

# Package ‘bingat’

February 19, 2015

**Type** Package

**Title** Binary Graph Analysis Tools

**Version** 1.1

**Date** 2014-06-20

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

**Maintainer** Berkley Shands <bshands@dom.wustl.edu>

**Depends** R (>= 2.13.0)

**Imports** matrixStats, network

**Description** This package provides tools to analyze binary graph objects.

**License** Apache License (== 2.0)

**LazyData** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-06-24 23:39:45

## R topics documented:

bingat-package	2
braingraphs	2
calcDistance	3
estGStar	4
estLogLik	4
estMLE	5
estTau	6
getGibbsMixture	7
getLoglikeMixture	8
getNumEdges	9
getNumNodes	9
glrtPvalue	10
graphNetworkPlot	11
rGibbs	12
testGoF	13

---

bingat-package      *Binary Graph Analysis Tools*

---

### Description

This package provides tools for: Calculating the MLE of a set of binary graphs, Generating random binary graphs sampled from the Gibb's distribution, and Doing regression analysis of a set of binary graphs.

### Details

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

### Author(s)

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 transformed into a vector. A value of 1 indicates that subject had a connection at that edge.

---

calcDistance	<i>Calculate the Distance Between Vectors</i>
--------------	---

---

### 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 compared. 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.

### Author(s)

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

### Examples

```
data(braingraphs)

dist <- calcDistance(braingraphs[,1], braingraphs[,2], "adjMatrix")
dist
```

---

`estGStar`*Estimate G-Star*

---

**Description**

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

**Usage**

```
estGStar(data, cutoff = 0.5)
```

**Arguments**

`data` A data frame in which the columns contain a 0/1 value for each node in the rows.  
`cutoff` The percentage of 1's a node must have to be kept in the gstar. Default is 0.50.

**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**

```
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 contain a 0/1 value for each node in the rows.
type	The type of graph being used.
g	A single columned data frame to estimate the likelihood for.
tau	A single value used in estimating the likelihood.

**Value**

The likelihood value of the input graph g.

**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
```

---

estMLE *Estimate the MLE Parameters*

---

**Description**

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

**Usage**

```
estMLE(data, type, cutoff = .5)
```

**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.
cutoff	The percentage of 1's a node must have to be kept in the gstar. Default is 0.50.

**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**

```
data(braingraphs)

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

---

 estTau

*Estimate Tau*


---

**Description**

This function estimates tau for a given set of graphs.

**Usage**

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

**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.
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.

**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)
braintau
```

---

getGibbsMixture	<i>Group Splitter</i>
-----------------	-----------------------

---

### Description

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

### Usage

```
getGibbsMixture(data, type, desiredGroups, maxIter = 50, digits = 3, cutoff = .5)
```

### 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.
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.
cutoff	The percentage of 1's a node must have to be kept in the gstar. Default is 0.50.

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

### Examples

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

---

getLoglikeMixture      *Group Finder*

---

### Description

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

### Usage

```
getLoglikeMixture(data, mixture)
```

### Arguments

data	A data frame in which the columns contain a 0/1 value for each node in the rows.
mixture	The output of the getGibbsMixture function.

### 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

```
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
## 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 of graph.

**Usage**

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

**Arguments**

<code>nodes</code>	The number of individual nodes in a given graph.
<code>type</code>	The type of graph being used.

**Value**

The number of edges between 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")
brainedges <- getNumEdges(brainnodes, "adjMatrix")
brainedges
```

---

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

---

**Description**

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

**Usage**

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

**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.

**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, PBmethod = FALSE, bootstraps = 10, cutoff = .5)
```

**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.  
`groups` A vector of 0/1s used to denote the starting group splits.  
`PBmethod` A boolean that determines if a parametric method of calculating the p-value should be used or not.  
`bootstraps` The number of bootstraps.  
`cutoff` The percentage of 1's a node must have to be kept in the gstar. Default is 0.50.

**Details**

The non parametric method in PBmethod allows for bootstrapping to sample the data from each individual group with replacement and calculates the GLRT for each bootstrapped dataset. Then the number that are greater than original GLRT result are used to calculate the p-value.

The parametric method in PBmethod estimates a central graph and tau of the entire data ignoring groups and then randomly generates data from the Gibbs distribution. It assigns data to groups based on the given covs and calculates the GLRT value and the number that are greater than original GLRT result are used to calculate the p-value.

**Value**

A list containing the results from glm.reg 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**

```
data(braingraphs)

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

btrue <- glrtPvalue(braingraphs, "adjMatrix", grps, TRUE, numBootStraps)
btrue
```

---

graphNetworkPlot

*Graph Network Plots*


---

**Description**

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

**Usage**

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

**Arguments**

data	A vector of a single graph.
type	The type of graph being used.
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**

```
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)
```

---

rGibbs

*Generate Random Data*

---

**Description**

Generate random data sampled from the Gibbs distribution.

**Usage**

```
rGibbs(gstar, tau, type, numGraphs = 1)
```

**Arguments**

gstar	A vector of a single graph.
tau	A single value that affects the dispersion of the generated data.
type	The type of graph being used.
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, plot = TRUE, cutoff = .5, main)
```

**Arguments**

<code>data</code>	A data frame in which the columns contain a 0/1 value for each node in the rows.
<code>type</code>	The type of graph being used.
<code>plot</code>	A boolean to create a plot of the results or not.
<code>cutoff</code>	The percentage of 1's a node must have to be kept in the gstar. Default is 0.50.
<code>main</code>	The title for the plot if applicable.

**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

**Examples**

```
data(braingraphs)

gof <- testGoF(braingraphs, "adjMatrix")
```

# Index

\*Topic **datasets**

braingraphs, [2](#)

\*Topic **package**

bingat-package, [2](#)

bingat (bingat-package), [2](#)

bingat-package, [2](#)

braingraphs, [2](#)

calcDistance, [3](#)

estGStar, [4](#)

estLogLik, [4](#)

estMLE, [5](#)

estTau, [6](#)

getGibbsMixture, [7](#)

getLoglikeMixture, [8](#)

getNumEdges, [9](#)

getNumNodes, [9](#)

glrtPvalue, [10](#)

graphNetworkPlot, [11](#)

rGibbs, [12](#)

testGoF, [13](#)