Totalumi**

- `y.text`: The radius of text position. Default: 3.8
- `y.max`: The value control the border of the legend. Default: 4.5
- `text.size`: The size of the text. Default: 3
- `addStageLabel`: Whether to add approximate discrete stage labels. Default: FALSE
- `G1.pos`: Approximate radius value of G1 label position. Default: 0
- `S.pos`: Approximate radius value of S label position. Default: 2.2
- `G2M.pos`: Approximate radius value of G2M label position. Default: 3.9

**Details**

The function will make a donut shape to serve as the cyclic color legend. The arguments should match the argument used in `plot_emb_circle_scale`.

**Value**

A ggplot object

**Author(s)**

Shijie C. Zheng

**Examples**

```r
circle_scale_legend()
```

---

**Diagnose Totalumi**

*Diagnostic function for UMI based datasets*

**Description**

The function will fit loess line for total UMIs numbers over cell cycle position to diagnose non-fitting data, of which cells are not cycling.

**Arguments**

- `theta.v`: The cell cycle position - a numeric vector with range between 0 to 2pi.
- `totalumis`: The total UMIs number for each cell (without log2 transformation) - a numeric vector with the same length as `theta.v`.
- `span`: The parameter $\alpha$ which controls the degree of smoothing. See `loess`. Default: 0.3
- `length.out`: The number of data points on the fitted lines to be output in the prediction data.frame. Default: 200
- `plot`: If TRUE, a ggplot scatter plot will be included in the output list. The figure will plot log2(totalumis) ~ theta.v with points and the fitted `loess` line. Default: TRUE
fig.title: The title of the figure. Default: NULL
point.size: The size of the point in scatter plot used by geom_scattermore. Default: 2.1
point.alpha: The alpha value (transparency) of the point in scatter plot used by geom_scattermore. Default: 0.6
line.size: The size of the fitted line, used by geom_path. Default: 0.8
line.alpha: The alpha value (transparency) of the fitted line, used by geom_path. Default: 0.8
x_lab: Title of x-axis. Default: "θ"
y_lab: Title of y-axis. Default: "log2(totalumis)"

Details

This function fit a loess line between cell cycle position and log2 transformed total UMI number, as described in fit_periodic_loess. If almost all cells are not cycling in a dataset, the estimated cell cycle positions might be incorrect due to the shifted embedding center. Using the fact that the cell should have highest total UMI number at the end of S phase and almost half of that highest total UMI number at M phase, we could detect those datasets which should be analysed and interpreted carefully when using tricycle package. For such problematic datasets, the default embedding center (0, 0) could lead to wrong inference. Thus, we don’t recommend using cell cycle position values if you get warnings from the diagnose_totalumi function.

Value

A diagnostic message and a list with the following elements:

- fitted: The fitted values on the loess line. A vector of the length of y.
- residual: The residual values from the fitted loess line, i.e. y - y.fit. A vector of the length of y.
- pred.df: The prediction data.frame by uniformly sampling theta from 0 - 2*pi. Names of variables: x and y. The number of rows equals to length.out.
- loess.o: The fitted loess object.
- rsquared: The coefficient of determination R2. Calculated as 1 - residual sum of squares / the total sum of squares.
- fig: When plot is TRUE, a ggplot scatter plot object will be returned with other items.

Author(s)

Shijie C. Zheng

See Also

fit_periodic_loess.
estimate_cycle_position

Assign cell cycle position

Description

Assign cell cycle position by the angle formed by PC1 and PC2 in the cell cycle space. If the cell cycle projection does not exist, the function will project the data.

Usage

estimate_cycle_position(
  x,
  exprs_values = "logcounts",
  dimred = "tricycleEmbedding",
  center.pc1 = 0,
  center.pc2 = 0,
  altexp = NULL,
  ...
)

Arguments

x A numeric matrix of **log-expression** values where rows are features and columns are cells. Alternatively, a SummarizedExperiment or SingleCellExperiment containing such a matrix.

exprs_values Integer scalar or string indicating which assay of x contains the **log-expression** values, which will be used for projection. If the projection already exists, you can ignore this value. Default: 'logcounts'

dimred The name of reducedDims in SingleCellExperiment (reducedDims). If the dimred already exists, it will be used to assign cell cycle position. If dimred does not exist, the projection will be calculated first by project_cycle_space and stored with name dimred in x. Default: 'tricycleEmbedding'

center.pc1 The center of PC1 when defining the angle. Default: 0

center.pc2 The center of PC2 when defining the angle. Default: 0

altexp String or integer scalar specifying an alternative experiment containing the **log-expression** data, which will be used for projection. If the projection is already calculated and stored in the SingleCellExperiment as a dimred, leave this value to default NULL.

