

Package: pGRN (via r-universe)

August 22, 2024

Type Package

Title Single-Cell RNA Sequencing Pseudo-Time Based Gene Regulatory Network Inference

Version 0.3.5

Author Gangcai Xie

Maintainer Gangcai Xie <gcxiester@gmail.com>

Description Inference and visualize gene regulatory network based on single-cell RNA sequencing pseudo-time information.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

VignetteBuilder knitr, rmarkdown

Depends R (>= 2.10),

Imports bigmemory, doParallel, dtw, foreach, ggplot2, ggraph, igraph, lmtest, proxy, tidygraph, visNetwork, future

Suggests knitr, webshot, rmarkdown,

NeedsCompilation no

Date/Publication 2023-01-17 17:20:02 UTC

Repository <https://gangcai.r-universe.dev>

RemoteUrl <https://github.com/cran/pGRN>

RemoteRef HEAD

RemoteSha f82aaedb035eaa9a3fcdd4aa84afddffb247902c

Contents

data_transform	2
get_dtw_dist_bidirectional	3
get_dtw_dist_mat	3

get_networks	4
matrix2adj	5
module_networks	6
pGRN	7
pGRNDB	8
plot_network	9
plot_network_i	9
run_dtw	10
run_granger_test	11
slideWindows	12

Index	13
--------------	-----------

data_transform	<i>Pseudotime based Expression Data Transformation</i>
----------------	--

Description

Based on single-cell pseudotime information, get the sliding window average expression, and then standard normalize the expression for each gene

Usage

```
data_transform(data, pseudotime, slide_window_size = 100, slide_step_size = 50)
```

Arguments

data	expression matrix data
pseudotime	list of pseudotime
slide_window_size	sliding window size
slide_step_size	sliding window step size

Value

Transformed new matrix

Examples

```
data <- matrix(1,100,1000)
ptime <- seq(1:1000)
data_transform(data,
               ptime,
               slide_window_size=100,
               slide_step_size=50)
```

`get_dtw_dist_bidirectional`
Bidirectional DTW Distance

Description

Get bidirectional DTW distance.

Usage

```
get_dtw_dist_bidirectional(x, y)
```

Arguments

x	list of x input
y	list of y input

Value

numeric

Examples

```
get_dtw_dist_bidirectional(c(1:1000),c(1:1000))
```

`get_dtw_dist_mat` *DTW distance matrix for all genes*

Description

Get DTW distance matrix for all genes using pseudotime based sliding window transformation, parallel computing allowed.

Usage

```
get_dtw_dist_mat(  
  data,  
  ptime,  
  slide_window_size = 50,  
  slide_step_size = 25,  
  cores = 2  
)
```

Arguments

data gene expression matrix (Gene * Cells)
 ptime pseudotime matched with the column cells of the gene expression matrix
 slide_window_size sliding window size
 slide_step_size sliding window step size
 cores number of cores for parallel computing

Value

bidirectional DTW distance matrix

Examples

```

example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
  
```

get_networks	<i>Get the list of sub-networks</i>
--------------	-------------------------------------

Description

Get sub-networks based on given adjacency data.frame input

Usage

```

get_networks(
  data,
  centrality_degree_mod = "out",
  components_mod = "weak",
  network_min_genes = 10
)
  
```

Arguments

data adjacency data.frame
 centrality_degree_mod mode of centrality degree for popularity calculation
 components_mod mode of sub-network extraction methods
 network_min_genes minimal number of gene elements required for extracted sub-networks

Value

list of `tbl_graph` objects

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
adj_df <- matrix2adj(dtw_dist_matrix)
get_networks(adj_df, network_min_genes=5)
```

`matrix2adj`*Convert distance matrix to adjacency dataframe*

Description

Convert distance matrix to adjacency dataframe for network construction.

Usage

```
matrix2adj(data, quantile_cutoff = 5)
```

Arguments

`data` distance matrix
`quantile_cutoff` an integer value (1-99) for quantile cutoff

Value

adjacency dataframe (with columns "from, to, distance,direction, similarity")

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
adj_df <- matrix2adj(dtw_dist_matrix)
```

module_networks	<i>Get module level networks</i>
-----------------	----------------------------------

Description

Given a distance matrix, calculate gene modules based on hierarchical clustering method and then get module level networks

Usage

```
module_networks(
  data,
  k = 10,
  quantile_cutoff = 10,
  centrality_degree_mod = "out",
  components_mod = "weak",
  network_min_genes = 10
)
```

Arguments

data	distance matrix
k	number of gene clusters for module inference
quantile_cutoff	distance cutoff based on quantile(1-99) for edge identification
centrality_degree_mod	"in" or "out" for nodes popularity calculation
components_mod	"weak" or "strong" for sub-network components inference
network_min_genes	minial number of genes required for a network

Value

a list networks for each module

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
nets <- module_networks(dtw_dist_matrix,k=1,quantile_cutoff=50)
plot_network(nets[["module1"]])
```

pGRN	<i>pGRN: creates gene regulatory network based on single cell pseudo-time information</i>
------	---

Description

Given single cell matrix and pseudotime, construct gene regulatory network (GRN)

Usage

