Package ‘bingat’

January 15, 2016

Type Package
Title Binary Graph Analysis Tools
Version 1.2.2
Date 2016-01-14
Author Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon
Maintainer Berkley Shands <rpackages@biorankings.com>
Depends R (>= 2.14.0)
Imports matrixStats, network, gplots, genalg, doParallel, foreach, parallel
Description Tools to analyze binary graph objects.
License Apache License (= 2.0)
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2016-01-15 22:34:06

R topics documented:

bingat-package ......................................................... 2
braingraphs ........................................................... 3
calcDistance ........................................................... 3
estGStar ................................................................. 4
estLogLik ................................................................. 5
estMLE ................................................................. 6
estTau ................................................................. 7
gaConsensus ......................................................... 7
getGibbsMixture ...................................................... 8
getLoglikeMixture ................................................... 9
genNumEdges ....................................................... 10
genNumNodes ....................................................... 11
Tools for analyzing binary graphs, including calculating the MLE of a set of binary graphs, comparing MLE of sets of graphs, regression analysis on sets of graphs, using genetic algorithm to identify nodes and edges separating sets of graphs, and generating random binary graphs sampled from the Gibbs distribution.

The following are the types of binary graphs that are accepted:

1. adjMatrix: An entire binary adjacency matrix as a single vector
2. adjMatrixLT: The upper or lower triangle of a binary adjacency matrix as a single vector
3. diag: The diagonal vector on a binary adjacency matrix

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**braingraphs**

**Brain Graph Data Set**

**Description**
A data set containing 38 brain scans each with 20 total nodes.

**Usage**
data(braingraphs)

**Format**
The format is a data frame of 400 rows by 38 columns, with each column being a separate subject and each row being a different edge between 2 nodes. Each column is a 20 by 20 matrix of brain connections transformed into a vector. A value of 1 indicates that subject had a connection at that edge.

---

**calcDistance**

**Calculate the Distance Between Vectors**

**Description**
This function calculates the distance between two vectors.

**Usage**
calcDistance(x, y, type = "", method = "hamming")

**Arguments**
- `x, y` Vectors of the same length that contain 1’s and 0’s.
- `type` The type of graph being used (adjmatrix or adjmatrixlt). See 'Details'
- `method` The distance metric to use, currently only "hamming" is supported.

**Details**
If the type = "adjMatrix" is used, the value will be divided by 2 to account for duplicate comparisons. Otherwise the type does not affect the output.

**Value**
A single number indicating the distance between the two input vectors.
**Summary**

This function estimates the g-star graph for a given set of graphs.

**Usage**

```r
estGStar(data)
```

**Arguments**

- `data` A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).

**Value**

A single vector that is the gstar is returned.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

braingstar <- estGStar(braingraphs)
braingstar[1:25]
```
estLogLik

Estimate the Log Likelihood Value

Description
This function estimates log likelihood value for a given graph.

Usage
estLogLik(data, type, g, tau)

Arguments
- data: A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- type: The type of graph being used (adjmatrix or adjmatrixlt).
- g: A single columned data frame to estimate the likelihood for.
- tau: A single value used in estimating the likelihood.

Value
The log-likelihood value of the data.

Author(s)
Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

Examples
data(braingraphs)
braingstar <- estGStar(braingraphs)
braintau <- estTau(braingraphs, "adjMatrix", braingstar)
brainll <- estLogLik(braingraphs, "adjMatrix", braingstar, braintau)
brainll
**Estimate the MLE Parameters**

**Description**

This function estimates the MLE parameters $g$-star and $\tau$ for a given set of graphs.

**Usage**

```r
estMLE(data, type)
```

**Arguments**

- `data` A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- `type` The type of graph being used (adjmatrix or adjmatrixLt).

**Details**

Essentially this function calls both `estGStar` and `estTau` and returns the results.

**Value**

A list containing $g$-star and $\tau$ named `gstar` and `tau` respectively.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

brainmle <- estMLE(braingraphs, "adjMatrix")
brainmle
```
**estTau**

*Estimate Tau*

**Description**

This function estimates tau for a given set of graphs.

