

# Package ‘statGraph’

April 16, 2024

**Type** Package

**Title** Statistical Methods for Graphs

**Version** 1.0.0

**Maintainer** Andre Fujita <andrefujita@usp.br>

**Depends** R (>= 4.3.0), stats, graphics

**Imports** igraph, MASS, rARPACK, cluster, foreach, parallel, doParallel, methods

**Description** Contains statistical methods to analyze graphs, such as graph parameter estimation, model selection based on the Graph Information Criterion, statistical tests to discriminate two or more populations of graphs, correlation between graphs, and clustering of graphs.

References: Takahashi et al. (2012) <[doi:10.1371/journal.pone.0049949](https://doi.org/10.1371/journal.pone.0049949)>, Fujita et al. (2017) <[doi:10.3389/fnins.2017.00066](https://doi.org/10.3389/fnins.2017.00066)>, Fujita et al. (2017) <[doi:10.1016/j.csda.2016.11.016](https://doi.org/10.1016/j.csda.2016.11.016)>, Fujita et al. (2019) <[doi:10.1093/comnet/cnz028](https://doi.org/10.1093/comnet/cnz028)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyLoad** yes

**URL** <https://www.ime.usp.br/~fujita/software.html>

**Date** 2024-04-16

**NeedsCompilation** no

**RoxygenNote** 7.3.1

**Repository** CRAN

**Author** Grover E. Castro Guzman [aut],  
Diogo R. da Costa [aut],  
Taiane C. Ramos [aut],  
Suzana S. Santos [aut],  
Eduardo S. Lira [aut],  
Andre Fujita [aut, cre]

**Date/Publication** 2024-04-16 21:00:02 UTC

## R topics documented:

anogva . . . . .	2
GIC . . . . .	3
graph.acf . . . . .	5
graph.cem . . . . .	6
graph.cor.test . . . . .	7
graph.dist . . . . .	8
graph.entropy . . . . .	9
graph.hclust . . . . .	10
graph.kmeans . . . . .	11
graph.model.selection . . . . .	13
graph.mult.scaling . . . . .	14
graph.param.estimator . . . . .	16
graph.spectral.density . . . . .	18
graph.takahashi.test . . . . .	20
graph.tang.test . . . . .	21
sp.anogva . . . . .	23
<b>Index</b>	<b>25</b>

---

anogva	<i>ANOGVA Analysis Of Graph Variability</i>
--------	---

---

### Description

anogva statistically tests whether two or more sets of graphs are generated by the same random graph model. It is a generalization of the takahashi.test function.

### Usage

```
anogva(Graphs, labels, maxBoot = 1000, dist = "KL", ...)
```

### Arguments

Graphs	a list of undirected graphs. If each graph has the attribute eigenvalues containing its eigenvalues, such values will be used to compute their spectral density.
labels	an array of integers indicating the labels of each graph.
maxBoot	integer indicating the number of bootstrap resamplings (default 1000).
dist	string indicating if you want to use the "KL" (default), "JS", "L1" or "L2" distances. "KL" means Kullback-Leibler divergence. "JS" means Jensen-Shannon divergence.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

**Value**

A list with class "hstest" containing the following components:

```

statistic:    the statistic of the test.
p.value:      the p-value of the test.
method:       a string indicating the used method.
data.name:    a string with the data's name(s).

```

**References**

Fujita, A., Vidal, M. C. and Takahashi, D. Y. (2017) A Statistical Method to Distinguish Functional Brain Networks. *Front. Neurosci.*, *11*, 66. doi:10.3389/fnins.2017.00066.

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *PLoS ONE*, *7*, e49949. doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *Density Estimation*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *J. Am. Statist. Assoc.*, *21*, 65-66.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, *53*, 683-690. <http://www.jstor.org/stable/2345597>.

**Examples**

```

set.seed(1)
g1 <- g2 <- g3 <- list()
for (i in 1:20) {
  g1[[i]] <- igraph::sample_gnp(50, 0.50)
  g2[[i]] <- igraph::sample_gnp(50, 0.50)
  g3[[i]] <- igraph::sample_gnp(50, 0.52)
}
G <- c(g1, g2, g3)
label <- c(rep(1,20), rep(2,20), rep(3,20))
result <- anogva(G, label, maxBoot=50)
result

```

---

GIC

*Graph Information Criterion (GIC)*


---

**Description**

GIC returns the Kullback-Leibler divergence, L1 or L2 distance between an undirected graph and a given graph model using the exact or degree-based spectral densities.