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_cycle_position(neurosphere_example)
diagnose.l <- diagnose_totalumi(neurosphere_example$tricyclePosition, neurosphere_example$TotalUMIs, plot = TRUE)
Arguments to be used by `project_cycle_space`. If `x` is a `SingleCellExperiment`, and the projection is already in the `reducedDim` with name `dimred`. The `dimred` will be directly used to assign cell cycle position without new projection.

Details

The function will use assign the cell cycle position by the angle formed by the PC1 and PC2 of cell cycle projections. If the input is a numeric matrix or a `SummarizedExperiment`, the projection will be calculated with the input \( \log \text{-expression} \) values. For `SingleCellExperiment`, the projection will also be calculated if the designated `dimred` does not exist. Otherwise, the `dimred` will directly be used to assign cell cycle position. Therefore, this function is a wrapper function if the input is a `SingleCellExperiment`. Refer to `project_cycle_space` to all arguments during the projection.

The estimated cell cycle position is bound between 0 and 2\( \pi \). Note that we strive to get high resolution of cell cycle state, and we think the continuous position is more appropriate when describing the cell cycle. However, to help users understand the position variable, we also note that users can approximately relate 0.5\( \pi \) to be the start of S stage, \( \pi \) to be the start of G2M stage, 1.5\( \pi \) to be the middle of M stage, and 1.75\( \pi \)-0.25\( \pi \) to be G1/G0 stage.

Value

If the input is a numeric matrix, the cell cycle position - a numeric vector bound between 0 \( \sim \) 2\( \pi \) with the same length as the number of input column will be returned.

If the input is `SummarizedExperiment`, the original `SummarizedExperiment` with cell cycle position stored in `colData` with name `tricyclePosition` will be returned.

If the input is `SingleCellExperiment`, the original `SingleCellExperiment` with cell cycle position stored in `colData` with name `tricyclePosition` will be returned and the projection will be stored in `reducedDims(..., dimred)` if it does not exist before.

Author(s)

Shijie C. Zheng

References


See Also

`project_cycle_space`, for projecting new data with a pre-learned reference

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_cycle_position(neurosphere_example)
reducedDimNames(neurosphere_example)
plot(reducedDim(neurosphere_example, "tricycleEmbedding"))
plot(neurosphere_example$tricyclePosition, reducedDim(neurosphere_example, "tricycleEmbedding")[, 1])
estimate_Schwabe_stage

plot(neurosphere_example$tricyclePosition, 
     reducedDim(neurosphere_example, "tricycleEmbedding")[, 2])

estimate_Schwabe_stage

Assign cell cycle stages using Schwabe method

Description

The function is a re-implementation of cell cycle stage assignment method proposed in Schwabe et al.(2020), with a little modification. The core assignment method is not designed by the authors of this package!

Usage

estimate_Schwabe_stage( 
  x, 
  exprs_values = "logcounts", 
  batch.v = NULL, 
  altexp = NULL, 
  cycleGene.l = NULL, 
  gname = NULL, 
  gname.type = c("ENSEMBL", "SYMBOL"), 
  species = c("mouse", "human"), 
  AnnotationDb = NULL, 
  corThres = 0.2, 
  tolerance = 0.3 
)

Arguments

x A numeric matrix of **log-expression** values where rows are features and columns are cells. Alternatively, a SummarizedExperiment or SingleCellExperiment containing such a matrix.

exprs_values Integer scalar or string indicating which assay of x contains the **log-expression** values, which will be used for projection. If the projection already exists, you can ignore this value. Default: 'logcounts'

batch.v A string specifies which column in colData of SummarizedExperiment or SingleCellExperiment to use as the batch variable. Or it can be a vector, of which the number of elements equals to the number of columns of x. The 5 stage cell cycle assignments are preformed for each batch separately. No NA is permitted. Default: NULL

altexp String or integer scalar specifying an alternative experiment containing the **log-expression** data, which will be used for projection. If the projection is already calculated and stored in the SingleCellExperiment as a dimred, leave this value to default NULL.
cycleGene.l A list contains the marker genes for each stage. The stage names should be included as names of the elements. If user feed custom list, they should make sure that the same gene id type for x and cycleGene.l. If not custom list is given, RevelioGeneList will be used. Default: NULL.