```
pGRN(
  expression_matrix,
  pseudotime_list,
  method = "DTW",
  slide_window_size = 20,
  slide_step_size = 10,
  centrality_degree_mod = "out",
  components_mod = "weak",
  network_min_genes = 10,
  quantile_cutoff = 5,
  order = 1,
  cores = 1
)
```

Arguments

expression_matrix	expression matrix data
pseudotime_list	list of pseudotime
method	method for GRN construction: DTW, granger
slide_window_size	sliding window size
slide_step_size	sliding window step size
centrality_degree_mod	(for DTW method) mode of centrality degree for popularity calculation
components_mod	(for DTW method) mode of sub-network extraction methods (weak or strong)
network_min_genes	minimal number of gene elements required for extracted sub-networks
quantile_cutoff	an integer value (1-99) for quantile cutoff
order	(for granger method) integer specifying the order of lags to include in the auxiliary regression
cores	number of cores for parallel computing

Value

a list of `tabl_graph` objects

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime

# try DTW method
nets <- pGRN(expression_matrix,
              pseudotime_list,
              method= "DTW",
              quantile_cutoff=50,
              cores=1)
plot_network(nets[[1]])

# plot the network interactively
plot_network_i(nets[[1]])
```

pGRNDB

pGRN example data

Description

A list with expression dataframe and pseudotime dataframe

Usage

```
pGRNDB
```

Format

pGRNDB:

A list with items `expression` and `ptime`

expression data frame of single cell expression

ptime pseudotime of the single cells ...

Source

pGRN

plot_network	<i>Plot stationary network</i>
--------------	--------------------------------

Description

Plot stationary network through ggraph

Usage

```
plot_network(graph, ...)
```

Arguments

graph	a tbl_graph object
...	other parameters for ggraph

Value

ggraph

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
nets <- module_networks(dtw_dist_matrix, k=1, quantile_cutoff=50)
plot_network(nets[["module1"]])
```

plot_network_i	<i>Plot interactive network</i>
----------------	---------------------------------

Description

Plot interactive network based on igraph layout input

Usage

```
plot_network_i(graph, save_file = NULL)
```

Arguments

graph	igraph layout object
save_file	file name of the saved file, not save if NULL

Value

visNetwork htmlwidget

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
dtw_dist_matrix <- get_dtw_dist_mat(expression_matrix,
                                   pseudotime_list,
                                   cores=1)
nets <- module_networks(dtw_dist_matrix,k=1,quantile_cutoff=50)
plot_network_i(nets[["module1"]])
```

run_dtw

Get network adjacency dataframe based on DTW method

Description

Use DTW to calculate gene-gene distance based on their expression and pseudotime

Usage

```
run_dtw(
  expression_matrix,
  pseudotime_list,
  slide_window_size = 50,
  slide_step_size = 25,
  quantile_cutoff = 5,
  cores = 1
)
```

Arguments

expression_matrix
expression matrix data

pseudotime_list
list of pseudotime

slide_window_size
sliding window size

slide_step_size
sliding window step size

quantile_cutoff
an integer value (1-99) for quantile cutoff

cores
number of cores for parallel computing

Value

adjacency dataframe (with columns "from, to, distance,direction, similarity")

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
adj_df <- run_dtw(expression_matrix,
                  pseudotime_list,
                  quantile_cutoff=50,
                  cores=1)
```

run_granger_test	<i>Use Granger-causality Test to get gene-gene regulatory relationship</i>
------------------	--

Description

Based on single-cell gene expression matrix and pseudotime, calculate Granger-causality Test based gene-gene regulatory relationship

Usage

```
run_granger_test(
  data,
  ptime,
  slide_window_size = 20,
  slide_step_size = 10,
  pvalue_cutoff = 0.01,
  order = 1,
  ...
)
```

Arguments

data	gene expression matrix (Gene * Cells)
ptime	pseudotime matched with the column cells of the gene expression matrix
slide_window_size	sliding window size
slide_step_size	sliding window step size
pvalue_cutoff	cutoff for the pvalue from transfer entropy test
order	integer specifying the order of lags to include in the auxiliary regression
...	other parameters for grangertest function in lmtree

Value

adjacency data frame

Examples

```
example_data <- pGRNDB
expression_matrix <- example_data[["expression"]]
pseudotime_list <- example_data[["ptime"]]$PseudoTime
gt_adj_df <- run_granger_test(expression_matrix, pseudotime_list)
```

slideWindows

Sliding Window Average

Description

Get sliding windows average values for given vector/list

Usage

```
slideWindows(data, window = 2, step = 1)
```

Arguments

data	list of expression
window	sliding window size
step	sliding window step size

Value

list/vector of sliding windows with average expression value

Examples

```
slideWindows(c(1:1000),window=200,step=100)
slideWindows(c(1:1000),window=100,step=50)
```

Index

* datasets

pGRNDB, 8

data_transform, 2

get_dtw_dist_bidirectional, 3

get_dtw_dist_mat, 3

get_networks, 4

matrix2adj, 5

module_networks, 6

pGRN, 7

pGRNDB, 8

plot_network, 9

plot_network_i, 9

run_dtw, 10

run_granger_test, 11

slideWindows, 12