

Package ‘scPOEM’

May 18, 2025

Type Package

Title Single-Cell Meta-Path Based Omics EMBedding

Version 0.1.0

Author Yuntong Hou [aut, cre] (<https://orcid.org/0009-0005-0587-4692>),
Yan Zhong [aut, ctb] (<https://orcid.org/0000-0003-2412-043X>)

Description This package provides a workflow to jointly embed chromatin accessibility peaks and expressed genes into a shared low-dimensional space using single-cell ATAC-seq (scATAC-seq) and RNA-seq (scRNA-seq) data. It integrates regulatory relationships among peak-peak interactions (via Cicero), peak-gene interactions (using Lasso, random forest, and XG-Boost), and gene-gene interactions (using principal component regression). Additionally, it supports the comparison of regulatory genes between two conditions through manifold alignment implemented in scTenifoldNet.

URL <https://github.com/Houyt23/scPOEM>

BugReports <https://github.com/Houyt23/scPOEM/issues>

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports methods, utils, stats, foreach, doParallel, tictoc, Matrix, glmnet, xgboost, reticulate, stringr, magrittr, monocle3, cicero, scTenifoldNet, SingleCellExperiment

Depends R (>= 3.5.0)

RoxygenNote 7.3.2

Contents

align_embedding	2
eNN	3
GGN	4
input_compare	5
input_single	5
PGN_Lasso	6
PGN_RF	7
PGN_XGBoost	8
pg_embedding	9
PPN	11
scPOEM	12

Index	15
--------------	-----------

align_embedding	<i>align_embedding</i>
-----------------	------------------------

Description

Manifold alignment in scTenifoldNet is applied to genes across two conditions using eNN constructed from separate gene representations.

Usage

```
align_embedding(
  gene_data1,
  gene_node1,
  E1,
  gene_data2,
  gene_node2,
  E2,
  dirpath,
  d = 100
)
```

Arguments

gene_data1	The information for genes in state1, must have a col names 'gene_name'.
gene_node1	Gene ids that are associated with other peaks or genes in state1.
E1	Embedding representations of peaks and genes in state1.
gene_data2	The information for genes in state2, must have a col names 'gene_name'.
gene_node2	Gene ids that are associated with other peaks or genes in state2.
E2	Embedding representations of peaks and genes in state2.
dirpath	The folder path to read or write file
d	The dimension of latent space.

Value

a list containing the following

manifoldAlignment	Embedding representations of genes in two conditions
diffRegulation	A list of differential regulation information for each gene

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
dirpath <- "../example_data"
# Download compare mode example data
data(input_compare)
data_S1 <- input_compare$S1
data_S2 <- input_compare$S2
gg_net1 <- GGN(data_S1$Y, file.path(dirpath, "compare/S1"))
```

```
pp_net1 <- PPN(data_S1$X, data_S1$peak_data, data_S1$cell_data,
               data_S1$genome, file.path(dirpath, "compare/S1"))
net_Lasso1 <- PGN_Lasso(data_S1$X, data_S1$Y,
                       data_S1$gene_data, data_S1$neighbor_peak,
                       file.path(dirpath, "compare/S1"))
net_RF1 <- PGN_RF(data_S1$X, data_S1$Y, data_S1$gene_data,
                  data_S1$neighbor_peak, file.path(dirpath, "compare/S1"))
net_XGB1 <- PGN_XGBoost(data_S1$X, data_S1$Y,
                       data_S1$gene_data, data_S1$neighbor_peak,
                       file.path(dirpath, "compare/S1"))
E_result_S1 <- pg_embedding(gg_net1, pp_net1, net_lasso1, net_RF1,
                           net_XGB1, file.path(dirpath, "compare/S1"))

gg_net2 <- GGN(data_S2$Y, file.path(dirpath, "compare/S2"))
pp_net2 <- PPN(data_S2$X, data_S2$peak_data,
               data_S2$cell_data, data_S2$genome,
               file.path(dirpath, "compare/S2"))
net_Lasso2 <- PGN_Lasso(data_S2$X, data_S2$Y,
                       data_S2$gene_data, data_S2$neighbor_peak,
                       file.path(dirpath, "compare/S2"))
net_RF2 <- PGN_RF(data_S2$X, data_S2$Y, data_S2$gene_data,
                  data_S2$neighbor_peak, file.path(dirpath, "compare/S2"))
net_XGB2 <- PGN_XGBoost(data_S2$X, data_S2$Y,
                       data_S2$gene_data, data_S2$neighbor_peak,
                       file.path(dirpath, "compare/S2"))
E_result_S2 <- pg_embedding(gg_net2, pp_net2, net_lasso2, net_RF2,
                           net_XGB2, file.path(dirpath, "compare/S2"))

compare_result <- align_embedding(gene_data1,
                                  E_result_S1$gene_node,
                                  E_result_S1$E,
                                  gene_data2,
                                  E_result_S2$gene_node,
                                  E_result_S2$E,
                                  file.path(dirpath, "compare/compare"))