**Usage**

```
GIC(Graph, model, p = NULL, dist = "KL", ...)
```

**Arguments**

Graph	the undirected graph (igraph object). If Graph has the attribute eigenvalues containing the eigenvalues of Graph, such values will be used to compute its spectral density.
model	<p>either a list, a string, or a function describing a graph model:</p> <p>A list that represents the spectral density of a model. It contains the components "x" and "y", which are numeric vectors of the same size. The "x" component contains the points at which the density was computed and the "y" component contains the observed density.</p> <p>A string that indicates one of the following models: "ER" (Erdos-Renyi random graph), "GRG" (geometric random graph), "KR" (k regular random graph), "WS" (Watts-Strogatz model), and "BA" (Barabási-Albert model). When the argument model is a string, the user must also provide the model parameter by using the argument p.</p> <p>A function that returns a graph (igraph object) generated by a graph model. It must contain two arguments: the first one corresponds to the graph size and the second to the parameter of the model. The model parameter will be provided by the argument p of the GIC function.</p>
p	<p>the model parameter. The user must provide a valid parameter if the argument model is a string or a function. For the predefined models ("ER", "GRG", "KR", "WS", and "BA"), the parameter the probability to connect a pair of vertices, for the "ER" model (Erdos-Renyi random graph);</p> <p>the radius used to construct the geometric graph in a unit square, for the "GRG" model (geometric random graph);</p> <p>the degree k of a regular graph, for the "KR" model (k regular random graph);</p> <p>the probability to reconnect a vertex, for the "WS" model (Watts-Strogatz model);</p> <p>and the scaling exponent, for the "BA" model (Barabási-Albert model).</p>
dist	string indicating if you want to use the "KL" (default), "L1" or "L2" distances. "KL" means Kullback-Leibler divergence.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

**Value**

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
data.name:	a string with the data's name(s).
value:	a real number corresponding to the Kullback-Leibler divergence, L1, or L2 distance between the observed graph and the graph model.

**References**

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *Density Estimation*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *J. Am. Statist. Assoc.*, \*21\*, 65-66.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

## Examples

```
set.seed(1)
G <- igraph::sample_gnp(n=50, p=0.5)

# Using a string to indicate the graph model
result1 <- GIC(G, "ER", 0.5)
result1

# Using a function to describe the graph model
# Erdos-Renyi graph
model <- function(n, p) {
  return (igraph::sample_gnp(n, p))
}
result2 <- GIC(G, model, 0.5)
result2
```

---

graph.acf

*Auto Correlation Function Estimation for Graphs*

---

## Description

The function `graph.acf` computes estimates of the autocorrelation function for graphs.

## Usage

```
graph.acf(Graphs, plot = TRUE)
```

## Arguments

Graphs	a list of undirected graphs. If each graph has the attribute <code>eigenvalues</code> containing its eigenvalues, such values will be used to compute their spectral density.
plot	logical. If TRUE (default) the <code>graph.acf</code> is plotted.

## Value

An object of class `acf`.

## References

Fujita, A., Takahashi, D. Y., Balardin, J. B., Vidal, M. C. and Sato, J. R. (2017) Correlation between graphs with an application to brain network analysis. *Computational Statistics & Data Analysis* \*109\*, 76-92.

**Examples**

```

set.seed(1)
G <- list()
p <- array(0, 100)
p[1:3] <- rnorm(3)
for (t in 4:100) {
  p[t] <- 0.5*p[t-3] + rnorm(1)
}
ma <- max(p)
mi <- min(p)
p <- (p - mi)/(ma-mi)
for (t in 1:100) {
  G[[t]] <- igraph::sample_gnp(100, p[t])
}
graph.acf(G, plot=TRUE)

```

---

graph.cem

*Clustering Expectation-Maximization for Graphs (graph.cem)*


---

**Description**

graph.cem clusters graphs following an expectation-maximization algorithm based on the Kullback-Leibler divergence between the spectral densities of the graph and of the random graph model.

**Usage**

```
graph.cem(Graphs, model, k, max_iter = 10, ...)
```

**Arguments**

Graphs	a list of undirected graphs. If each graph has the attribute eigenvalues containing its eigenvalues, such values will be used to compute their spectral density.
model	a string that indicates one of the following random graph models: "ER" (Erdos-Renyi random graph), "GRG" (geometric random graph), "KR" (k regular graph), "WS" (Watts-Strogatz model), and "BA" (Barabási-Albert model).
k	an integer specifying the number of clusters.
max_iter	the maximum number of expectation-maximization steps to execute.
...	Other relevant parameters for <a href="#">graph.param.estimator</a> .

