

Package ‘leidenbase’

December 16, 2025

Type Package

Title R and C/C++ Wrappers to Run the Leiden find_partition() Function

Version 0.1.36

Description An R to C/C++ interface that runs the Leiden community detection algorithm to find a basic partition (). It runs the equivalent of the 'leidenalg' find_partition() function, which is given in the 'leidenalg' distribution file 'leiden/src/functions.py'. This package includes the required source code files from the official 'leidenalg' distribution and functions from the R 'igraph' package. The 'leidenalg' distribution is available from <https://github.com/vtraag/leidenalg/> and the R 'igraph' package is available from <https://igraph.org/r/>. The Leiden algorithm is described in the article by Traag et al. (2019) <[doi:10.1038/s41598-019-41695-z](https://doi.org/10.1038/s41598-019-41695-z)>. Leidenbase includes code from the packages:
igraph version 0.9.8 with license GPL (>= 2),
leidenalg version 0.8.10 with license GPL 3.

Imports igraph (>= 0.9.0)

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Suggests rmarkdown, knitr, testthat (>= 3.1.0), pandoc

URL <https://github.com/cole-trapnell-lab/leidenbase>

BugReports <https://github.com/cole-trapnell-lab/leidenbase/issues>

NeedsCompilation yes

Maintainer Brent Ewing <bge@uw.edu>

VignetteBuilder knitr

Author Brent Ewing [aut, cre],
Vincent Traag [ctb],
Gábor Csárdi [ctb],
Tamás Nepusz [ctb],
Szabolcs Horvat [ctb],
Fabio Zanini [ctb]
Repository CRAN
Date/Publication 2025-12-16 03:30:02 UTC

Contents

leiden_find_partition	2
Index	5

leiden_find_partition	<i>Leiden find partition community detection function</i>
-----------------------	---

Description

R to C wrapper that runs the basic Leiden community detection algorithm, which is similar to the find_partition() function in the python Leidenalg distribution.

Usage

```
leiden_find_partition(  
  igraph,  
  partition_type = c("CPMVertexPartition", "ModularityVertexPartition",  
    "RBConfigurationVertexPartition", "RBERVertexPartition",  
    "SignificanceVertexPartition", "SurpriseVertexPartition"),  
  initial_membership = NULL,  
  edge_weights = NULL,  
  node_sizes = NULL,  
  seed = NULL,  
  resolution_parameter = 0.1,  
  num_iter = 2,  
  verbose = FALSE  
)
```

Arguments

- igraph R igraph graph.
- partition_type String partition type name. Default is CPMVertexParition.
- initial_membership Numeric vector of initial membership assignments of nodes. These are 1-based indices. Default is one community per node.

edge_weights	Numeric vector of edge weights. Default is 1.0 for all edges.
node_sizes	Numeric vector of node sizes. Default is 1 for all nodes.
seed	Numeric random number generator seed. The seed value must be either NULL for random seed values or greater than 0 for a fixed seed value. Default is NULL.
resolution_parameter	Numeric resolution parameter. The value must be greater than 0.0. Default is 0.1. The resolution_parameter is ignored for the partition_types ModularityVertexPartition, SignificanceVertexPartition, and SurpriseVertexPartition.
num_iter	Numeric number of iterations. Default is 2.
verbose	A logic flag to determine whether or not we should print run diagnostics.

Details

The Leiden algorithm is described in From Louvain to Leiden: guaranteeing well-connected communities. V. A. Traag and L. Waltman and N. J. van Eck Scientific Reports, 9(1) (2019) DOI: 10.1038/s41598-019-41695-z.

Significance is described in Significant Scales in Community Structure V. A. Traag, G. Krings, and P. Van Dooren Scientific Reports, 3(1) (2013) DOI: 10.1038/srep02930

Notes excerpted from `leidenalg/src/VertexPartition.py`

- *CPMVertexPartition* Implements Constant Potts Model. This quality function uses a linear resolution parameter and is well-defined for both positive and negative edge weights.
- *ModularityVertexPartition* Implements modularity. This quality function is well-defined only for positive edge weights.
- *RBConfigurationVertexPartition* Implements Reichardt and Bornholdt's Potts model with a configuration null model. This quality function uses a linear resolution parameter and is well-defined only for positive edge weights.
- *RBERVertexPartition* Implements Reichardt and Bornholdt's Potts model with an Erdos-Renyi null model. This quality function uses a linear resolution parameter and is well-defined only for positive edge weights.
- *SignificanceVertexPartition* Implements Significance. This quality function is well-defined only for unweighted graphs.
- *SurpriseVertexPartition* Implements (asymptotic) Surprise. This quality function is well-defined only for positive edge weights.

Value

A named list consisting of a numeric vector of the node community memberships (1-based indices), a numeric quality value, a numeric modularity, a numeric significance, a numeric vector of edge weights within each community, a numeric vector of edge weights from each community, a numeric vector of edge weights to each community, and total edge weight in the graph.

Index

`leiden_find_partition`, [2](#)