## End(Not run)
```

eNN
 eNN

Description

Make gene-gene net after meta-path based embedding via epsilon-NN.

Usage

$$eNN(E_g)$$

Arguments

E_g Embedding representations of genes.

Value

The eNN network.

GGN	<i>GGN</i>
-----	------------

Description

Make gene-gene net via principle component regression.

Usage

```
GGN(Y, dirpath, count_device, nComp = 5, rebuild_GGN = T)
```

Arguments

<code>Y</code>	The scRNA-seq data, sparse matrix.
<code>dirpath</code>	The folder path to read or write file.
<code>count_device</code>	The number of cpus used to train the Lasso model.
<code>nComp</code>	The number of PCs used for regression
<code>rebuild_GGN</code>	Logical. Whether to rebuild the gene-gene network (GGN) from scratch. If FALSE, the function will attempt to read from 'GGN.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.

Value

The GGN network.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
dirpath <- "../example_data"
# Download single mode example data
data(input_single)
# Construct GGN net.
gg_net <- GGN(input_single$Y, file.path(dirpath, "single"))

## End(Not run)
```

`input_compare`*Example Input Data for Compare Mode Analysis*

Description

A list containing example single-cell multi-omics data used in "compare" mode of the scPOEM package.

Usage

```
data(input_compare)
```

Format

A named list of length 2. Each element is itself a named list with the following components:

X Gene expression matrix.

Y Peak accessibility matrix.

peak_data A data.frame containing peak information.

gene_data A data.frame containing gene information (must contain column "gene_name").

cell_data A data.frame containing cell metadata.

neibor_peak The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0.

genome The genome length for the species.

Examples

```
data(input_compare)
```

`input_single`*Example Input Data for Single Mode Analysis*

Description

A list containing example single-cell multi-omics data used in "single" mode of the scPOEM package.

Usage

```
data(input_single)
```

Format

A named list with 7 elements:

X Gene expression matrix.

Y Peak accessibility matrix.

peak_data A data.frame containing peak information.

gene_data A data.frame containing gene information (must contain column "gene_name").

cell_data A data.frame containing cell metadata.

neibor_peak The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0.

genome The genome length for the species.

Examples

```
data(input_single)
```

PGN_Lasso

PGN_Lasso

Description

Make peak-gene net via Lasso.

Usage

```
PGN_Lasso(  
  X,  
  Y,  
  gene_data,  
  neibor_peak,  
  dirpath,  
  count_device,  
  rebuild_PGN_Lasso  
)
```

Arguments

X	The scATAC-seq data, sparse matrix.
Y	The scRNA-seq data, sparse matrix.
gene_data	The information for genes, must have a col names 'gene_name'.
neibor_peak	The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0.
dirpath	The folder path to read or write file.
count_device	The number of cpus used to train the Lasso model.
rebuild_PGN_Lasso	Logical. Whether to rebuild the peak-gene network via Lasso from scratch. If FALSE, the function will attempt to read from 'PGN_Lasso.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.

Value

The PGN_Lasso network.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
dirpath <- "../example_data"
# Download single mode example data
data(input_single)
# Construct PGN net via Lasso.
net_Lasso <- PGN_Lasso(input_single$X,
                        input_single$Y,
                        input_single$gene_data,
                        input_single$neibor_peak,
                        file.path(dirpath, "single"))