**Value**

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
data.name:	a string with the data's name(s).

**cluster:** a vector of the same length of `g` containing the clusterization labels.

**parameters:** a vector containing the estimated parameters for the groups. It has the length equals to `k`.

## References

Celex, Gilles, and Gerard Govaert. "Gaussian parsimonious clustering models." *Pattern recognition* 28.5 (1995): 781-793.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

## Examples

```
set.seed(1)
g <- list()
for(i in 1:2){
  g[[i]] <- igraph::sample_gnp(n=10, p=0.5)
}
for(i in 3:4){
  g[[i]] <- igraph::sample_gnp(n=10, p=1)
}
res <- graph.cem(g, model="ER", k=2, max_iter=1)
res
```

---

graph.cor.test

*Test for Association / Correlation Between Paired Samples of Graphs*

---

## Description

`graph.cor.test` tests for association between paired samples of graphs, using Spearman's rho correlation coefficient.

## Usage

```
graph.cor.test(Graphs1, Graphs2)
```

## Arguments

**Graphs1** a list of undirected graphs. If each graph has the attribute `eigenvalues` containing its eigenvalues, such values will be used to compute their spectral density.

**Graphs2** a list of undirected graphs. If each graph has the attribute `eigenvalues` containing its eigenvalues, such values will be used to compute their spectral density.

**Value**

A list with class "htest" containing the following components:

statistic: the value of the test statistic.  
 p.value: the p-value of the test.  
 method: a string indicating the used method.  
 data.name: a string with the data's name(s).  
 estimates: the estimated measure of association 'rho'.

**References**

Fujita, A., Takahashi, D. Y., Balardin, J. B., Vidal, M. C. and Sato, J. R. (2017) Correlation between graphs with an application to brain network analysis. *Computational Statistics & Data Analysis* \*109\*, 76-92.

**Examples**

```
set.seed(1)
G1 <- G2 <- list()

p <- MASS::mvrnorm(50, mu=c(0,0), Sigma=matrix(c(1, 0.5, 0.5, 1), 2, 2))

ma <- max(p)
mi <- min(p)
p[,1] <- (p[,1] - mi)/(ma - mi)
p[,2] <- (p[,2] - mi)/(ma - mi)

for (i in 1:50) {
  G1[[i]] <- igraph::sample_gnp(50, p[i,1])
  G2[[i]] <- igraph::sample_gnp(50, p[i,2])
}
graph.cor.test(G1, G2)
```

---

graph.dist

*Distance matrix on a list of graphs.*

---

**Description**

Given a list of graphs, graph.dist builds a distance matrix according to the Jensen-Shannon divergence, L2 norm, or L1 norm between the spectral density of the graphs graphs.

**Usage**

```
graph.dist(Graphs, dist = "JS", ...)
```



**Arguments**

Graphs	a list of undirected graphs. If each graph has the attribute <code>eigenvalues</code> containing its eigenvalues, such values will be used to compute their spectral density.
dist	string indicating if you want to use the "JS" (default), "L1" or "L2" distances. "JS" means Jensen-Shannon divergence.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

**Value**

a distance matrix

**Examples**

```
set.seed(1)
g <- list()
for(i in 1:5){
  g[[i]] <- igraph::sample_gnp(n=50, p=0.1)
}
for(i in 6:10){
  g[[i]] <- igraph::sample_gnp(n=50, p=0.5)
}
for(i in 11:15){
  g[[i]] <- igraph::sample_gnp(n=50, p=0.9)
}
graph.dist(g, dist = "JS")
```

---

graph.entropy

*Graph spectral entropy*


---

**Description**

`graph.entropy` returns the spectral entropy of an undirected graph.

**Usage**

```
graph.entropy(Graph, ...)
```

**Arguments**

Graph	the undirected graph (igraph object). If Graph has the attribute <code>eigenvalues</code> containing the eigenvalues of Graph, such values will be used to compute its spectral density.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

**Value**

A list with class "statGraph" containing the following components:

method: a string indicating the used method.  
 info: a string showing details about the method.  
 data.name: a string with the data's name(s).  
 entropy: a real number corresponding to the graph spectral entropy.

**References**

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_, \*7\**, e49949. doi:10.1371/journal.pone.0049949.

**Examples**

```
set.seed(1)
G <- igraph::sample_gnp(n=100, p=0.5)
entropy <- graph.entropy(Graph = G)
entropy
```

---

graph.hclust                      *Hierarchical cluster analysis on a list of graphs.*

---

**Description**

Given a list of graphs, `graph.hclust` builds a hierarchy of clusters according to the Jensen-Shannon divergence between graphs.