**Usage**

```r
estTau(data, type, gstar)
```

**Arguments**

- `data` A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- `type` The type of graph being used (adjmatrix or adjmatrixlt).
- `gstar` A single columned data frame to be used as the g-star of the data set.

**Value**

The tau value for the data based on g star.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

braingstar <- estGStar(braingraphs)
braintau <- estTau(braingraphs, "adjMatrix", braingstar)
braintau
```

**gaConsensus**

*Genetic Algorithm Consensus Solution*

**Description**

Runs a Genetic Algorithm to find nodes or edges that separate groups

**Usage**

```r
gaConsensus(data, groups, iters = 10, nRuns = 1, popSize = 200, method = "manhattan", parallel = FALSE, cores = 3)
```
Arguments

data          A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
groups        A vector of 0/1s used to denote the group membership of columns in the data.
iters         The number of iterations (ideally, 1000 or more).
nRuns         The number of times to repeat the GA to get a consensus.
popSize       The population size.
method        The distance metric to use.
parallel      TRUE or FALSE depending on whether the analysis will be parallelized for speed.
cores         The number of cores to use for parallelization. Ignored if parallel = FALSE.

Value

A list containing all the solutions from the last run(s) of the GA and their correlation values

Author(s)

Berkley Shands, Elena Deych, William D. Shannon

See Also

rbga.bin.

Examples

data(braingraphs)

grps <- c(rep(0, 19), rep(1, 19))
iters <- 1 ### This is set low for speed
nRuns <- 1 ### This is set low for speed

consensus <- gaConsensus(braingraphs, grps, iters, nRuns)
consensus$corr[1:5]
consensus$solutions[1:2,]

getGibbsMixture        Group Splitter

Description

This function splits the data into groups based on the Gibbs criteria.

Usage

getGibbsMixture(data, type, desiredGroups, maxIter = 50, digits = 3)
**getLoglikeMixture**

**Arguments**

- `data` A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- `type` The type of graph being used (adjmatrix or adjmatrixlt).
- `desiredGroups` The number of groups to test for.
- `maxIter` The maximum number of iterations to run searching for an optimal split.
- `digits` The number of digits to round internal values to when checking the stop criteria.

**Details**

Generally this function is not used by itself but in conjunction with getLoglikeMixture.

**Value**

A list that contains information about the group splits. The list contains the final weights, gstars and taus for every group, a boolean indicating convergence, the number of iterations it took, and the group for each graph.

**Author(s)**

Terrence Brooks, Berkly Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

braingm <- getGibbsMixture(braingraphs, "adjMatrix", 5)
```

---

**getLoglikeMixture**  
*Group Finder*

**Description**

This function takes group splits and determines the likelihood of those groups.

**Usage**

```r
geloglikemixture(data, mixture, numConst)
```

**Arguments**

- `data` A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- `mixture` The output of the getGibbsMixture function.
- `numConst` The numeric constant to multiply the loglikelihood by.
**Value**

A list containing the BIC criteria and the log likelihood named bic and ll respectively.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

braingm <- getGibbsMixture(braingraphs, "adjMatrix", 5)
brainlm <- getLoglikeMixture(braingraphs, braingm)
brainlm

### By running the loglik mixture over several groups you can find which is the optimal CC not run:
mixtures <- NULL
for(i in 1:5){
tempgm <- getGibbsMixture(braingraphs, "adjMatrix", i)
mixtures[i] <- getLoglikeMixture(braingraphs, tempgm)$bic
}
bestgroupnum <- which(min(mixtures) == mixtures)
bestgroupnum

### End(Not run)
```

---

**getNumEdges**

*Get the Number of Edges in a Graph*

**Description**

This function will return the number of edges for a given graph.

**Usage**

```r
getNumEdges(nodes, type)
```

**Arguments**

- **nodes**  The number of individual nodes in a given graph.
- **type**  The type of graph being used (adjmatrix or adjmatrixlt).

**Value**

The number of edges between individual nodes in the given graph.
getNumNodes

Author(s)
Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

Examples