gname Alternative rownames of x. If provided, this will be used to map genes within x with genes in ref.m. If not provided, the rownames of x will be used instead. Default: NULL.

gname.type The type of gene names as in gname or rownames of x. It can be either 'ENSEMBL' or 'SYMBOL'. If the user uses custom ref.m, this value will have no effect. Default: 'ENSEMBL'.

species The type of species in x. It can be either 'mouse' or 'human'. If the user uses custom cycleGene.l, this value will have no effect. Default: 'mouse'.

AnnotationDb An AnnotationDb objects. It is used to map ENSEMBL IDs to gene SYMBOLs. If no AnnotationDb object being given, the function will use org.Hs.eg.db or org.Mm.eg.db for human and mouse respectively.

corThres For each batch and each stage, correlations between expression of each gene and the mean of all genes belonging to that stage will be calculated to filter the final gene list used for inference. The genes with a correlation between corThres will not be used for calculating z-scores. Default: 0.2

tolerance For each cell, the function will compare the largest two z-scores. If the difference between those two z-scores is less than tolerance, the cell will be treated unassignable with NA value returned for that cell. Default: 0.3

Details

The function is a re-implementation of cell cycle stage assignment method proposed in Schwabe et al.(2020), with a little modification. We include this function only for the purpose of convenience. The core assignment method is not designed by the authors of this package! Briefly, the function assigns cells to discretized cell cycle stages by comparing the z-scores calculated for each stage markers. Without cycleGene.l input, RevelioGeneList will be used. If you use this function, you should cite Schwabe et al.(2020).

Value

If the input is a numeric matrix, the discretized cell cycle stages - a factor vector corresponding to each cell will be returned.

If the input is SummarizedExperiment, the original SummarizedExperiment with the discretized cell cycle stages stored in colData with name 'CCStage' will be returned.

If the input is SingleCellExperiment, the original SingleCellExperiment with the discretized cell cycle stages stored in colData with name 'CCStage' will be returned.

Author(s)

Shijie C. Zheng
References


Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_Schwabe_stage(neurosphere_example,
   gname.type = "ENSGBL", species = "mouse")
neurosphere_example2 <- estimate_Schwabe_stage(neurosphere_example, batch.v = "sample")
neurosphere_example3 <- estimate_Schwabe_stage(neurosphere_example,
   batch.v = neurosphere_example$sample)
neurosphere_example <- project_cycle_space(neurosphere_example)
plot(reducedDim(neurosphere_example, "tricycleEmbedding"),
   col = neurosphere_example$CCStage)

fit_periodic_loess  Fit periodic loess line with circular predictor

Description

The function will fit a loess line using cell cycle position and other variables, such as expression levels of a gene or log-transformed totalUMIs numbers. The circular nature of cell cycle position is taken into account by making 3 copies inside the function. For convenience, the function will also return a scatter plot with fitted line if needed.

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta.v</td>
<td>The cell cycle position - a numeric vector with range between 0 to 2pi.</td>
</tr>
<tr>
<td>y</td>
<td>The response variable - a numeric vector with the same length as theta.v.</td>
</tr>
<tr>
<td>span</td>
<td>The parameter α which controls the degree of smoothing. See loess. Default: 0.3</td>
</tr>
<tr>
<td>length.out</td>
<td>The number of data points on the fitted lines to be output in the prediction</td>
</tr>
<tr>
<td>plot</td>
<td>If TRUE, a ggplot scatter plot will be included in the output list. The figure will plot y ~ theta.v with points and the fitted loess line. Default: FALSE</td>
</tr>
<tr>
<td>fig.title</td>
<td>The title of the figure. Default: NULL</td>
</tr>
<tr>
<td>point.size</td>
<td>The size of the point in scatter plot used by geom_scattermore. Default: 2.1</td>
</tr>
<tr>
<td>point.alpha</td>
<td>The alpha value (transparency) of the point in scatter plot used by geom_scattermore. Default: 0.6</td>
</tr>
<tr>
<td>line.size</td>
<td>The size of the fitted line, used by geom_path. Default: 0.8</td>
</tr>
<tr>
<td>line.alpha</td>
<td>The alpha value (transparency) of the fitted line, used by geom_path. Default: 0.8</td>
</tr>
</tbody>
</table>
color.vars  Optional. A vector of categorical variable of the same length of theta.v, and it will be used to color points in figure. Default: NULL