**Usage**

```
graph.hclust(Graphs, k, clus_method = "complete", dist = "JS", ...)
```

**Arguments**

Graphs                      a list of undirected graphs. If each graph has the attribute `eigenvalues` containing its eigenvalues, such values will be used to compute their spectral density.

k                              the number of clusters.

clus\_method                  the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

dist                          string indicating if you want to use the "JS" (default), "L1" or "L2" distances. "JS" means Jensen-Shannon divergence.

...                            Other relevant parameters for [graph.spectral.density](#).

**Value**

A list containing:

- hclust: an object of class hclust which describes the tree produced by the clustering process.
- cluster: the clustering labels for each graph.

**References**

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_, \*7\*, e49949.* doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *\_Density Estimation\_*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *\_J. Am. Statist. Assoc.\_, \*21\*, 65-66.*

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *\_Journal of the Royal Statistical Society series B\_, 53, 683-690.* <http://www.jstor.org/stable/2345597>.

**Examples**

```
set.seed(1)
G <- list()
for (i in 1:5) {
  G[[i]] <- igraph::sample_gnp(50, 0.5)
}
for (i in 6:10) {
  G[[i]] <- igraph::sample_smallworld(1, 50, 8, 0.2)
}
for (i in 11:15) {
  G[[i]] <- igraph::sample_pa(50, power = 1, directed = FALSE)
}
graph.hclust(G, 3)
```

---

graph.kmeans

*K-means for Graphs*


---

**Description**

graph.kmeans clusters graphs following a k-means algorithm based on the Jensen-Shannon divergence between the spectral densities of the graphs.

**Usage**

```
graph.kmeans(Graphs, k, nstart = 2, dist = "JS", ...)
```

**Arguments**

Graphs	a list of undirected graphs. If each graph has the attribute <code>eigenvalues</code> containing its eigenvalues, such values will be used to compute their spectral density.
k	an integer specifying the number of clusters.
nstart	the number of trials of k-means clusterizations. The algorithm returns the clusterization with the best silhouette.
dist	string indicating if you want to use the "JS" (default), "L1" or "L2" distances. "JS" means Jensen-Shannon divergence.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

**Value**

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
data.name:	a string with the data's name(s).
cluster:	a vector of the same length of x containing the clusterization labels.

**References**

MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.

Lloyd, Stuart. "Least squares quantization in PCM." IEEE transactions on information theory 28.2 (1982): 129-137.

**Examples**

```
set.seed(1)
g <- list()
for(i in 1:5){
  g[[i]] <- igraph::sample_gnp(30, p=0.2)
}
for(i in 6:10){
  g[[i]] <- igraph::sample_gnp(30, p=0.5)
}
res <- graph.kmeans(g, k=2, nstart=2)
res
```

---

graph.model.selection *Graph model selection*


---

**Description**

graph.model.selection selects the graph model that best approximates the observed graph according to the Graph Information Criterion (GIC).

**Usage**

```
graph.model.selection(Graph, models = NULL, parameters = NULL, ...)
```

**Arguments**

Graph	the undirected graph (igraph object). If Graph has the attribute eigenvalues containing the eigenvalues of Graph, such values will be used to compute its spectral density.
models	either a vector of strings, or a list of functions: A vector of strings containing some of the following models: "ER" (Erdos-Renyi random graph), "GRG" (geometric random graph), "KR" (k regular random graph), "WS" (Watts-Strogatz model), and "BA" (Barabási-Albert model). A list of functions. Each function returns a graph (igraph object) generated by a graph model and has two arguments: the graph size and the model parameter, in this order. If the argument models is NULL, then the "ER", "WS", and "BA" models will be considered for the model selection.
parameters	list of numeric vectors or list of lists. If a list of numeric vectors is given, then each vector contains the values that will be considered for the parameter estimation of the corresponding model. If a list of lists is given, then each list contains lo and hi elements that indicate the model's parameter search interval <lo,hi>. If the user does not provide the argument parameters, then default values are used for the predefined models ("ER", "GRG", "KR", "WS", and "BA") as done in graph.param.estimator.
...	Other relevant parameters for <a href="#">graph.param.estimator</a> .

**Value**

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
model:	the indice(s) or name(s) of the selected model(s), i. e. the model(s) that minimize(s) the Graph Information Criterion (GIC).
estimates:	a matrix in which each row corresponds to a model, the column "param" corresponds to the parameter estimate, and the column "GIC" corresponds to the Graph Information Criterion (GIC), i. e. the distance measure (Kullback-Leibler divergence by default) between the observed graph and the model.

