Package

October 19, 2018

Type Package

Title Low-Rank Factorization for Subspace-clustering

Version 0.0.0.9000

Maintainer Matt Karikomi <mattkarikomi@gmail.com>

Description Clustering based on single-cell data.
Population signaling profiles based on expression pathway knowledge.

Depends R (>= 2.10)

License MIT

Encoding UTF-8

LazyData true

Imports FNN (>= 1.1.2.1),
stringr (>= 1.3.1),
psych (>= 1.8.4),
igraph (>= 1.2.2),
ggplot2 (>= 3.0.0),
Matrix (>= 1.2-14),
tsne (>= 0.1-3),
Rtsne (>= 0.13),
pracma (>= 2.1.4),
RSpectra (>= 0.13-1),
NNLM (>= 0.4.2),
knitr (>= 1.20)

Suggests testthat,
rmarkdown

RoxygenNote 6.1.0

VignetteBuilder knitr

R topics documented:

`computM` *Perform ADMM*

Description

Computing cell-to-cell similarity matrix by solving the following optimization problem via ADMM

Usage

```
computM(D, X, lambda)
```

Arguments

`D`	the unweighted KNN adjacency matrix
`X`	normalized sample vectors
`lambda`	the balance term between the rank of Z and the error

Details

min_Z,E ||Z||_* + lambda ||E||_2,1 s.t. X = XZ + E; Z'1 = 1; Z_i,j = 0 for (i,j)\in Omega

Value

a list containing the low rank approximation of X and manifold learning error

CountClusters *Use clustering consensus to infer cluster number*

Description

Use clustering consensus to infer cluster number

Usage

```
CountClusters(data, tol = 0.01, range = 1:20, eigengap = TRUE)
```

Arguments

data	a symmetric nonnegative similarity matrix
tol	cutoff for lambda zero
range	a vector specifying the min and max number of clusters to iteratively test when building the consensus matrix
eigengap	whether or not to use the max eigengap (upper bound) cluster count

Value

the number of clusters

FindRootCell *Infer the root cell*

Description

Use either the primary manifold embedding of cell similarity to find the root cell by cluster-cell rank-correlation or use the flattened representation of this embedding to find the root cell by maximum separation heuristic possible values for cluster_order_by are predecessor and distance possible values for cell_order_by are index and distance

Usage

```
FindRootCell(use_flat_dist = TRUE, cluster_order_by = "distance",
  cell_order_by = "distance", graph_cluster_mst = NULL,
  dist_graph = NULL, dist_flat, cluster_labels = NULL,
  root_cluster = NULL)
```

Arguments

use_flat_dist	use max path length heuristic on flat embedding
cluster_order_by	cluster rank parameter for measuring cluster/cell rank correlation
cell_order_by	cell-wise pseudotime parameter for measuring cluster/cell rank correlation

`graph_cluster_mst`
: igraph object representing an mst on the cluster-cluster graph

`dist_graph`
: a distance matrix of cells embedded in a graph

`dist_flat`
: the manifold embedding of the cells

`cluster_labels`
: the cluster label for each cell

`root_cluster`
: the id of the root cluster on the cluster-cluster graph

Value

integer index of the root cell

Examples

```
root_cell <- FindRootCell(dist_flat = RSoptSC::GuoPtime$Params$latent)
```

`FindRootCluster` *Find the root cluster, given a weighted adjacency matrix Generate a cluser-to-cluster graph based on a similarity-matrix of connected cells Generate the minimum spanning tree on the cluster-to-cluster graph Use one of the max dist clusters from this graph, unless The cluster is given, then proceed to find the root cell The root cell is given, then use its cluster*

Description

Find the root cluster, given a weighted adjacency matrix Generate a cluser-to-cluster graph based on a similarity-matrix of connected cells Generate the minimum spanning tree on the cluster-to-cluster graph Use one of the max dist clusters from this graph, unless The cluster is given, then proceed to find the root cell The root cell is given, then use its cluster

Usage

```
FindRootCluster(cluster_labels, flat_embedding, dist_graph, dist_flat,
  reverse = FALSE)
```

Arguments

`cluster_labels`
: the cluster label for each cell

`flat_embedding`
: rows are cells and columns are coordinates in n-col space

`dist_graph`
: distances on a connected graph

`dist_flat`
: flat distance matrix for all cells

`reverse`
: a boolean variable whether to take the root cluster based on minimum dispersion on the flat embedding

Value

a list containing:

`cluster_adj_matrix`
: a weighted upper-triangular adjacency matrix for the clusters based on avg pseudotime between their cells

`graph_cluster`
: an igraph object on the clusters, with a minimum spanning tree

`root_cluster` index of the root cluster

`cluster_mst` an igraph mst object

Examples

```
FindRootCluster(cluster_labels = RSoptSC::GuoPtime$Params$cluster_label,
          flat_embedding = RSoptSC::GuoPtime$Params$latent,
          dist_graph = RSoptSC::GuoPtime$Values$Short_pathd,
          dist_flat = RSoptSC::GuoPtime$Values$low_dis,
          reverse = FALSE)
```