```
data(braingraphs)
brainnodes <- getNumNodes(braingraphs, "adjMatrix")
brainedges <- getNumEdges(braingraphs, "adjMatrix")

brainnodes
```

---

**getNumNodes**  
*Get the Number of Nodes in a Graph*

Description

This function will return the number of nodes for a given graph.

Usage

```
getNumNodes(data, type)
```

Arguments

- **data**: A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- **type**: The type of graph being used (adjmatrix or adjmatrixLt).

Value

The number of individual nodes in the given graph.

Author(s)
Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

Examples

```
data(braingraphs)

brainnodes <- getNumNodes(braingraphs, "adjMatrix")
brainnodes
```
glrtpvalue  

**GLRT Regression Results**

**Description**

This function returns the p-value of the significance of b1 in the regression model.

**Usage**

```
glrtpvalue(data, type, groups, numPerms = 10, parallel = FALSE, cores = 3)
```

**Arguments**

- **data**
  A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).

- **type**
  The type of graph being used (adjmatrix or adjmatrixlt).

- **groups**
  A vector of 0/1s used to denote the group membership of columns in the data.

- **numPerms**
  The number of permutations (ideally, 1000 or more).

- **parallel**
  TRUE or FALSE depending on whether the analysis will be parallelized for speed.

- **cores**
  The number of cores to use for parallelization. Ignored if parallel = FALSE.

**Value**

A list containing the results from **glmNreg** and **GLRT.reg** as well as the p-value.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

grops <- sample(0:1, ncol(braingraphs), TRUE)
umPermutations <- 1 ### This is set low for speed

glrt <- glrtpvalue(braingraphs, "adjMatrix", grps, numPermutations)
glrt
```
**graphNetworkPlot**

---

**graphNetworkPlot**  *Graph Network Plots*

---

**Description**

This function plots the connections between nodes in a single subject.

**Usage**

```r
graphNetworkPlot(data, type, main = "Network Plot", labels, groupCounts, groupLabels)
```

**Arguments**

- `data`  A vector of a single graph.
- `type`  The type of graph being used (adjmatrix or adjmatrixLt).
- `main`  The title for the plot.
- `labels`  A vector which contains the names for each node.
- `groupCounts`  A vector which contains the number of nodes in each group of nodes.
- `groupLabels`  A vector which contains the names for each group of nodes.

**Value**

A plot displaying the connections between the nodes.

**Author(s)**

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

**Examples**

```r
data(braingraphs)

main <- "Brain Connections"
gc <- c(5, 5, 4, 6)
gl <- c("Grp1", "Grp2", "Grp3", "Grp4")

graphNetworkPlot(braingraphs[,1], "adjMatrix", main, groupCounts=gc, groupLabels=gl)
```
lrtPvalue  

Likelihood Ratio Test

Description

This function returns the p-value of the significance between two groups.

Usage

lrtPvalue(data, type, groups, numPerms = 10, parallel = FALSE, cores = 3)

Arguments

data A data frame in which the columns contain a 0/1 value for each node in the rows.
type The type of graph being used (adjmatrix or adjmatrixLt).
groups A vector of 0/1s used to denote the group membership of columns in the data.
numPerms The number of permutations (ideally, 1000 or more).
parallel TRUE or FALSE depending on whether the analysis will be parallelized for speed.
cores The number of cores to use for parallelization. Ignored if parallel = FALSE.

Value

The p-value for the difference between the two groups being tested.

Author(s)

Berkley Shands, Elena Deych, William D. Shannon

Examples

data(braingraphs)

grps <- sample(0:1, ncol(braingraphs), TRUE)
numPermutations <- 1 ### This is set low for speed

lrt <- lrtPvalue(braingraphs, "adjMatrix", grps, numPermutations)
lrt
pairedPvalue

P-Value for Paired Data Results

Description
This function returns the p-value of the significance of the difference in g-star values for paired data.

