Package 'DiffNet'

January 20, 2025

Type Package

Title Identifying Significant Node Scores using Network Diffusion Algorithm

Version 1.0.2

Description Designed for network analysis, leveraging the personalized PageRank algorithm to calculate node scores in a given graph. This innovative approach allows users to uncover the importance of nodes based on a customized perspective, making it particularly useful in fields like bioinformatics, social network analysis, and more.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Depends igraph, assertthat

Imports MASS, parallel

NeedsCompilation no

Author Farzaneh Firoozbakht [aut, cre, cph]

Maintainer Farzaneh Firoozbakht <faren.firoozbakht@gmail.com>

Repository CRAN

Date/Publication 2023-11-22 09:20:15 UTC

Contents

actual_score 2
graph_generation 2
multiple_testing_correction 3
null_score 4
pval 5

Index **6**

actual_score *Calculation of diffusion score for each node*

Description

Calculation of diffusion score for each node

Usage

```
actual_score(graph, initial.score, damping = 0.7)
```

Arguments

graph	an igraph object with the length of N
initial.score	a named vector of node preferences of length N served as the initial values for diffusion algorithm.
damping	The damping factor of the diffusion algorithm.

Details

This function calculates the diffusion score for each node using the personalized page rank algorithm.

Value

a vector of diffusion scores.

Examples

```
graph = graph_generation(n.nodes = 10, prob.connection = 0.5)
initial_score = c(rep(0,5),0.2, 0.3, 0, 0, 0.5)
names(initial_score) = igraph::V(graph)
Actual_score = actual_score(graph = graph, initial.score = initial_score, damping = 0.7)
```

graph_generation *dummy graph generation*

Description

dummy graph generation

Usage

```
graph_generation(n.nodes = 10, prob.connection = 0.5)
```

Arguments

n.nodes	number of nodes
prob.connection	node connection probability (default=0.5)

Details

Generate a random graph

Value

igraph object

Examples

```
graph = graph_generation(n.nodes = 10, prob.connection = 0.5)
initial_score = c(rep(0,5),0.2, 0.3, 0, 0, 0.5)
names(initial_score) = igraph::V(graph)
```

---

multiple_testing_correction

Correction for multiple testing

Description

Correction for multiple testing

Usage

```
multiple_testing_correction(p.values, method = "BH")
```

Arguments

p.values	a vector of p.values
method	method of correction: c("BH", "bonferroni")

Details

Correction for multiple testing

Value

vector of q-values

Examples

```
graph = graph_generation(n.nodes = 10, prob.connection = 0.5)
initial_score = c(rep(0,5),0.2, 0.3, 0, 0, 0.5)
names(initial_score) = igraph::V(graph)
Actual_score = actual_score(graph = graph, initial.score = initial_score, damping = 0.7)
Null_score = null_score(graph = graph, initial.score = initial_score, damping = 0.7, N.repeat = 10)
pvalue = pval(actual.scores = Actual_score, null.scores = Null_score, method = "non_parametric")
adj_nodes = multiple_testing_correction(pvalue)
```

---

null_score *Calculation of diffusion null scores for each node*

Description

Calculation of diffusion null scores for each node

Usage

```
null_score(graph, initial.score, damping = 0.7, N.repeat = 10, n.cores = 1)
```

Arguments

graph	an igraph object with the length of N
initial.score	a named vector of node preferences of length N served as the initial values for diffusion algorithm.
damping	The damping factor of the diffusion algorithm.
N.repeat	number of permutation repeats of null scores.
n.cores	number of cores for parallel processing.

Details

This function calculates the null diffusion score for each node using the personalized page rank algorithm. The initial values are obtained by permuting the given `initial.score`

Value

a matrix of null diffusion scores (`N.repeat`—BY—number_of_nodes).

Examples

```
graph = graph_generation(n.nodes = 10, prob.connection = 0.5)
initial_score = c(rep(0,5),0.2, 0.3, 0, 0, 0.5)
names(initial_score) = igraph::V(graph)
Null = null_score(graph, initial_score)
```

pval *Calculation of p-values for each score with respect to the null.*

Description

Calculation of p-values for each score with respect to the null.

Usage

```
pval(actual.scores, null.scores, method = "exponential")
```

Arguments

actual.scores	a vector including actual scores with the length of number of nodes (N_nodes).
null.scores	a matrix of null scores with the dimension of N_nodes x N_repeat
method	statistical test method: c("exponential", "gamma", "non_parametric")

Details

Calculate the p-value for each node based on the actual and null diffusion scores.

Value

vector of p-values

Examples

```
graph = graph_generation(n.nodes = 10, prob.connection = 0.5)
initial_score = c(rep(0,5),0.2, 0.3, 0, 0, 0.5)
names(initial_score) = igraph::V(graph)
Actual_score = actual_score(graph = graph, initial.score = initial_score, damping = 0.7)
Null_score = null_score(graph = graph, initial.score = initial_score, damping = 0.7, N.repeat = 10)
pvalue = pval(actual.scores = Actual_score, null.scores = Null_score, method = "exponential")
```

Index

actual_score, 2

graph_generation, 2

multiple_testing_correction, 3

null_score, 4

pval, 5