GetComponents *Use the graph laplacian to get the number of graph components*

Description

Use the graph laplacian to get the number of graph components

Usage

```
GetComponents(data, tol = 0.01)
```

Arguments

`data`	a symmetric nonnegative similarity matrix
`tol`	cutoff for lambda zero

Value

the number of components

GetConsensus *Produce a consensus matrix*

Description

Produce a consensus matrix

Usage

```
GetConsensus(clusters)
```

Arguments

clusters	the cluster assignment of each cell

Value

a consensus matrix

GetDGFromTable *Get a weighted, directed graph from a table of edges and weights*

Description

Get a weighted, directed graph from a table of edges and weights

Usage

```
GetDGFromTable(directed_edge_table)
```

Arguments

edge_table	a numeric matrix whose rows are edges, col 1 is v1, col2 is v2, col3 is weight

Value

a directed igraph object

GetDominatorTree *Get a directed graph from a predecessor vector*

Description

Get a directed graph from a predecessor vector

Usage

```
GetDominatorTree(predecessors)
```

Arguments

predecessors	the tree to be directionalized

Value

a directed igraph object

GetEnsemble *Produce a truncated ensemble consensus matrix*

Description

Produce a truncated ensemble consensus matrix

Usage

```
GetEnsemble(data, tol, n_prcs = 3, tau, range = 1:20,
  method = "kmeans")
```

Arguments

data	a symmetric nonnegative similarity matrix
tol	cutoff for lambda zero
tau	the drop tolerance, controlling the sparsification (uncoupling) of the consensus matrix
range	a vector specifying the min and max number of clusters to iteratively test when building the consensus matrix
method	the clustering method for building consensus clusters
prcs_dim	the number of pcs to use for clustering method

Value

a truncated ensemble consensus matrix

GetMarkerTable *Get the marker genes for each cluster*

Description

Get the marker genes for each cluster

Usage

```
GetMarkerTable(counts_data, cluster_labels, H, gene_expression_threshold,
  n_features)
```

Arguments

cluster_labels	a vector of cluster labels
H	a nonnegative matrix such that W = H*t(H), H_i,j is the cells_weight by which cell i belongs to the jth cluster
gene_expression_threshold	for n cells, for gene_expression_threshold = m, dont consider genes expressed in more than n-m cells or genes expressed in less than m cells
n_features	number of marker genes per cluster to retrieve
M	a matrix of expression values for each cell (rows) and gene (columns)

Value

a table of marker genes

GetPredecessors *Get the predecessor vector for a dominator tree encoded as an undirected tree and root*

Description

Get the predecessor vector for a dominator tree encoded as an undirected tree and root

Usage

```
GetPredecessors(minspantree, root)
```

Arguments

minspantree	the tree to be directionalized
root	the id of the root

Value

a vector[] of node ids, where vector[i] is the predecessor of node i

Examples

```
predecessors <- GetPredecessors(minspantree = RSoptSC::GuoPtimeFull$mst, root = 6)
```

`GetSignalingPartners` *Compute Cell-cell interaction probability*

Description

We compute three expressions. Here is a sample: $K_{i,j} = \frac{\alpha_{i,j}}{\alpha_{i,j}+\beta_{i,j}}$

Usage

```
GetSignalingPartners(counts_data, cluster_labels, H,
  gene_expression_threshold, n_features)
```

Arguments

`cluster_labels`	a vector of cluster labels
`H`	a nonnegative matrix such that W = H*t(H), H_i,j is the cells_weight by which cell i belongs to the jth cluster
`gene_expression_threshold`	for n cells, for `gene_expression_threshold` = m, dont consider genes expressed in more than n-m cells or genes expressed in less than m cells
`n_features`	number of marker genes per cluster to retrieve
`M`	a matrix of expression values for each cell (rows) and gene (columns)

Value

a table of marker genes

`GSE67602_Joost` *Skin Cell scRNA-seq Data from Joost et al*

Description

A dataset containing sequencing on over 20k genes across 719 annotated cells

Usage

```
GSE67602_Joost
```

Format

A list with three items:

annotation the type of cell (Basal IFE-DI IFE-DII IFE-KI IFE-KII)

data values are mRNA counts, row names are genes, column names are cells

InitSVD *Initialize non-negative factorization of the similarity matrix*

Description

Initialize non-negative factorization of the similarity matrix

Usage

```
InitSVD(A, k)
```

Arguments

A	The similarity matrix.
k	The rank of the output.

Value

W and H such that A = W * H.

JoinGraphComponents *In case the graph is not connected, join the components This function updates the original adjacency matrix and returns a new object.*

Description

In case the graph is not connected, join the components This function updates the original adjacency matrix and returns a new object.