Usage
pairedPvalue(data, type, groups, numPerms = 10, parallel = FALSE, cores = 3)

Arguments
- data: A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
- type: The type of graph being used (adjmatrix or adjmatrixlt).
- groups: A vector of 0/1s used to denote the group membership of columns in the data.
- numPerms: The number of permutations (ideally, 1000 or more).
- parallel: TRUE or FALSE depending on whether the analysis will be parallelized for speed.
- cores: The number of cores to use for parallelization. Ignored if parallel = FALSE.

Value
A p-value.

Author(s)
Berkley Shands, Elena Deych, William D. Shannon

Examples
```r
data(braingraphs)

grs <- c(rep(0, 19), rep(1, 19))
numPermutations <- 1 # This is set low for speed

pval <- pairedPvalue(braingraphs, "adjMatrix", grs, numPermutations)
pval```
**plotHeatmap**  
*Plot Heat Map*

**Description**
This function plots the connections between nodes in a single subject as a heat map.

**Usage**
plotHeatmap(data, type, names, ...)

**Arguments**
- **data**: A vector of a single graph.
- **type**: The type of graph being used (adjmatrix or adjmatrixlt).
- **names**: A vector of names for labeling the nodes on the plot.
- **...**: Arguments to be passed to the plot method.

**Value**
A plot displaying the connections between the nodes as a heat map.

**Author(s)**
Berkley Shands, Elena Deych, William D. Shannon

**Examples**
data(braingraphs)

braingstar <- estGstar(braingraphs)
plotHeatmap(braingstar, "adjMatrix")

---

**plotMDS**  
*Plot MDS*

**Description**
This function plots all the data on an MDS plot.

**Usage**
plotMDS(data, groups, estGstar = TRUE, paired = FALSE, returnCoords = FALSE, ...)

---
rgibbs

Generate Random Data

Description

Generate random data sampled from the Gibbs distribution.

Usage

rgibbs(gstar, tau, type, numGraphs = 1)

Arguments

- **gstar**: G star vector.
- **tau**: A single value that affects the dispersion of the generated data.
- **type**: The type of graph being used (adjmatrix or adjmatrixlt).
- **numGraphs**: The number of graphs to generate.
Value

A data frame containing all the graphs generated.

Author(s)

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon

Examples

data(braingraphs)

braingstar <- estGStar(braingraphs)
braintau <- estTau(braingraphs, "adjMatrix", braingstar)
randombraingraphs <- rGibbs(braingstar, braintau, "adjMatrix", 3)
randombraingraphs[1:5,]

testGoF

Test the Goodness of Fit

Description

This function tests the goodness of fit for given a set of graphs.

Usage

testGoF(data, type, numSims = 10, plot = TRUE, main)

Arguments

data A data frame in which the columns (subjects) contain a 0/1 value for row (Node or Edge).
type The type of graph being used (adjmatrix or adjmatrixlt).
numSims Number of simulations for Monte Carlo estimation of p-value (ideally, 1000 or more). Ignored if Chi-Square method is used.
plot A boolean to create a plot of the results or not.
main A title for the plot.

Value

A list containing information about the goodness of fit and potentially a plot. The list contains the Pearson statistics, degrees of freedom, and p-value, the G statistics and p-value, the Chi Squared statistics and p-value and finally the table with the observed and expected counts.

Author(s)

Terrence Brooks, Berkley Shands, Skye Buckner-Petty, Patricio S. La Rosa, Elena Deych, William D. Shannon
testGoF

Examples

data(braingraphs)

numSims <- 1 ### This is set low for speed
braingof <- testGoF(braingraphs, "adjMatrix", numSims)
Index

*Topic **datasets**
  braingraphs, 3
*Topic **package**
  bingat-package, 2

bingat (bingat-package), 2
bingat-package, 2
braingraphs, 3

calcDistance, 3

estGStar, 4
estLogLik, 5
estMLE, 6
estTau, 7

gaConsensus, 7
getGibbsMixture, 8
getLoglikeMixture, 9
getNumEdges, 10
getNumNodes, 11
glrtpvalue, 12
graphNetworkPlot, 13

lrtPvalue, 14

pairedPvalue, 15
plotHeatmap, 16
plotMDS, 16

rbga.bin, 8
rGibbs, 17

testGoF, 18