## References

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *\_Density Estimation\_*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *\_J. Am. Statist. Assoc.\_*, \*21\*, 65-66.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *\_Journal of the Royal Statistical Society series B\_*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

## Examples

```
## Example using an igraph object as input data
set.seed(1)
G <- igraph::sample_gnp(n=30, p=0.5)

# Using strings to indicate the graph models
result1 <- graph.model.selection(G, models=c("ER", "WS"), eps = 0.5)
result1

## Using functions to describe the graph models
# Erdos-Renyi graph
model1 <- function(n, p) {
  return(igraph::sample_gnp(n, p))
}
# Watts-Strogatz small-world graph
model2 <- function(n, pr, K=8) {
  return(igraph::sample_smallworld(1, n, K, pr))
}
parameters <- list(seq(0.01, 0.99, 0.49), seq(0.01, 0.99, 0.49))
result2 <- graph.model.selection(G, list(model1, model2), parameters)
result2
```

---

graph.mult.scaling      *Multidimensional scaling of graphs*

---

## Description

graph.mult.scaling performs multidimensional scaling of graphs. It takes the Jensen-Shannon divergence between graphs (JS) and uses the cmdscale function from the stats package to obtain a set of points such that the distances between the points are similar to JS.

**Usage**

```
graph.mult.scaling(
  Graphs,
  plot = TRUE,
  type = "n",
  dist = "JS",
  main = "",
  ...
)
```

**Arguments**

Graphs	a list of undirected graphs. If each graph has the attribute <code>eigenvalues</code> containing its eigenvalues, such values will be used to compute their spectral density.
plot	logical. If TRUE (default) the points chosen to represent the Jensen-Shannon divergence between graphs are plotted.
type	what type of plot should be drawn. The default value is "n", which indicates that the points will not be plotted (i. e. only the labels of the graphs will be plotted).
dist	string indicating if you want to use the "JS" (default), "L1" or "L2" distances. "JS" means Jensen-Shannon divergence.
main	title of the plot (default value is "").
...	additional parameters for <a href="#">graph.spectral.density</a> .

**Value**

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
data.name:	a string with the data's name(s).
values:	a matrix in which each column corresponds to a coordinate and each row corresponds to a graph (point). Then, each row gives the coordinates of the points chosen to represent the Jensen-Shannon divergence (by default), L1, or L2 distance between graphs.

**References**

- Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.
- Silverman, B. W. (1986) *\_Density Estimation\_*. London: Chapman and Hall.
- Sturges, H. A. The Choice of a Class Interval. *\_J. Am. Statist. Assoc.\_*, \*21\*, 65-66.
- Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *\_Journal of the Royal Statistical Society series B\_*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

**Examples**

```

set.seed(1)
G <- list()
for (i in 1:5) {
  G[[i]] <- igraph::sample_gnp(50, 0.5)
}
for (i in 6:10) {
  G[[i]] <- igraph::sample_smallworld(1, 50, 8, 0.2)
}
for (i in 11:15) {
  G[[i]] <- igraph::sample_pa(50, power = 1, directed = FALSE)
}
graph.mult.scaling(G)

```

---

graph.param.estimator *Graph parameter estimator*

---

**Description**

graph.param.estimator estimates the parameter that best approximates the model to the observed graph according to the Graph Information Criterion (GIC).

**Usage**

```

graph.param.estimator(
  Graph,
  model,
  interval = NULL,
  eps = 0.01,
  search = "grid",
  ...
)

```

**Arguments**

Graph	the undirected graph (igraph object). If Graph has the attribute 'eigenvalues' containing the eigenvalues of Graph, such values will be used to compute spectral density of the graph.
model	either a string or a function: A string that indicates one of the following models: "ER" (Erdos-Renyi random graph), "GRG" (geometric random graph), "KR" (k regular random graph), "WS" (Watts-Strogatz model), and "BA" (Barabási-Albert model). A function that returns a graph (represented by its adjacency matrix) generated by a graph model. It must contain two arguments: the first one corresponds to the graph size and the second to the parameter of the model.



