# Package ‘SLICER’

## August 22, 2017

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Selective Locally Linear Inference of Cellular Expression Relationships</td>
</tr>
<tr>
<td>Version</td>
<td>0.2.0</td>
</tr>
<tr>
<td>Author</td>
<td>Joshua Welch</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Joshua Welch &lt;<a href="mailto:jwelch@cs.unc.edu">jwelch@cs.unc.edu</a>&gt;</td>
</tr>
</tbody>
</table>
| 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) &lt;doi:10.1186/s13059-016-0975-3&gt;.
| License  | ACM |
| LazyData | TRUE |
| Imports  | alphahull, igraph, lle, grDevices, graphics, stats |
| Depends  | R (&gt;= 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:

<table>
<thead>
<tr>
<th>assign_branches</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>cell_order</td>
<td>3</td>
</tr>
<tr>
<td>compute_geodesic_entropy</td>
<td>3</td>
</tr>
<tr>
<td>conn_knn_graph</td>
<td>4</td>
</tr>
<tr>
<td>detect_cell_types</td>
<td>5</td>
</tr>
<tr>
<td>find_extreme_cells</td>
<td>5</td>
</tr>
<tr>
<td>graph_gene</td>
<td>6</td>
</tr>
<tr>
<td>graph_process_distance</td>
<td>7</td>
</tr>
<tr>
<td>process_distance</td>
<td>7</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>traj_graph</td>
<td>Nearest neighbor graph built from LLE embedding</td>
</tr>
<tr>
<td>start</td>
<td>Index of start cell</td>
</tr>
<tr>
<td>min_branch_len</td>
<td>Minimum number of cells required to call a branch</td>
</tr>
<tr>
<td>cells</td>
<td>List of indices indicating which cells to assign to branches (used for recursive calls; not intended to be set by users).</td>
</tr>
</tbody>
</table>

Value

Vector of integers assigning each cell to a branch

Examples

```r
## 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

```r
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

```r
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

```r
compute_geodesic_entropy(traj_graph, start)
```
conn_knn_graph

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

```r
## 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

```r
genes=1:200
cells=sample(1:500,30)
k=10
traj_lle = lli::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
graph_gene

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)

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**

```r
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**

```r
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**

```r
process_distance(traj_graph, start)
```
select_genes

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**

```r
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**

```r
## Not run:
genesis = select_genes(traj)
k = select_k(traj[,genesis])
## End(Not run)
```

**traj**

This is a dataset containing a synthetic branching trajectory.

**Description**

This is a dataset containing a synthetic branching trajectory.

**Usage**

```r
data(traj)
```

**Format**

A matrix with samples down rows and genes across columns
width_k  

**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

*Topic **datasets**
  traj, 9

assign_branches, 2

cell_order, 3
compute_geodesic_entropy, 3
conn_knn_graph, 4

detect_cell_types, 5

find_extreme_cells, 5

graph_gene, 6
graph_process_distance, 7

process_distance, 7

select_genes, 8
select_k, 9

traj, 9

width_k, 10