

dNetInduce

January 19, 2018

dNetInduce	<i>Function to generate a subgraph induced by given vertices and their k nearest neighbors</i>
------------	--

Description

dNetInduce is supposed to produce a subgraph induced by given vertices and its k nearest neighbors. The input is a graph of "igraph" or "graphNEL" object, a list of the vertices of the graph, and a k value for finding k nearest neighbors for these vertices. The output is a subgraph induced by given vertices plus their k neighbours. The resultant subgraph inherits the class from the input one. The induced subgraph contains exactly the vertices of interest, and all the edges between them.

Usage

```
dNetInduce(g, nodes_query, knn = 0, remove.loops = F, largest.comp = T,  
min.comp.size = 1)
```

Arguments

g	an object of class "igraph" or "graphNEL"
nodes_query	the vertices for which the calculation is performed
knn	an integer specifying how many k steps are used to find the nearest neighbours of the given vertices. By default, knn is set to zero; it means no neighbors will be considered. When knn is 1, the immediate neighbors of the given vertices will be also considered for inducing the subgraph. The same is true when knn is 2, etc
remove.loops	logical to indicate whether the loop edges are to be removed. By default, it sets to false
largest.comp	logical to indicate whether the largest component is only retained. By default, it sets to true for the largest component being left
min.comp.size	an integer specifying the minimum size of component that will be retained. This parameter only works when setting the false to keep the largest component. By default, it sets to 1 meaning all nodes will be retained

Value

- subg: an induced subgraph, an object of class "igraph" or "graphNEL". Appended with a node attribute 'comp' if multiple components are kept

Note

The given vertices plus their k nearest neighbors will be used to induce the subgraph.

See Also

[dNetInduce](#)

Examples

```
# 1) generate a random graph according to the ER model
g <- erdos.renyi.game(100, 1/100)

# 2) select the first 10 vertices as the query nodes
nodes_query <- V(g)[1:10]

# 3) produce the induced subgraph only based on the nodes in query
subg <- dNetInduce(g, nodes_query, knn=0)

# 4) produce the induced subgraph based on the nodes in query and their immediate neighbours
subg <- dNetInduce(g, nodes_query, knn=1)
```