interval	numeric vector containing the values that will be considered for the parameter estimation, or a list containing 'lo' and 'hi' that indicates the model's parameter search interval <lo,hi>. The graph.param.estimator will return the element of 'parameter' that minimizes the GIC. If the user does not provide the argument parameters, and model is a string, then default values are used for the predefined models ("ER", "GRG", "KR", "WS", and "BA"). The default parameter argument corresponds to a sequence from 0 to 1 with step eps for the "ER" model (Erdos-Renyi random graph), in which the parameter corresponds to the probability to connect a pair of vertices; 0 to sqrt(2) with step eps for the "GRG" model (geometric random graph), in which the parameter corresponds to the radius used to construct the geometric graph in a unit square; 0 to 'n' with step n*eps for the "KR" model (k regular random graph), in which the parameter of the model corresponds to the degree k of a regular graph; 0 to 1 with step eps for the "WS" model (Watts-Strogatz model), in which the parameter corresponds to the probability to reconnect a vertex; and 0 to 3 with step eps for the "BA" model (Barabási-Albert model), in which the parameter corresponds to the scaling exponent.
eps	precision of the grid and ternary search (default is 0.01).
search	string that indicates the search algorithm to find the parameter with the smallest GIC. If "grid" (default) parameter is estimated using grid search, and only works when method is not "fast". If "ternary" parameter is estimated using ternary search.
...	Other relevant parameters for <a href="#">GIC</a> .

### Value

A list with class "statGraph" containing the following components:

method:	a string indicating the used method.
info:	a string showing details about the method.
data.name:	a string with the data's name(s).
param:	the parameter estimate. For the "ER", "GRG", "KR", "WS", and "BA" models, the parameter corresponds to the probability to connect a pair of vertices, the radius used to construct the geometric graph in a unit square, the degree k of a regular graph, the probability to reconnect a vertex, and the scaling exponent, respectively.
dist:	the distance between the observed graph and the graph model with the estimated parameter.

### References

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *PLoS ONE*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *Density Estimation*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *J. Am. Statist. Assoc.*, \*21\*, 65-66.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

### Examples

```
set.seed(1)
G <- igraph::sample_gnp(n=50, p=0.5)

# Using a string to indicate the graph model
result1 <- graph.param.estimator(G, "ER", eps=0.25)
result1

# Using a function to describe the graph model
# Erdos-Renyi graph
set.seed(1)
model <- function(n, p) {
  return(igraph::sample_gnp(n, p))
}
result2 <- graph.param.estimator(G, model, seq(0.2, 0.8, 0.1))
result2
```

---

graph.spectral.density

*Graph spectral density*

---

### Description

graph.spectral.density returns the exact or degree-based spectral density in the interval <from,to> by using npoints discretization points.

### Usage

```
graph.spectral.density(Graph, method = "diag", ...)
```

### Arguments

Graph	the undirected graph (igraph object). If Graph has the attribute eigenvalues containing the eigenvalues of Graph, such values will be used to compute its spectral density.
method	String that specifies the method to obtain the spectral density. It can take two possible values 'diag' (Default) and 'fast'. If 'diag' is used then the exact spectral density is obtained, otherwise the degree-based spectral density is obtained.

... Other relevant parameters to obtain the spectral density such as `from`, `to`, and `npoints`. `from`, `to` specify the lower and upper bound of the eigenvalues' support (automatically computed if not given); and `npoints` is the number of discretization points (default 1024) of the interval `<from,to>`. There are other parameters that depend on the value of the parameter `method`: If `method='diag'`, then the parameter `bandwidth` can be used. This parameter is a string that specifies the criterion to choose the bandwidth during the spectral density estimation. Choose between the following criteria: "Silverman" (default), "Sturges", "bcv", "ucv" and "SJ". "bcv" is an abbreviation of biased cross-validation, while "ucv" means unbiased cross-validation. "SJ" implements the methods of Sheather & Jones (1991) to select the bandwidth using pilot estimation of derivatives. Otherwise, if `method='fast'`, then the parameter `numCores` can be used. This parameter specifies the number of cores (default 1) to use for parallelization.

### Value

A list with class "statGraph" containing the following components:

<code>method</code> :	a string indicating the used method.
<code>info</code> :	a string showing details about the method.
<code>data.name</code> :	a string with the data's name(s).
<code>x</code> :	a vector corresponding to the x axis coordinates of the density function.
<code>y</code> :	a vector corresponding to the y axis coordinates of the density function.
<code>from</code> :	a real number corresponding to the smallest value of the x axis.
<code>to</code> :	a real number corresponding to the largest value of the x axis.

### References

- # Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.
- Silverman, B. W. (1986) *\_Density Estimation\_*. London: Chapman and Hall.
- Sturges, H. A. The Choice of a Class Interval. *\_J. Am. Statist. Assoc.\_*, \*21\*, 65-66.
- Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *\_Journal of the Royal Statistical Society series B\_*, 53, 683-690. <http://www.jstor.org/stable/2345597>.
- Newman, M. E. J., Zhang, X., & Nadakuditi, R. R. (2019). Spectra of random networks with arbitrary degrees. *Physical Review E*, 99(4), 042309.