color.name  The name of the color variables. Used as the same for legend. Default: NULL

x_lab  Title of x-axis. Default: "θ"

y_lab  Title of y-axis. Default: "y"

hue.colors  The string vector gives custom colors. If not given, the default `scale_color_discrete` will be used. Default: NULL

...  Other arguments input to `loess`.

Details

This function fit a normal loess line, but take the circularity of cell cycle position into account by making theta.v 3 periods (c(theta.v - 2 * pi, theta.v, theta.v + 2 * pi)) and repeating y 3 times. Only the fitted values corresponding to original theta.v will be returned. For convenience, the function will also return a scatter plot with fitted line if needed. Or user can use `pred.df` to visualize the loess line themselves.

Value

A list with the following elements:

- fitted - The fitted values on the loess line. A vector of the length of y.
- residual - The residual values from the fitted loess line, i.e. y - y.fit. A vector of the length of y.
- pred.df - The prediction data.frame by uniformly sampling theta from 0 - 2*pi. Names of variables: x and y. The number of rows equals to `length.out`.
- loess.o - The fitted loess object.
- rsquared - The coefficient of determination R2. Calculated as 1 - residual sum of squares / the total sum of squares.
- fig - When `plot` is TRUE, a `ggplot` scatter plot object will be returned with other items.

Author(s)

Shijie C. Zheng

See Also

`estimate_cycle_position`, for inferring cell cycle position.

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_cycle_position(neurosphere_example)
top2a.idx <- which(rowData(neurosphere_example)$Gene == "Top2a")
fit.l <- fit_periodic_loess(neurosphere_example$tricyclePosition, assay(neurosphere_example, "logcounts")[,top2a.idx,], plot = TRUE)
fit.l$fig
**neuroRef**

Pre-learned reference projection matrix from the Neurosphere dataset

---

**Description**

Default reference projection matrix learned from the Neurosphere dataset.

**Usage**

data(neuroRef)

**Format**

An object of class `"data.frame"`, with 5 variables. Normally, user won’t call this data directly as it will be automatically used in `project_cycle_space` if no custom reference projection matrix is provided. Each row is gene, and rotation scores for PC1 and PC2, mouse ENSEMBL IDs, and mouse gene SYMBOLs are included. The ‘SYMBOL’ values are just the upper-case ‘symbol’ values.

**References**


**Examples**

data(neuroRef)

---

**neurosphere_example**

Example SingleCellExperiment dataset

---

**Description**

This a subset of mouse Neurosphere data. 200 cells from sample AX1 and AX2 were randomly sampled from the full data. All genes in the GO cell cycle gene list and `RevelioGeneList` as well as other random 573 genes were included.

**Usage**

data(neurosphere_example)

**Format**

A `SingleCellExperiment` object of 1500 genes and 400 cells.

**Examples**

data(neurosphere_example)
plot_ccposition_den  
*Plot cell cycle position kernel density stratified by a factor*

**Description**

The function will compute and plot cell cycle position kernel density.

**Arguments**

- `theta.v`: The cell cycle position - a numeric vector with range between 0 to 2pi.
- `color_var.v`: A factor variable to stratify `theta.v`, such as samples or 'CCStage'. The length of it should equal to the length of `theta.v`.
- `color_name`: The name of the `color_var.v` to be used in the legend.
- `palette.v`: A string vector to give the color names. It should has the length of the number of levels of `color_var.v`. If not given, the 'Set1' palette will be used. (See `display.brewer.all` If the number of levels of `color_var.v` is greater than 8, only the top 8 levels of most cell will be shown. You can show them all by feeding enough colors in `palette.v`. Default: NULL
- `fig.title`: The title of the figure. Default: NULL
- `type`: It can be either of 'linear' or 'circular'. 'linear' corresponds to Cartesian coordinate system and 'circular' for polar coordinate system. Default: 'linear'
- `bw`: The smoothing bandwidth to be used. It is equal to the concentration parameter of the von Mises distribution. See `density.circular`. Default: 30
- `weighted`: Whether the density should be weighted on the percentage of each level of `color_var.v`. Default: FALSE
- `line.size`: The size of the line used by `geom_path`. Default: 0.5
- `line.alpha`: The alpha value of the line used by `geom_path`. Default: 0.5
- `addRug`: Whether to add rug on the bottom of the linear density plot or an inner circle on the circular plot to show the continuous scale of `theta`. Default: TRUE
- `RugPalette.v`: The palette used for the rug plot. If not given, it will used the same default palette as in `plot_emb_circle_scale`.
- `...`: Other arguments accepted by `geom_path`.

