Package 'MCL'

January 20, 2025

Type Package

Title Markov Cluster Algorithm

Version 1.0

Date 2015-03-11

Author Martin L. Jäger

Maintainer Ronja Foraita <foraita@bips.uni-bremen.de>

Description Contains the Markov cluster algorithm (MCL) for identifying clusters in networks and graphs. The algorithm simulates random walks on a (n x n) matrix as the adjacency matrix of a graph. It alternates an expansion step and an inflation step until an equilibrium state is reached.

License GPL (>= 2)

Depends R (>= 2.10.0)

Imports expm

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-11 09:56:32

Contents

MCL-package *Markov Cluster Algorithm*

Description

Contains the Markov cluster algorithm (MCL) by van Dongen (2000) for identifying clusters in networks and graphs. The algorithm simulates random walks on a (n x n) matrix as the adjacency matrix of a graph. It alternates an expansion step and an inflation step until an equilibrium state is reached.

Details

Package:	MCL
Type:	Package
Version:	1.0
Date:	2015-03-10
License:	GPL-2 | GPL-3 [expanded from: GPL (= 2)]

The Markov Cluster Algorithm (MCL) is a method to identify clusters in undirected network graphs. It is suitable for high-dimensional data (e.g. gene expression data).

The original MCL uses the adjacency matrix of a graph (propsed by van Dongen (2000)). The function `mcl` in this package allows in addition the input of a (n x n) matrix.

Note

We thank Moritz Hanke for his help in realizing this package.

Author(s)

Martin L. Jäger

Maintainer: Ronja Foraita <foraita@bips.uni-bremen.de>
Leibniz Institute for Prevention Research and Epidemiology (BIPS)

References

van Dongen, S.M. (2000) Graph Clustering by Flow Simulation. Ph.D. thesis, Universtiy of Utrecht. Utrecht University Repository: http://dspace.library.uu.nl/handle/1874/848

Examples

```
### Load adjacency matrix
adjacency <- matrix(c(0,1,1,1,0,0,0,0,0,1,0,1,1,1,0,0,0,0,0,1,1,
             0,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,1,1,0,
             0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,
```

```
              0,0,0,0,0,0,0,0,0,0,0,0), byrow=TRUE, nrow=9)

### Run MCL
mcl(x = adjacency, addLoops = TRUE )
```

---

mcl *Markov Cluster Algorithm*

Description

Perform the Markov Cluster Algorithm on an adjacency or (n x n) matrix.

Usage

```
mcl(x, addLoops = NULL, expansion = 2, inflation = 2, allow1 = FALSE,
       max.iter = 100, ESM = FALSE)
```

Arguments

x	an adjacency or (n x n) matrix
addLoops	logical; if TRUE, self-loops with weight 1 are added to each vertex of x (see Details).
expansion	numeric value > 1 for the expansion parameter
inflation	numeric value > 0 for the inflation power coefficient
allow1	logical; if TRUE, vertices are allowed to form their own cluster. If FALSE, clusters of size 1 are interpreted as background noise and grouped in one cluster.
max.iter	an interger, the maximum number of iterations for the MCL
ESM	logical whether the equilibrium state matrix should be returned (default is FALSE)

Details

The adjacency or correlation matrix x is clustered by the Markov Cluster algorithm. The algorihtm is controlled by the expansion parameter and the inflation power coefficient (for further details, see reference below). Adding self-loops is necessary, if either x contains at least one vertex of degree 0 or x represents a directed, non-bipartite graph adjacency matrix (i.e. the upper or lower matrix of x contains only zeros).

Value

K	the number of clusters
n.iterations	the number of iterations
Cluster	a vector of integers indicating the cluster to which each vertex is allocated
Equilibrium.state.matrix	a matrix; rows contain clusters

Note

If an error occurs, `mcl` returns the number of the last iteration. If an error occurs at iteration 1, there might be a problem with the matrix `x`. If an error occurs at iteration `max.iter`, `x` could not be transformed into an equilibrium state matrix.

Author(s)

Martin L. Jäger

References

van Dongen, S.M. (2000) Graph Clustering by Flow Simulation. Ph.D. thesis, Universtiy of Utrecht. Utrecht University Repository: http://dspace.library.uu.nl/handle/1874/848

Examples

```
### Generate adjacency matrix of undirected graph
adjacency <- matrix(c(0,1,1,1,0,0,0,0,0,1,0,1,1,1,0,0,0,0,1,1,
                      0,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,0,0,
                      0,1,1,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,
                      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
                      byrow=TRUE, nrow=9)

### Plot graph (requires package igraph)
# library(igraph)
# gu <- graph.adjacency( adjacency, mode="undirected" )
# plot( gu )

### Run MCL
mcl(x = adjacency, addLoops=TRUE, ESM = TRUE)

### Allow clusters of size 1
mcl(x = adjacency, addLoops = TRUE, allow1 = TRUE)

### Error: Small inflation coefficient prevents that an
###        equilibrium state matrix is reached within 100 iterations
mcl(x = adjacency, addLoops=TRUE, inflation = 1.01, max.iter = 100)


### Generate adjacency matrix of directed graph
dgr <- matrix(0,nrow = 10,ncol = 10)
dgr[2:3,1] <- 1; dgr[3:4,2] <- 1; dgr[5:6,4] <- 1
dgr[6:7,5] <- 1; dgr[8:9,7] <- 1; dgr[10,8:9] <- 1

### Plot graph (requires package igraph)
# library( igraph )
# gd <- graph.adjacency( dgr )
# plot( gd )

### Directed graphs require self-loops!
mcl(x = dgr, addLoops = TRUE)
```

Index