Package ‘SLICER’

August 22, 2017

Type Package

Title Selective Locally Linear Inference of Cellular Expression Relationships

Version 0.2.0

Author Joshua Welch

Maintainer Joshua Welch <jwelch@cs.unc.edu>

Description Provides an implementation of SLICER, an algorithm for inferring cellular trajectories from single cell RNA sequencing data. See Welch, JD, Hartemink AJ, Prins JF (2016) <doi:10.1186/s13059-016-0975-3>.

License ACM

LazyData TRUE

Imports alphahull, igraph, lle, grDevices, graphics, stats

Depends R (>= 2.10)

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-22 17:13:19 UTC

R topics documented:

`assign_branches` *Detect branches in the trajectory and assign cells to branches*

Description

This function uses geodesic entropy to automatically determine the number and location of branches in the trajectory. Each cell is then assigned to the corresponding branch.

Usage

```
assign_branches(traj_graph, start, min_branch_len = 10,
  cells = V(traj_graph))
```

Arguments

`traj_graph`	Nearest neighbor graph built from LLE embedding
`start`	Index of start cell
`min_branch_len`	Minimum number of cells required to call a branch
`cells`	List of indices indicating which cells to assign to branches (used for recursive calls; not intended to be set by users).

Value

Vector of integers assigning each cell to a branch

Examples

```
## Not run:
traj_lle = lle::lle(traj[,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
start=1
branches = assign_branches(traj_graph,start)
plot(traj_lle,pch=16,col=branches)

## End(Not run)
```

cell_order *Sort cells according to their progress through a process*

Description

Uses the values computed by `process_distance` to order cells.

Usage

```
cell_order(traj_graph, start)
```

Arguments

`traj_graph`	Nearest neighbor graph built from LLE embedding
`start`	Index of starting cell

Value

Sorted vector of cell indices

Examples

```
genes=1:200
cells=sample(1:500,30)
data(traj)
k=10
traj_lle = lle::lle(traj[cells,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
start=1
cells_ordered = cell_order(traj_graph,start)
```

compute_geodesic_entropy *Compute the geodesic entropy profile of a trajectory*

Description

The geodesic entropy of a trajectory can be used to detect branches. This function computes geodesic entropy and produces a plot that can be used to visually confirm the branches detected by `assign_branches`.

Usage

```
compute_geodesic_entropy(traj_graph, start)
```

Arguments

traj_graph	Nearest neighbor graph built from LLE embedding
start	Index of start cell

Value

Vector of geodesic entropy values. Item k is the geodesic entropy k steps away from the start cell.

Examples

```
## Not run:
traj_lle = lle(traj[,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
start=1
compute_geodesic_entropy(traj_graph,start)

## End(Not run)
```

conn_knn_graph *Construct a k-nearest neighbor graph that is fully connected*

Description

This function constructs a k-nearest neighbor graph using an LLE embedding, then adds the minimum number of edges needed to make the graph fully connected.

Usage

```
conn_knn_graph(embedding, k)
```

Arguments

embedding	Low-dimensional LLE embedding of cells
k	Number of nearest neighbors

Value

An igraph object corresponding to the k-NN graph

Examples

```
genes=1:200
cells=sample(1:500,30)
k=10
traj_lle = lle::lle(traj[cells,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
```

`detect_cell_types` *Identify clusters corresponding to putative cell types*

Description

`detect_cell_types` divides the k-nearest neighbor graph (built from the LLE embedding) into connected components. These connected components represent clusters of cells corresponding to putative cell types.

Usage

```
detect_cell_types(embedding, k)
```

Arguments

`embedding`	Low-dimensional LLE embedding of cells
`k`	Number of nearest neighbors to use when detecting clusters

Value

Vector containing a numerical cluster assignment for each cell

`find_extreme_cells` *Identify candidate start cells for the trajectory*

Description

Plots the embedding generated by LLE and highlights potential starting cells for the trajectory. The candidates are chosen based on the longest shortest path through the nearest neighbor graph.