**Details**

The function first estimates kernel density using the von Mises distribution. Then, it plots out the density in the polar coordinate system or Cartesian coordinate system. Different colors represents different levels of `color_var.v` and the dashed black line is the marginal distribution of all cells.

**Value**

A ggplot object
Author(s)

Shijie C. Zheng

See Also

estimate_Schwabe_stage, for inferring 5 stages of cell cycle

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_cycle_position(neurosphere_example)
plot_ccposition_den(neurosphere_example$tricyclePosition, neurosphere_example$sample, "sample")

neurosphere_example <- estimate_Schwabe_stage(neurosphere_example,
gname.type = "ENSEMBL", species = "mouse")
plot_ccposition_den(neurosphere_example$tricyclePosition, neurosphere_example$CCStage, "CCStage")

plot_emb_circle_scale  Plot embedding with cyclic cell cycle position

Description

Generate scat plot of embedding with cyclic cell cycle position or other cyclic variables

Usage

plot_emb_circle_scale(
  sce.o,
  color_by = "tricyclePosition",
  facet_by = NULL,
  dimred = 1,
  dim = seq_len(2),
  fig.title = NULL,
  point.size = 2.1,
  point.alpha = 0.6,
  x_lab = NULL,
  y_lab = NULL,
  hue.colors = c("#2E22EA", "#9E3DFB", "#F86BE2", "#FCE7E2", "#C4E416", "#4BBA0F",
                 "#447D87", "#2C24E9"),
  hue.n = 500,
  plot.legend = FALSE
)
plot_emb_circle_scale

Arguments

sce.o A SingleCellExperiment contains the embedding to be plotted against.

color_by The name of variable in colData(sce.o) to be used to show colors. Default: "tricyclePosition"

facet_by The name of variable in colData(sce.o) to be used to facet scatter plots. If used, the function will return a list of ggplot objects. If NULL, no faceted panels will be returned. Default: NULL

dimred The name or index of reducedDims in SingleCellExperiment (reducedDims). Default: 1

dim The indices of dimred to be plotted. At the moment, it has to be two integers. Default: 1:2

fig.title The title of the figure. Default: NULL

point.size The size of the point in scatter plot used by geom_scattermore. Default: 2.1

point.alpha The alpha value (transparency) of the point in scatter plot used by geom_scattermore. Default: 0.6

x_lab Title of x-axis. If not given, the colname of dimred will be used. Default: NULL

y_lab Title of y-axis. If not given, the colname of dimred will be used. Default: NULL

hue.colors The string vector gives the cyclic colors. The first color should look very similar to the last one. Default: c("#2E22EA", "#9E3DFB", "#F86BE2", "#FCCE7B", "#C4E416", "#4BBA0F", "#447D87", "#2C24E9")

hue.n The number of breaks of color scheme. Default: 500

plot.legend Whether the legend should be plotted with the scatter plot. We recommend not to use this legend but use the cyclic legend produced by circle_scale_legend instead. Default: FALSE

Details

This function help users plot embedding scater plot colored by cyclic variables, such as cell cycle position, which is bound between 0 - 2pi. It will take a SingleCellExperiment object as input, and plot out its dimred such as PCA, UMAP, and etc with a cyclic color scheme.

Value

A ggplot object or a list of ggplot objects. If facet_by is not assigned, a single ggplot plot of the scatter plot will be return, Otherwise, apart from the first scatter plot showing all cells together, other faceted scatter plots will also be given in the list.

Author(s)

Shijie C. Zheng

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- estimate_cycle_position(neurosphere_example)
plot_emb_circle_scale(neurosphere_example, point.size = 3.1, point.alpha = 0.8)
Description

Project mouse and human single cell RNAseq data into a cell cycle embedding by a pre-learned reference projection matrix.

