

Package: gephi (via r-universe)

August 23, 2024

Title Write graph objects to a file format that gephi understands

Version 1.0.1

Description Sometimes you want to visualize your graph information in 'gephi'. 'Gephi' needs certain column names to make sense of the file. This package writes edge and node information to csv files with correct names so you can read it in easily with gephi. You can also read the files in with igraph/tidygraph. It makes it much easier to switch between R and 'gephi'. This R-package does not interface with the open source network visualisation tool 'gephi' but it writes and reads csv files in a way that 'gephi' expects, reducing the friction when switching between tools.

License MIT + file LICENSE

BugReports <https://github.com/RMHogervorst/gephi/issues>

URL <http://rmhogervorst.nl/gephi/>,
<https://github.com/RMHogervorst/gephi>

Depends R(>= 2.10)

Suggests readr, testthat, spelling

ByteCompile true

Encoding UTF-8

LazyData true

Imports igraph (>= 1.2.4.0)

RoxygenNote 7.2.0

Language en-US

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

RemoteUrl <https://github.com/RMHogervorst/gephi>

RemoteRef HEAD

RemoteSha 52552a21ae5b0bed4f17c6c4693b7b27dafba14

Contents

gephi	2
gephi_read_edges_csv	2
gephi_write_edges_from_df	3
gephi_write_nodes	4
graphexample	5
Index	6

gephi	<i>Gephi.</i>
-------	---------------

Description

Make your workflow with network data easier. The gephi package does not interface with the open source network visualisation software 'gephi' but only writes csv files in a format that can be easily read by 'gephi'.

Graphs, networks, edges and nodes

Network science is a broad field with many practitioners who, unfortunately don't use the same language to describe the same things. A graph is a network, not to be confused with graphics, or a graph like a scatterplot or line chart. A network (known as graph) consists of points (known as vertices or nodes) and connections between them (known as vertices, links or edges). The gephi network visualisation tool uses edges and nodes as description, igraph talks about graphs $G()$, vertices $V()$ and edges $E()$, tidygraph talks about edges and nodes. We don't care as long as we can work with it.

gephi_read_edges_csv	<i>Read edges</i>
----------------------	-------------------

Description

This function reads in an edges csv file exported from gephi into igraph.

Usage

```
gephi_read_edges_csv(
  path,
  source = "Source",
  target = "Target",
  directed = TRUE
)
```

Arguments

path	The file where to read from e.g.: edges.csv or data/edges.csv
source	What column is the source, defaults to 'Source'.
target	What column is the target, defaults to 'Target'.
directed	is the graph directed or not? defaults to TRUE.

gephi_write_edges_from_df

Write edges from a data.frame to a file

Description

If you have a dataframe that represents a graph and you don't want to turn it into an igraph/tidygraph object first, this function will turn it into a csv. Give the column name where the edges start and the column name of the edge end. If you don't specify it, this function expects the first column to be a the start and second column the end point of an edge.

Usage

```
gephi_write_edges_from_df(
  dataframe,
  from = NULL,
  to = NULL,
  path,
  na = "",
  verbose = TRUE
)
```

Arguments

dataframe	which dataframe
from	which column does the edge start
to	which column does the edge end
path	The file where to save to e.g.: edges.csv or data/edges.csv
na	How to record missing values, defaults to "", nothing / empty values
verbose	by default these functions are chatty and will tell you what they do if you do not want that, set to FALSE

Value

invisible original object so you could continue using it in a pipe if you want to

See Also

Other write_graphs: [gephi_write_nodes\(\)](#)

gephi_write_nodes *Write igraph/ tidygraph to a format that gephi can read*

Description

write_nodes will write the nodes, and write_edges will write the edges to a csv file. You will always want to use the write_edges function, the write_nodes function has only added value if the nodes have attributes, such as Label, Age, color etc.

Usage

```
gephi_write_nodes(graph, path, na = "", verbose = TRUE)
```

```
gephi_write_edges(graph, path, na = "", verbose = TRUE)
```

```
gephi_write_both(graph, pathedges, pathnodes, na = "", verbose = TRUE)
```

Arguments

graph	The igraph or tidygraph object your work with
path	The file where to save to e.g.: edges.csv or data/edges.csv
na	How to record missing values, defaults to "", nothing / empty values
verbose	by default these functions are chatty and will tell you what they do if you do not want that, set to FALSE
pathedges	where to save the edges file e.g.: 'edges.csv'
pathnodes	where to save the nodes file e.g.: 'nodes.csv'

Details

If the file is not an igraph / tidygraph object, but a dataframe you could use the write_*_from_df functions. This function will assume the edges are between the first and second column, but you can specify if that is not the case.

Value

invisible original object so you could continue using it in a pipe if you want to

Functions

- `gephi_write_edges`: Write nodes data to csv
- `gephi_write_both`: Write both node and edge data away

See Also

Other write_graphs: [gephi_write_edges_from_df\(\)](#)

graphexample

a super simple graph example

Description

I created this file to show off and test some functions.

Usage

graphexample

Format

An object of class igraph of length 5.

Index

- * **datasets**

- graphexample, 5

- * **read_graphs**

- gephi_read_edges_csv, 2

- * **write_graphs, read_graphs**

- gephi, 2

- * **write_graphs**

- gephi_write_edges_from_df, 3

- gephi_write_nodes, 4

gephi, 2

gephi_read_edges_csv, 2

gephi_write_both (gephi_write_nodes), 4

gephi_write_edges (gephi_write_nodes), 4

gephi_write_edges_from_df, 3, 4

gephi_write_nodes, 3, 4

graphexample, 5