### Examples

```
set.seed(42)
G <- igraph::sample_smallworld(dim = 1, size = 50, nei = 2, p = 0.2)

# Obtain the spectral density
density <- graph.spectral.density(Graph = G)
density
```

---

graph.takahashi.test *Test for the Jensen-Shannon divergence between graphs*

---

### Description

graph.takahashi.test tests whether two sets of graphs were generated by the same random graph model. This bootstrap test is based on the Jensen-Shannon (JS) divergence between graphs.

### Usage

```
graph.takahashi.test(Graphs1, Graphs2, maxBoot = 1000, dist = "JS", ...)
```

### Arguments

Graphs1	a list of undirected Graphs. If each graph has the attribute eigenvalues containing its eigenvalues, such values will be used to compute their spectral density.
Graphs2	a list of undirected Graphs. If each graph has the attribute eigenvalues containing its eigenvalues, such values will be used to compute their spectral density.
maxBoot	integer indicating the number of bootstrap resamplings (default 1000).
dist	string indicating if you want to use the "JS" (default), "L1" or "L2" distances. "JS" means Jensen-Shannon divergence.
...	Other relevant parameters for <a href="#">graph.spectral.density</a> .

### Details

Given two lists of graphs, Graphs1 and Graphs2, graph.takahashi.test tests H0: "JS divergence between Graphs1 and Graphs2 is 0" against H1: "JS divergence between Graphs1 and Graphs2 is larger than 0".

### Value

A list with class "hctest" containing the following components:

statistic:	the value of the Jensen-Shannon divergence (default), L1 or L2 between 'Graphs1' and 'Graphs2'.
p.value:	the p-value of the test.
method:	a string indicating the used method.
data.name:	a string with the data's name(s).

### References

Takahashi, D. Y., Sato, J. R., Ferreira, C. E. and Fujita A. (2012) Discriminating Different Classes of Biological Networks by Analyzing the Graph Spectra Distribution. *\_PLoS ONE\_*, \*7\*, e49949. doi:10.1371/journal.pone.0049949.

Silverman, B. W. (1986) *\_Density Estimation\_*. London: Chapman and Hall.

Sturges, H. A. The Choice of a Class Interval. *J. Am. Statist. Assoc.*, \*21\*, 65-66.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

### Examples

```
set.seed(1)
G1 <- G2 <- list()
for (i in 1:20) {
  G1[[i]] <- igraph::sample_gnp(n=50, p=0.5)
}
for (i in 1:20) {
  G2[[i]] <- igraph::sample_gnp(n=50, p=0.51)
}
result <- graph.takahashi.test(G1, G2, maxBoot=100)
result
```

---

graph.tang.test	<i>Tang hypothesis testing for random graphs.</i>
-----------------	---

---

### Description

Given two independent finite-dimensional random dot product graphs, `graph.tang.test` tests if they have generating latent positions that are drawn from the same distribution.

### Usage

```
graph.tang.test(
  Graph1,
  Graph2,
  dim,
  sigma = NULL,
  alpha = 0.05,
  bootstrap_sample = 200,
  printResult = FALSE
)
```

### Arguments

Graph1	the first undirected graph (igraph object). If Graph has the attribute <code>eigenvalues</code> containing the eigenvalues of Graph, such values will be used to compute its spectral density.
Graph2	the second undirected graph (igraph object). If Graph has the attribute <code>eigenvalues</code> containing the eigenvalues of Graph, such values will be used to compute its spectral density.
dim	dimension of the adjacency spectral embedding.

sigma	a real value indicating the kernel bandwidth. If NULL (default) the bandwidth is calculated by the method.
alpha	the significance level for the test (default is 0.05).
bootstrap_sample	integer indicating the number of bootstrap resamples (default is 200).
printResult	logical indicating if the test must print the result (default is FALSE).

### Value

A list with class "hctest" containing the following components:

statistic:	the T-value of the test.
p.value:	the p-value of the test.
method:	a string indicating the used method.
data.name:	a string with the data's name(s).

### References

Tang, Minh, et al. "A nonparametric two-sample hypothesis testing problem for random graphs." *Bernoulli* 23.3 (2017): 1599-1630.

Tang, Minh, et al. "A semiparametric two-sample hypothesis testing problem for random graphs." *Journal of Computational and Graphical Statistics* 26.2 (2017): 344-354.

