Package ‘incgraph’

October 12, 2017

Type Package
Title Incremental Graphlet Counting for Network Optimisation
Version 1.0.1
Date 2017-10-02
Author Robrecht Cannoodt
Maintainer Robrecht Cannoodt <robrecht.cannoodt@gmail.com>
Description An efficient and incremental approach for calculating
the differences in orbit counts when performing single edge modifications
in a network. Calculating the differences in orbit counts is much more efficient than
recalculating all orbit counts from scratch for each time point.
License GPL-3
Depends R (>= 3.2)
Imports dplyr, methods, Rcpp (>= 0.11.4), orca, purrr, testthat,
tibble
LinkingTo Rcpp, BH
Encoding UTF-8
URL http://www.github.com/rcannood/incgraph
BugReports https://github.com/rcannood/incgraph/issues
RoxygenNote 6.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-10-12 08:50:00 UTC

R topics documented:
calculate.delta .................................................. 2
calculate.orbit.counts ........................................ 3
contains ............................................................. 4
flip ................................................................. 4
**Description**

calculate.delta calculates the changes in orbit counts as a result of a single edge modification.

**Usage**

calculate.delta(network, i, j)

**Arguments**

- network: An instance of the incgraph.network class
- i: A node in network
- j: A node in network

**Details**

This method iterates over and counts all graphlets which were added to or removed from the network due to one edge modification.

**Value**

A list containing two N-by-73 matrices, with N the number of nodes in the network and 1 column for each possible orbit. The value of list$\$add[i, j] (resp. list$\$rem[i, j]) is the number of times a subgraph was added to (resp. removed from) the network such that node i has orbit j in that subgraph.

**Author(s)**

Cannoodt Robrecht, <robrecht.cannoodt@gmail.com>

**References**

calculate.orbit.counts

See Also
See new.incgraph.network for examples and usage.

---

calculate.orbit.counts

*Calculate orbit counts from scratch*

---

**Description**

calculate.orbit.counts calculates the orbit counts of the current network.

**Usage**

calculate.orbit.counts(network)

**Arguments**

network An instance of the incgraph.network class

**Details**

The complete orbit counts is calculated using the count5 from the orca package.

Calling this method repeatedly becomes very inefficient for evolving networks. For evolving networks, the usage of calculate.delta is recommended.

For more details on this method, see Hočevar and Demšar (2014).

**Value**

An N-by-73 matrix, with N the number of nodes in the network and 1 column for each possible orbit. The value of mat[i,j] is the number of times node i has orbit j in a subgraph in the network.

**References**


**See Also**

See new.incgraph.network for examples and usage.
contains

Description
contains returns TRUE if the network contains the edge (i, j).

Usage
contains(network, i, j)

Arguments
- network: An instance of the incgraph.network class
- i: A node in network
- j: A node in network

Value
TRUE if the network contains (i, j)

See Also
See new.incgraph.network for examples and usage.

flip

Description
flip modifies an edge in the network. If it is contained in the network, it is removed from the network, otherwise it is added to the network.

Usage
flip(network, i, j)

Arguments
- network: An instance of the incgraph.network class
- i: A node in network
- j: A node in network

See Also
See new.incgraph.network for examples and usage.
**generate.dynamic.network**

*Generate a dynamic network*

**Description**

Generate a dynamic network

**Usage**

```r
generate.dynamic.network(
  model, amnt.nodes, amnt.edges, amnt.operations, trace = T, ...
)
```

```r
generate.geometric(amnt.nodes, amnt.edges, amnt.operations,
  amnt.dimensions = 3, trace = T)
```

```r
generate.barabasialbert(amnt.nodes, amnt.edges, amnt.operations,
  offset.exponent = 1, trace = T)
```

```r
generate.erdosrenyi(amnt.nodes, amnt.edges, amnt.operations, trace = T)
```

**Arguments**

- `model`: The network model with which to generate the network; "BA" for Barabási–Albert, "ER" for Erdős–Rényi, or "GEO" for geometric
- `amnt.nodes`: the number of nodes in the network at any given type
- `amnt.edges`: the number of edges in the network at any given type
- `amnt.operations`: the number of edge additions/deletions to generate
- `trace`: will print output text if TRUE
- `...`: extra parameters to pass to a specific network generator
- `amnt.dimensions`: (only GEO) the number of dimensions in which to operate
- `offset.exponent`: (only BA) the offset exponent for the weighted sampling

**Value**

A list containing the starting network network and the dynamic operations performed on it operations.

**Examples**

```r
# dyn.net.ba <- generate.dynamic.network("BA", 300, 300, 1000)
dyn.net.er <- generate.dynamic.network("ER", 300, 300, 1000)
dyn.net.geo <- generate.dynamic.network("GEO", 300, 300, 1000)
```
get.neighbours returns a vector of all neighbours of i.

Usage
get.neighbours(network, i)

Arguments

- network: An instance of the incgraph.network class
- i: A node in network

Value
Returns all neighbours of node i

See Also
See new.incgraph.network for examples and usage.
Examples

# Create a new (empty) network with 4 nodes
net <- new.incgraph.network(amnt.nodes = 4)

# Create a new network with 4 nodes and some edges
net <- new.incgraph.network(links = matrix(c(1L, 2L, 3L, 1L, 4L), ncol=2))

# Create a new network with 10 nodes and some edges
net <- new.incgraph.network(amnt.nodes = 10, links = matrix(c(1L, 2L, 3L, 1L, 4L), ncol=2))

# Create a more complex network from a matrix
mat <- matrix(c(1L, 2L, 1L, 3L, 1L, 4L, 1L, 5L, 1L, 6L, 1L, 7L, 2L, 7L, 2L, 8L, 2L, 9L, 2L, 10L), ncol=2)
net <- new.incgraph.network(links=mat)