Usage

```
find_extreme_cells(traj_graph, embedding)
```

Arguments

`traj_graph`	Nearest neighbor graph built from LLE embedding
`embedding`	Low-dimensional LLE embedding of cells

Value

Indices of potential starting cells

Examples

```
genes=1:200
cells=sample(1:500,30)
k=10
traj_lle = lle::lle(traj[cells,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
find_extreme_cells(traj_graph,traj_lle)
```

---

graph_gene *Plot trajectory colored by expression level of a gene*

Description

This function plots the embedding produced by LLE, coloring cells by their expression levels of a gene of interest.

Usage

```
graph_gene(exp_mat, embedding, samples, gene_ind, cell_symbols = 16,
  title = "Gene Expression")
```

Arguments

`exp_mat`	Matrix of expression levels
`embedding`	Low-dimensional LLE embedding of cells
`samples`	Indices of cells to include in the plot
`gene_ind`	Index of gene to use
`cell_symbols`	Symbols to use for plotting each cell
`title`	Plot title

Value

None

Examples

```
## Not run:
graph_gene(traj,traj_lle,1:nrow(traj),1)

## End(Not run)
```

`graph_process_distance` *Plot trajectory colored by process distance*

Description

This function plots the embedding produced by LLE, coloring cells by their progress through a process.

Usage

```
graph_process_distance(traj_graph, embedding, start, cell_symbols = 16)
```

Arguments

`traj_graph`	Nearest neighbor graph built from LLE embedding
`embedding`	Low-dimensional LLE embedding of cells
`start`	Index of start cell
`cell_symbols`	Symbols to use for plotting each cell

Value

None

Examples

```
genes=1:200
cells=sample(1:500,30)
k=10
traj_lle = lle::lle(traj[cells,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
start=1
graph_process_distance(traj_graph,traj_lle,start)
```

`process_distance` *Determine the position of each cell within the trajectory*

Description

This function calculates the geodesic distance from the start cell to each other cell. This value corresponds to the distance a cell has migrated through the process described by the cell trajectory.

Usage

```
process_distance(traj_graph, start)
```

Arguments

`traj_graph`	Nearest neighbor graph built from LLE embedding
`start`	Index of starting cell

Value

Vector of distances

Examples

```
genes=1:200
cells=sample(1:500,30)
k=10
traj_lle = lle::lle(traj[cells,genes],m=2,k)$Y
traj_graph = conn_knn_graph(traj_lle,5)
start = 1
dists = process_distance(traj_graph,start)
```

`select_genes` *Select genes to use in building a cell trajectory*

Description

This function uses "neighborhood variance" to identify genes that vary smoothly, rather than fluctuating randomly, across the set of cells. Genes selected in this way can then be used to construct a trajectory.

Usage

```
select_genes(embedding)
```

Arguments

`embedding`	Low-dimensional LLE embedding of cells

Value

Vector containing indices of selected genes

Examples

```
## Not run:
genes = select_genes(traj)

## End(Not run)
```

`select_k` *Select the number of nearest neighbors for LLE to use*

Description

`select_k` uses the alpha-hull to determine which value of k yields an embedding that most resembles a trajectory.

Usage

```
select_k(exp_mat, kmin = 5, kmax = 50, by = 5)
```

Arguments

`exp_mat`	Matrix of expression levels
`kmin`	Smallest value of k to try
`kmax`	Largest value of k to try
`by`	Increment

Value

The optimal value of k

Examples

```
## Not run:
genes = select_genes(traj)
k = select_k(traj[,genes])

## End(Not run)
```

`traj` *This is a dataset containing a synthetic branching trajectory.*

Description

This is a dataset containing a synthetic branching trajectory.

Usage

```
data(traj)
```

Format

A matrix with samples down rows and genes across columns `width_k` *Helper function for k selection*

Description

Helper function for k selection

Usage

```
width_k(k, traj_exp)
```

Arguments

`k`	Nearest neighbors
`traj_exp`	Cell expression matrix

Value

Width of LLE embedding

Index