### Examples

```
set.seed(1)

## test under H0
lpvs <- matrix(rnorm(200), 20, 10)
lpvs <- apply(lpvs, 2, function(x) { return (abs(x)/sqrt(sum(x^2))) })
Graph1 <- igraph::sample_dot_product(lpvs)
Graph2 <- igraph::sample_dot_product(lpvs)
D <- graph.tang.test(Graph1,Graph2, 5, printResult = TRUE)

## test under H1
lpvs2 <- matrix(pnorm(200), 20, 10)
lpvs2 <- apply(lpvs2, 2, function(x) { return (abs(x)/sqrt(sum(x^2))) })
g2 <- suppressWarnings(igraph::sample_dot_product(lpvs2))
D <- graph.tang.test(Graph1,Graph2, 5, printResult = TRUE)
```

---

 sp.anogva

*Semi-Parametric Analysis Of Graph Variability (ANOGVA)*


---

## Description

sp.anogva statistically tests whether two or more graphs are generated by the same model and set of parameters.

## Usage

```
sp.anogva(Graphs, model, maxBoot = 100, ...)
```

## Arguments

Graphs	a list of undirected graphs. If each graph has the attribute <code>eigenvalues</code> containing its eigenvalues, such values will be used to compute their spectral density.
model	A string that indicates one of the following models: "ER" (Erdos-Renyi random graph model), "GRG" (geometric random graph model), "WS" (Watts-Strogatz random graph model), and "BA" (Barabási-Albert random graph model).
maxBoot	integer indicating the number of bootstrap resamples (default is 500).
...	Other relevant parameters for <a href="#">graph.param.estimator</a> .

## Value

A list with class "htest" containing the following components:

statistic:	the F statistic of the test.
p.value:	the p-value of the test.
method:	a string indicating the used method.
data.name:	a string with the data's name(s).
estimates:	a vector containing the estimated parameters for each graph.

## References

Andre Fujita, Eduardo Silva Lira, Suzana de Siqueira Santos, Silvia Yumi Bando, Gabriela Eleuterio Soares, Daniel Yasumasa Takahashi. A semi-parametric statistical test to compare complex networks, *Journal of Complex Networks*, cnz028, <https://doi.org/10.1093/comnet/cnz028>

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society series B*, 53, 683-690. <http://www.jstor.org/stable/2345597>.

**Examples**

```
set.seed(1)
model <- "ER"
G <- list()

# Under H0
G[[1]] <- igrph::sample_gnp(50, 0.5)
G[[2]] <- igrph::sample_gnp(50, 0.5)
G[[3]] <- igrph::sample_gnp(50, 0.5)
result1 <- sp.anogva(G, model, maxBoot = 300)
result1

# Under H1
G[[1]] <- igrph::sample_gnp(50, 0.5)
G[[2]] <- igrph::sample_gnp(50, 0.55)
G[[3]] <- igrph::sample_gnp(50, 0.5)
result2 <- sp.anogva(G, model, maxBoot = 300)
result2
```



# Index

- \* **analysis\_of\_graph\_variability**
    - anogva, 2
  - \* **autocorrelation**
    - graph.acf, 5
  - \* **clustering**
    - graph.hclust, 10
  - \* **correlation\_coefficient**
    - graph.cor.test, 7
  - \* **distance\_matrix**
    - graph.dist, 8
  - \* **eigenvalue\_density**
    - graph.spectral.density, 18
  - \* **graph.cem**
    - graph.cem, 6
  - \* **graph\_comparison**
    - graph.takahashi.test, 20
  - \* **graph\_information\_criterion**
    - GIC, 3
  - \* **k-means**
    - graph.kmeans, 11
  - \* **model\_selection**
    - graph.model.selection, 13
  - \* **multidimensional\_scaling**
    - graph.mult.scaling, 14
  - \* **parameter\_estimation**
    - graph.param.estimator, 16
  - \*
    - semi\_parametric\_analysis\_of\_graph\_variability**
      - sp.anogva, 23
  - \* **spectral\_entropy**
    - graph.entropy, 9
- anogva, 2
- GIC, 3, 17
- graph.acf, 5
- graph.cem, 6
- graph.cor.test, 7
- graph.dist, 8
- graph.entropy, 9
- graph.hclust, 10
- graph.kmeans, 11
- graph.model.selection, 13
- graph.mult.scaling, 14
- graph.param.estimator, 6, 13, 16, 23
- graph.spectral.density, 2, 4, 9, 10, 12, 15, 18, 20
- graph.takahashi.test, 20
- graph.tang.test, 21
- sp.anogva, 23