# Calculate the initial orbit counts using orca
orb.counts <- calculate.orbit.counts(net)

# Modify an edge and calculate the differences in orbit counts
flip(net, 5, 10) # add (5,10)
delta1 <- calculate.delta(net, 5, 10)

# Modify another edge
flip(net, 6, 10) # add (6, 10)
delta2 <- calculate.delta(net, 6, 10)

# And another
flip(net, 1, 5) # remove (1, 5)
delta3 <- calculate.delta(net, 1, 5)

# Verify that the new orbit counts equals the old orbit counts plus the delta counts
new.orb.counts.incremental <- orb.counts +
  delta1$add - delta1$rem +
  delta2$add - delta2$rem +
  delta3$add - delta3$rem
new.orb.counts <- calculate.orbit.counts(net)
all(new.orb.counts.incremental == new.orb.counts) # TRUE

## Additional helper functions
# Transform the network to a matrix
network.as.matrix(net)
# Get all neighbours of a node
get.neighbours(net, 1)
# Does the network contain a specific interaction?
contains(net, 5, 10)
contains(net, 7, 10)
# Reinitialise to an empty network
reset(net)
network.as.matrix(net)
network.as.matrix  

**Description**

network.as.matrix returns the network as a matrix

**Usage**

network.as.matrix(network)

**Arguments**

- network: An instance of the incgraph.network class

**See Also**

See new.incgraph.network for examples and usage.

new.incgraph.network  

**Description**

new.incgraph.network creates a new IncGraph object containing either an empty network or a network initialised from a given matrix.

**Usage**

- new.incgraph.network(amnt.nodes, links=NULL)
- new.incgraph.network(amnt.nodes=NULL, links)
- new.incgraph.network(amnt.nodes, links)

**Arguments**

- amnt.nodes: The number of nodes in the network
- links: A matrix with 2 columns and N rows, 1 row for each edge to be loaded in the network

**Details**

This creates a new instance of the incgraph.network class. At least one of the parameters (amnt.nodes or links) needs to be passed to this function. Please note that this is a stateful object.
**Value**

An instance of the `incgraph.network` class

**See Also**

`incgraph`, `calculate.orbit.counts`, `calculate.delta`

**Examples**

```r
# Create a new (empty) network with 4 nodes
net <- new.incgraph.network(amnt.nodes = 4)

# Create a new network with 4 nodes and some edges
net <- new.incgraph.network(links = matrix(c(1, 2, 2, 3, 1, 4), ncol=2))

# Create a new network with 10 nodes and some edges
net <- new.incgraph.network(amnt.nodes = 10, links = matrix(c(1, 2, 2, 3, 1, 4), ncol=2))

# Create a more complex network from a matrix
mat <- matrix(c(1, 2,
               1, 3,
               1, 4,
               1, 5,
               1, 6,
               1, 7,
               2, 7,
               2, 8,
               2, 9,
               2, 10), ncol=2)
net <- new.incgraph.network(links=mat)
# Calculate the initial orbit counts using orca
orb.counts <- calculate.orbit.counts(net)
# Modify an edge and calculate the differences in orbit counts
flip(net, 5, 10) # add (5,10)
delta1 <- calculate.delta(net, 5, 10)
# Modify another edge
flip(net, 6, 10) # add (6, 10)
delta2 <- calculate.delta(net, 6, 10)
# And another
flip(net, 1, 5) # remove (1, 5)
delta3 <- calculate.delta(net, 1, 5)
# Verify that the new orbit counts equals the old orbit counts plus the delta counts
new.orb.counts.incremental <- orb.counts +
  delta1$add - delta1$rem +
  delta2$add - delta2$rem +
  delta3$add - delta3$rem
new.orb.counts <- calculate.orbit.counts(net)
all(new.orb.counts.incremental == new.orb.counts) # TRUE

## Additional helper functions
# Transform the network to a matrix
network.as.matrix(net)
```
# Get all neighbours of a node
get.neighbours(net, 1)

# Does the network contain a specific interaction?
contains(net, 5, 10)
contains(net, 7, 10)

# Reinitialise to an empty network
reset(net)
network.as.matrix(net)

---

**orca.halfdelta**  **Modify edge**

**Description**
orca.halfdelta calculates the orca counts for a network that has just been changed.

**Usage**
orca.halfdelta(network, i, j)

**Arguments**
- **network**: An instance of the incgraph.network class
- **i**: A node in network
- **j**: A node in network

---

**reset**  **Reset network**

**Description**
reset resets all the data structures so that all edges are removed from the network.

**Usage**
reset(network)

**Arguments**
- **network**: An instance of the incgraph.network class

**See Also**
See new.incgraph.network for examples and usage.
set.network

set.network | Set a given network to contain the given links

Description

set.network sets a given network to contain the given links.

Usage

set.network(network, links)

Arguments

network | An instance of the incgraph.network class
links | A matrix with 2 columns and N rows, 1 row for each edge to be loaded in the network

Details

This first resets the network and adds all given links. For minor changes to the network, the usage of flip is recommended.

See Also

See new.incgraph.network for examples and usage.
Index

calculate.delta, 2, 3, 6, 9
calculate.orbit.counts, 3, 6, 9
contains, 4
count5, 3

flip, 4, 11

generate.barabasialbert
  (generate.dynamic.network), 5
generate.dynamic.network, 5
generate.erdosrenyi
  (generate.dynamic.network), 5
generate.geometric
  (generate.dynamic.network), 5
get.neighbours, 6

incgraph, 6, 9
incgraph-package (incgraph), 6

network.as.matrix, 8
new.incgraph.network, 3, 4, 6, 8, 8, 10, 11

orca.halfdelta, 10
reset, 10

set.network, 11