## End(Not run)
```

PGN_RF

*PGN_PF***Description**

Make peak-gene net via random forest.

Usage

```
PGN_RF(
  X,
  Y,
  gene_data,
  neibor_peak,
  dirpath,
  count_device = 1,
  rebuild_PGN_RF = T,
  seed = 0
)
```

Arguments

X	The scATAC-seq data, sparse matrix.
Y	The scRNA-seq data, sparse matrix.
gene_data	The information for genes, must have a col names 'gene_name'.
neibor_peak	The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0.
dirpath	The folder path to read or write file.
count_device	The number of cpus used to train the Lasso model.

rebuild_PGN_RF Logical. Whether to rebuild the peak-gene network via random forest from scratch. If FALSE, the function will attempt to read from 'PGN_RF.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.

seed An integer specifying the random seed to ensure reproducible results.

Value

The PGN_RF network.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
dirpath <- "../example_data"
# Download single mode example data
data(input_single)
# Construct PGN net via random forest (RF).
net_RF <- PGN_RF(input_single$X,
                 input_single$Y,
                 input_single$gene_data,
                 input_single$neibor_peak,
                 file.path(dirpath, "single"))

## End(Not run)
```

PGN_XGBoost

PGN_XGBoost

Description

Make peak-gene net via XGBoost.

Usage

```
PGN_XGBoost(
  X,
  Y,
  gene_data,
  neibor_peak,
  dirpath,
  count_device = 1,
  rebuild_PGN_XGB = T
)
```

Arguments

X The scATAC-seq data, sparse matrix.

Y The scRNA-seq data, sparse matrix.

gene_data The information for genes, must have a col names 'gene_name'.

neibor_peak	The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0.
dirpath	The folder path to read or write file.
count_device	The number of cpus used to train the Lasso model.
rebuild_PGN_XGB	Logical. Whether to rebuild the peak-gene network via XGBoost from scratch. If FALSE, the function will attempt to read from 'PGN_XGB.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.

Value

The PGN_XGBoost network.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
dirpath <- "../example_data"
# Download single mode example data
data(input_single)
# Construct PGN net via XGBoost.
net_XGB <- PGN_XGBoost(input_single$X,
                       input_single$Y,
                       input_single$gene_data,
                       input_single$neibor_peak,
                       file.path(dirpath, "single"))

## End(Not run)
```

pg_embedding

pg_embedding

Description

Learn the low-dimensional representations for peaks and genes with a meta-path based method.

Usage

```
pg_embedding(
  gg_net,
  pp_net,
  net_lasso,
  net_RF,
  net_XGB,
  dirpath,
  relearn_pg_embedding = T,
  d = 100,
  seed = 0
)
```

Arguments

gg_net	The gene-gene network.
pp_net	The peak-peak network.
net_lasso	The peak-gene network constructed by Lasso.
net_RF	The peak-gene network constructed by Random Forest.
net_XGB	The peak-gene network constructed by XGBoost.
dirpath	The folder path to read or write file.
relearn_pg_embedding	Logical. Whether to relearn the low-dimensional representations for peaks and genes from scratch. If FALSE, the function will attempt to read from 'node_embeddings.mtx', 'node_used_peak.csv', 'node_used_gene.csv' under dirpath/embedding in single mode or dirpath/state_name/embedding in compare mode.
d	The dimension of latent space.
seed	An integer specifying the random seed to ensure reproducible results.

Value

a list containing the following

E low-dimensional representations of peaks and genes

peak_node Peak ids that are associated with other peaks or genes.

gene_node Gene ids that are associated with other peaks or genes.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
library(data.table)
dirpath <- "./example_data"
# Download single mode example data
data(input_single)
gg_net <- GGN(input_single$Y, file.path(dirpath, "single"), 1, 5, T)
pp_net <- PPN(input_single$X, input_single$peak_data,
              input_single$cell_data, input_single$genome,
              file.path(dirpath, "single"))
net_Lasso <- PGN_Lasso(input_single$X, input_single$Y,
                      input_single$gene_data, input_single$neighbor_peak,
                      file.path(dirpath, "single"))
net_RF <- PGN_RF(input_single$X, input_single$Y,
                 input_single$gene_data, input_single$neighbor_peak,
                 file.path(dirpath, "single"))
net_XGB <- PGN_XGBoost(input_single$X, input_single$Y,
                       input_single$gene_data, input_single$neighbor_peak,
                       file.path(dirpath, "single"))
E_result <- pg_embedding(gg_net, pp_net, net_lasso, net_RF, net_XGB,
                         file.path(dirpath, "single"))

## End(Not run)
```

PPN

*PPN***Description**

Make peak-peak net.

Usage

```
PPN(X, peak_data, cell_data, genome, dirpath, rebuild_PPN = T, seed = 0)
```

Arguments

X	The scATAC-seq data, sparse matrix.
peak_data	The information for peaks, must have a col names 'peak_name'.
cell_data	The information for cells, must have a col names 'cell_name'.
genome	The genome length for the species.
dirpath	The folder path to read or write file.
rebuild_PPN	Logical. Whether to rebuild the peak-peak network (PPN) from scratch. If FALSE, the function will attempt to read from 'PPN.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
seed	An integer specifying the random seed to ensure reproducible results.