Usage

```
JoinGraphComponents(root_cell, adjacency_matrix, flat_distances,
  n_components, component_members)
```

Arguments

root_cell	the root cell of the lineage tree
adjacency_matrix	the graph embedding of the cells
flat_distances	the flattened embedding of the cells
n_components	the number of components
component_members	a list of vectors containing the cells in each component

Value

adjacency_matrix such that new edges between disconnected components have length 2

Examples

```
my_matrix <- JoinGraphComponents(root_cell = RSoptSC::GuoPtime$Values$root_cell0,
    adjacency_matrix = RSoptSC::GuoPtime$Values$W_graph1,
    flat_distances = RSoptSC::GuoPtime$Values$low_dis,
    n_components = RSoptSC::GuoPtime$Values$nComponents,
    component_members = RSoptSC::GuoPtime$Values$members)
```

PlotMatlabDtree *Produce a plot of matlab DTree data and return the object If an output dir and filename are provided, a plot will be saved, otherwise the function will just return the graph*

Description

Produce a plot of matlab DTree data and return the object If an output dir and filename are provided, a plot will be saved, otherwise the function will just return the graph

Usage

```
PlotMatlabDtree(edge_table, predecessors, outputdir = NULL,
  outputfile = NULL)
```

Arguments

edge_table	a numeric matrix whose rows are directed edges of a tree: col 1 is v1, col2 is v2, col3 is weight
predecessors	a vector of tree predecessors such that pred[i] = the predecessor of i
outputdir	the output directory, relative to getwd()
outputfile	the output file

Value

an igraph representation of the tree

ProcessMatlabDTree *Convert a matlab edge table and predecessor list into a directed weighted edge table*

Description

Convert a matlab edge table and predecessor list into a directed weighted edge table

Usage

```
ProcessMatlabDTree(edge_table, predecessors)
```

Arguments

edge_table	a numeric matrix whose rows are undirected edges of a tree, col 1 is v1, col2 is v2, col3 is weight
predecessors	a vector of tree predecessors such that pred[i] = the predecessor of i

Value

a weighted, directed edge table

PseudotimeScatterPlot

Produce a scatter plot of the cells on selected 2-dim ebedding colored by pseudotime Here pseudotime is defined as the distance from the root cell according to the pseudotime metric recorded in `pseudotime`

Description

If an output dir and filename are provided, a plot will be saved, otherwise just return the plot

Usage

```
PseudotimeScatterPlot(flat_embedding, pseudotime, outputdir = NULL,
  outputfile = NULL)
```

Arguments

flat_embedding	a low dim embedding of cells
pseudotime	a scalar representation of pseudotime
outputdir	the output directory, relative to getwd()
outputfile	the output file

Value

a ggplot2 object

RepresentationMap

generate convenient representations of the data representations of the data are necessary for subsequent lineage analysis

---

Description

generate convenient representations of the data representations of the data are necessary for subsequent lineage analysis

Usage

```
RepresentationMap(flat_embedding = NULL, similarity_matrix,
  join_components = TRUE)
```

Arguments

flat_embedding	optionally provided low dim embedding, if not then 2d tsne will be used
similarity_matrix	the graphical embedding of the cells
join_components	boolean, whether or not to join disconnected components of the similarity matrix

Value

a list containing:

dist_flat	distance matrix on flat embedding
dist_graph	distance matrix on graph
adj_matrix	unweighted adjacency matrix
flat_embedding	the low dimensional embedding of the cells
similarity_graph	igraph object on the unweighted adjacency matrix
components	an igraph components object based on similarity_graph
n_components	number of components in components
sizes	sizes of components in components
members	list of members of components, sorted by component size and member index

Examples

```
embeddings <- RepresentationMap(flat_embedding = RSoptSC::GuoPtime$Params$latent,
    similarity_matrix = RSoptSC::GuoPtime$Params$W,
    join_components = FALSE)
```

SelectData — *Return a set of the most variable genes First filter using the expression threshold Then use the coefficient of the top variance PCA components to determine the variability of the gene*

Description

Return a set of the most variable genes First filter using the expression threshold Then use the coefficient of the top variance PCA components to determine the variability of the gene

Usage

```
SelectData(M, gene_expression_threshold, n_features)
```

Arguments

M	a matrix of expression values for each gene (rows) and cell (columns)
gene_expression_threshold	for n cells, for gene_expression_threshold = m, dont consider genes expressed in more than n-m cells or genes expressed in less than m cells
n_features	number of marker genes per cluster to retrieve

Value

a table of features (rows) and samples (columns)

`SimilarityM` *Compute the similarity matrix*

Description

Computes low dim embedding, constructs KNN graph on the embedding -> unweighted adjacency Calls manifold learning algorithm which uses the normalized sample vectors and the unweighted adjacency matrix to compute a low rank approximation of the data

Usage

```
SimilarityM(lambda, data)
```

Arguments

`lambda`	the balance term between the rank of Z and the error
`data`	the expression data, where each column is treated as a normalized vector

Value

a list containing the symetric cell to cell similarity matrix and manifold learning error

---

`SymNMF` *Factor a matrix based on an initial factor guess from SVD*

Description

Factor a matrix based on an initial factor guess from SVD

Usage

```
SymNMF(A, nC, H, gamma = 1e-06, mu = 10^(-6), maxiter = 1e+06)
```

Arguments

`A`	The similarity matrix.
`k`	The rank of the output.

Value

`W` and `H` such that `A` = `W` * `D` * `H`.

Index