

Package: scoredec (via r-universe)

October 20, 2024

Type Package

Title S-Core Graph Decomposition

Version 0.1.1

Maintainer Christos Adam <econp266@econ.soc.uoc.gr>

Description S-Core Graph Decomposition algorithm for graphs. This is a method for decomposition of a weighted graph, as proposed by Eidsaa and Almaas (2013) <[doi:10.1103/PhysRevE.88.062819](https://doi.org/10.1103/PhysRevE.88.062819)>. The high speed and the low memory usage make it suitable for large graphs.

License GPL-3

LinkingTo Rcpp

Encoding UTF-8

URL <https://github.com/cadam00/scoredec>,
<https://cadam00.github.io/scoredec/>

BugReports <https://github.com/cadam00/scoredec/issues>

Imports Rcpp (>= 1.0.12), Rfast, igraph

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr, rmarkdown

Repository <https://cadam00.r-universe.dev>

RemoteUrl <https://github.com/cadam00/scoredec>

RemoteRef HEAD

RemoteSha 905461dcd8a52347c53ad5d81947e0085dbc5d19

Contents

<code>s_coreness</code>	2
Index	4

`s_coreness`*s-core community decomposition*

Description

s-core community decomposition

Usage

```
s_coreness(g = NULL, W = NULL, mode = "all")
```

Arguments

<code>g</code>	<code>igraph</code> object. It is a weighted graph. If it is not weighted, then the <code>igraph::coreness</code> function will be used. It can be used as an alternative to <code>W</code> .
<code>W</code>	<code>matrix</code> object. It is an adjacency matrix. It can be used as an alternative to <code>g</code> .
<code>mode</code>	character object. It can be one of "all", "in" or "out".

Details

s-core community decomposition implementation. Only one of `g` or `W` must be provided.

While the source code is not as clear as the one at `brainGraph::s_core`, it is very speed and memory efficient. In case that the adjacency matrix `W` is provided instead of the graph `g` is provided, then this function is very speed and memory efficient.

Note that in cases that the adjacency matrix `W` is known to be symmetric (checked, for example, with `base::isSymmetric` or `Rfast::is.symmetric`), then `mode = "in"` and `mode = "out"` will produce the same result more efficiently. For efficiency reasons not checking it is chosen, but user should do it.

Value

Integer vector with s-coreness attribute to each vertex.

References

Eidsaa, M. and Almaas, E. (2013) 's-core network decomposition: A generalization of k-core analysis to weighted networks', *Phys. Rev. E.*, American Physical Society, **88**, 062819. [doi:10.1103/PhysRevE.88.062819](https://doi.org/10.1103/PhysRevE.88.062819).

Watson, C.G. (2024). `brainGraph`: Graph Theory Analysis of Brain MRI Data. R package version 3.1.0. [doi:10.32614/CRAN.package.brainGraph](https://doi.org/10.32614/CRAN.package.brainGraph).

Examples

```
set.seed(42)

## Create a dummy symmetric adjacency matrix
n <- 5
W <- matrix(runif(n^2),n)
W[lower.tri(W)] = t(W)[lower.tri(W)]
diag(W) <- 0

print(scoredec::s_coreness(g = NULL, W = W, mode = "all"))
#> [1] 3 1 2 4 4

base::isSymmetric(W)
#> [1] TRUE

all.equal(scoredec::s_coreness(g = NULL, W = W, mode = "all"),
scoredec::s_coreness(g = NULL, W = W, mode = "in"))
#> [1] TRUE

# Create a dummy undirected graph
g <- igraph::graph_from_adjacency_matrix(adjmatrix = W,
                                         mode       = "undirected",
                                         weighted  = TRUE)

print(scoredec::s_coreness(g = g, W = NULL, mode = "all"))
#> [1] 3 1 2 4 4
```

Index

s_coreness, [2](#)