Usage

```r
project_cycle_space(
  x,
  exprs_values = "logcounts",
  altexp = NULL,
  name = "tricycleEmbedding",
  ref.m = NULL,
  gname = NULL,
  gname.type = c("ENSEMBL", "SYMBOL"),
  species = c("mouse", "human"),
  AnnotationDb = NULL)
```

Arguments

- **x**: A numeric matrix of **log-expression** values where rows are features and columns are cells. Alternatively, a `SummarizedExperiment` or `SingleCellExperiment` containing such a matrix.
- **exprs_values**: Integer scalar or string indicating which assay of `x` contains the **log-expression** values. Default: 'logcounts'
- **altexp**: String or integer scalar specifying an alternative experiment containing the input data.
- **name**: String specifying the name to be used to store the result in the `reducedDims` of the output. Default: 'tricycleEmbedding'
- **ref.m**: A custom reference projection matrix to project the new data, where rows are features and columns are dimensions. Users need to use the same type of `gname` (or rownames of `x`) as for the `ref.m`. If no custom `ref.m` is given, the internal reference `neuroRef` will be used.
- **gname**: Alternative rownames of `x`. If provided, this will be used to map genes within `x` with genes in `ref.m`. If not provided, the rownames of `x` will be used instead. Default: NULL
- **gname.type**: The type of gene names as in `gname` or rownames of `x`. It can be either 'ENSEMBL' or 'SYMBOL'. If the user uses custom `ref.m`, this value will have no effect. Default: 'ENSEMBL'
- **species**: The type of species in `x`. It can be either 'mouse' or 'human'. If the user uses custom `ref.m`, this value will have no effect. Default: 'mouse'
AnnotationDb An AnnotationDb objects. If the user uses the internal reference to project human data, and provide rownames in the format of Ensembl IDs, this object will be used to map Ensembl IDs to gene SYMBOLS. If no AnnotationDb object being given, the function will use org.Hs.eg.db.

Details

The function will use pre-learned cell cycle pattern to project new data to show the cell cycle progression. If the user uses internal Neuropshere reference, the expression values must be **log-transformed**. Besides, we would assume the input data has been already preprocessed, library size normalized at least. The projection process is to take sum of weighted mean-centered expression of chosen genes, so the mean expression of a given gene could be affected without library size normalization.

Value

If the input is a numeric matrix or a SummarizedExperiment, a projection matrix with rows cells and column dimensions will be returned. The actual rotation matrix used to project the data is included in the attributes with name 'rotation'.

For SingleCellExperiment, an updated SingleCellExperiment is returned containing projection matrix in reducedDims(..., name).

Author(s)

Shijie C. Zheng

References


See Also

estimate_cycle_position, for inferring cell cycle position.

Examples

data(neurosphere_example, package = "tricycle")
neurosphere_example <- project_cycle_space(neurosphere_example)
reducedDimNames(neurosphere_example)
head(reducedDim(neurosphere_example, "tricycleEmbedding"))
plot(reducedDim(neurosphere_example, "tricycleEmbedding"))
names(attributes(reducedDim(neurosphere_example, "tricycleEmbedding")))
RevelioGeneList

5 stage cell cycle gene marker list from Revelio

Description

This 5 stage cell cycle gene marker list is directly from Revelio package. Within the list, 5 vectors corresponds to highly expressed genes at the cell cycle stage. The genes are given as the human gene SYMBOLS. This gene list is originally from the Whitfield et al. (2020).

Usage

data(RevelioGeneList)

Format

An list of 5 string vector. The names of the elements are the names of cell cycle stages with the order of: G1S, S, G2, G2M, MG1.

References


Examples

data(RevelioGeneList)

run_pca_cc_genes

Run PCA on Gene Ontology cell cycle genes

Description

Run PCA on Gene Ontology cell cycle genes abd get a new SingleCellExperiment. User could use this function to learn new reference projection matrix.

