Package 'lpbrim'

July 18, 2015

Version 1.0.0

Date 2015-07-17

Title LP-BRIM Bipartite Modularity

Author Timothee Poisot <tim@poisotlab.io>, Daniel B Stouffer <daniel.stouffer@canterbury.ac.nz>

Maintainer Timothee Poisot <tim@poisotlab.io>

Depends R (>= 2.12.0)

Imports Matrix, plyr, RColorBrewer

Suggests doMC

Description Optimization of bipartite modularity using LP-BRIM (Label propagation followed by Bipartite Recursively Induced Modularity).

License BSD_2_clause + file LICENSE

URL http://poisotlab.io/software

NeedsCompilation no

Repository CRAN

Date/Publication 2015-07-18 17:00:13

R topics documented:

bBRIM *BRIM*

Description

Performs the BRIM algorithm

Usage

```
bBRIM(x)
```

Arguments

x	Bipartite adjacency matrix

bLP *Bipartite label propagation (LP)*

Description

Performs LP

Usage

```
bLP(x, as.adjacency = TRUE)
```

Arguments

x	Bipartite network
as.adjacency	TRUE if x is a matrix, FALSE if it its an object with three columns

findModules *Find modules*

Description

This function takes a matrix, and performs a search for the best partition in modules.

Usage

```
findModules(M, iter = 50, sparse = TRUE, ...)
```

Arguments

M	An adjacency matrix
iter	Number of optimization runs to do
sparse	Whether the matrix should be made sparse
...	Other arguments

Examples

```
M <- matrix(rbinom(100, 1, 0.3), ncol=10)
M <- M[rowSums(M)>0, colSums(M)>0]
mod <- findModules(M, iter=2, sparse=FALSE)
```

---

getmodules *Get modules*

Description

This function takes a mod object, and returns a list of matrices, each corresponding to a single module

Usage

```
getmodules(mod)
```

Arguments

mod	The output of findModules

---

mostFrequent *Most frequent value*

Description

Returns the most frequent value in a vector

Usage

```
mostFrequent(vec, w = NA)
```

Arguments

vec	The vector of labels
w	The weights of each elements (currently not used)

netToAdjacency *Edge list to adjacency matrix*

Description

From a three column matrix (from, to, value) to an adjacency matrix

Usage

```
netToAdjacency(net)
```

Arguments

net	Three-column matrix

plotMatrixModules *Plot the modules as a matrix*

Description

Plot the modules as a network

Usage

```
plotMatrixModules(mod, mode = "blocks")
```

Arguments

mod	The output of findModules
mode	The type of plot, either blocks (different modules are colored), frames (modules are outlined), or both

plotModules *Plot the modules as a network*

Description

Plot the modules as a network

Usage

```
plotModules(mod)
```

Arguments

mod	The output of findModules

Qbip *Barber's modularity*

Description

Returns Q

Usage

```
Qbip(x, s)
```

Arguments

x	Bipartite adjacency matrix
s	Community partition

spread *Spread*

Description

Transforms values in v from m to M

Usage

```
spread(v, m = 0, M = 1)
```

Arguments

v	Values to transform
m	New min
M	New max

Index