Value

The PPN network.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
library(data.table)
dirpath <- "../example_data"
# Download single mode example data
data(input_single)
# Construct PPN net.
pp_net <- PPN(input_single$X,
              input_single$peak_data,
              input_single$cell_data,
              input_single$genome,
              file.path(dirpath, "single"))

## End(Not run)
```

scPOEM

scPOEM

Description

A embedding method that jointly projects chromatin accessibility peaks and expressed genes into a shared low-dimensional space.

Usage

```
scPOEM(
  mode = c("single", "compare"),
  input_data,
  dirpath,
  count_device = 1,
  nComp = 5,
  seed = 0,
  d = 100,
  rebuild_GGN = T,
  rebuild_PPN = T,
  rebuild_PGN_Lasso = T,
  rebuild_PGN_RF = T,
  rebuild_PGN_XGB = T,
  relearn_pg_embedding = T
)
```

Arguments

mode	The mode indicating whether to analyze data from a single condition or to compare two conditions.
input_data	<p>A list of input data.</p> <p>If mode = "single", input_data must be a list containing the following seven objects:</p> <ul style="list-style-type: none"> • X: Gene expression matrix. • Y: Peak accessibility matrix. • peak_data: A data.frame containing peak information. • gene_data: A data.frame containing gene information (must contain column "gene_name"). • cell_data: A data.frame containing cell metadata. • neighbor_peak: The peak IDs within a certain range of each gene, must have cols c("gene_name", "start_use", "end_use"). The id numbers in "start_use" and "end_use" are start from 0. • genome: The genome length for the species. <p>If mode = "compare", input_data must be a named list of two elements, with names corresponding to two state names (e.g., "state1" and "state2"). Each element must itself be a list containing the same seven components as described above for mode = "single".</p>
dirpath	The folder path to read or write file.
count_device	The number of cpus used to train models.

nComp	The number of PCs used for regression in constructing GGN.
seed	An integer specifying the random seed to ensure reproducible results.
d	The dimension of latent space.
rebuild_GGN	Logical. Whether to rebuild the gene-gene network from scratch. If FALSE, the function will attempt to read from 'GGN.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
rebuild_PPN	Logical. Whether to rebuild the peak-peak network from scratch. If FALSE, the function will attempt to read from 'PPN.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
rebuild_PGN_Lasso	Logical. Whether to rebuild the peak-gene network via Lasso from scratch. If FALSE, the function will attempt to read from 'PGN_Lasso.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
rebuild_PGN_RF	Logical. Whether to rebuild the peak-gene network via random forest from scratch. If FALSE, the function will attempt to read from 'PGN_RF.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
rebuild_PGN_XGB	Logical. Whether to rebuild the peak-gene network via XGBoost from scratch. If FALSE, the function will attempt to read from 'PGN_XGB.mtx' under dirpath/test in single mode or dirpath/state_name/test in compare mode.
relearn_pg_embedding	Logical. Whether to relearn the low-dimensional representations for peaks and genes from scratch. If FALSE, the function will attempt to read from 'node_embeddings.mtx', 'node_used_peak.csv', 'node_used_gene.csv' under dirpath/embedding in single mode or dirpath/state_name/embedding in compare mode.

Value

The scPOEM result.

Examples

```
## Not run:
library(scPOEM)
library(Matrix)
library(data.table)
dirpath <- "./example_data"
# An example for analysing a single dataset.
# Download and read data.
data(input_single)
single_result <- scPOEM(mode = "single",
                        input_data=input_single,
                        dirpath=file.path(dirpath, "single"))

# An example for analysing and comparing datasets from two conditions.
# Download compare mode example data
data(input_compare)
compare_result <- scPOEM(mode = "compare",
                        input_data=input_compare,
                        dirpath=file.path(dirpath, "compare"))
```

```
## End(Not run)
```

Index

* datasets

input_compare, [5](#)

input_single, [5](#)

align_embedding, [2](#)

eNN, [3](#)

GGN, [4](#)

input_compare, [5](#)

input_single, [5](#)

pg_embedding, [9](#)

PGN_Lasso, [6](#)

PGN_RF, [7](#)

PGN_XGBoost, [8](#)

PPN, [11](#)

scPOEM, [12](#)