Arguments

sce.o  A SingleCellExperiment contains library size normalized **log-expression** matrix.
gname  Alternative rownames of sce.o. If provided, this will be used to map genes within Gene Ontology cell cycle gene list. If not provided, the rownames of sce.o will be used instead. Default: NULL
exprs_values  Integer scalar or string indicating which assay of sce.o contains the **log-expression** values, which will be used to run PCA. Default: ‘logcounts’
### run_pca_cc_genes

#### Description

The function `run_pca_cc_genes` is used to calculate principal component analysis (PCA) on GO cell cycle genes. It requires an output of a `SingleCellExperiment` object which contains the library size normalized log-expression matrix. The full dataset will be subsetted to genes in the Gene Ontology cell cycle gene list (GO:0007049). The corresponding AnnotationDb object will be `org.Mm.eg.db` and `org.Hs.eg.db` for mouse and human respectively. If `runSeuratBy` is set, the data will be integrated to remove batch effect between samples/batches by Seurat.

#### Parameters

- **gname.type** (The type of gene names as in `gname` or rownames of `sce.o`): It can be either 'ENSEMBL' or 'SYMBOL'. Default: 'ENSEMBL'
- **species** (The type of species in `sce.o`): It can be either 'mouse' or 'human'. If the user uses custom `cycleGene.l`, this value will have no effect. Default: 'mouse'
- **AnnotationDb** (An AnnotationDb objects): It is used to map ENSEMBL IDs to gene SYMBOLs. If no AnnotationDb object being given, the function will use `org.Hs.eg.db` or `org.Mm.eg.db` for human and mouse respectively.
- **ntop** (The number of genes with highest variance to use when calculating PCA): Default: 500
- **ncomponents** (The number of component components to obtain): Default: 20
- **name** (String specifying the name to be used to store the result in the `reducedDims` of the output): Default: 'PCA'

#### Details

The function requires an output of a `SingleCellExperiment` object which contains the library size normalized log-expression matrix. The full dataset will be subsetted to genes in the Gene Ontology cell cycle gene list (GO:0007049). The corresponding AnnotationDb object will be `org.Mm.eg.db` and `org.Hs.eg.db` for mouse and human respectively. If `runSeuratBy` is set, the data will be integrated to remove batch effect between samples/batches by Seurat.

User can use this function to make new reference projection matrix by getting the 'rotation' attribute in PCA results. Such as `attr(reducedDim(sce.o, 'PCA'), 'rotation'[, 1:2]`. See examples for more details.

#### Value

A subset `SingleCellExperiment` object with only GO cell cycle genes will be return. The PCA resulting will be save in `reducedDims` with chosen name `reducedDims(..., name)`. If Seurat integration is performed, another `reducedDims` with name 'matched.'+name will also be included in the `SingleCellExperiment`.

#### Author(s)

Shijie C. Zheng

#### Examples

```r
data(neurosphere_example, package = "tricycle")
### Use internal NeuroRef to project and infer tricyclePosition
neurosphere_example <- estimate_cycle_position(neurosphere_example)

### Build new reference
gocc_sce.o <- run_pca_cc_genes(neurosphere_example)
new.ref <- attr(reducedDim(gocc_sce.o, "PCA"), "rotation"[, seq_len(2)])

### Use new reference to project and infer tricyclePosition
new_sce <- estimate_cycle_position(neurosphere_example, ref.m = new.ref,
```
run_pca_cc_genes

dimred = "tricycleEmbedding2")
plot(neurosphere_example$tricyclePosition, new_sce$tricyclePosition)
Index

* datasets
  - neuroRef, 11
  - neurosphere_example, 11
  - RevelioGeneList, 17

  calculatePCA, 18
  circle_scale_legend, 2, 14

  density.circular, 12
diagnose_totalumi, 3
display.brewer.all, 12

  estimate_cycle_position, 5, 10, 16
  estimate_Schwabe_stage, 7, 13

  fit_periodic_loess, 4, 9

  geom_path, 4, 9, 12
  geom_scattermore, 4, 9, 14

  loess, 3, 4, 9, 10

  neuroRef, 11
  neurosphere_example, 11

  org.Hs.eg.db, 8, 16, 18
  org.Mm.eg.db, 8, 18

  plot_ccposition_den, 12
  plot_emb_circle_scale, 3, 12, 13
  project_cycle_space, 5, 6, 11, 15

  reducedDims, 5, 6, 14–16, 18
  RevelioGeneList, 8, 11, 17
  run_pca_cc_genes, 17

  scale_color_discrete, 10
  SingleCellExperiment, 5–8, 11, 14–18
  SummarizedExperiment, 5–8, 15, 16