Package: triversity (via r-universe)

September 6, 2024

Title Diversity Measures on Multipartite Graphs

Version 2.0

Author Robin Lamarche-Perrin [aut, cre]

Maintainer Robin Lamarche-Perrin <Robin.Lamarche-Perrin@lip6.fr>

Description Computing diversity measures on multipartite graphs. This package first implements a parametrized family of such diversity measures which apply on probability distributions. Sometimes called ``True Diversity'', this family contains famous measures such as the richness, the Shannon entropy, the Herfindahl-Hirschman index, and the Berger-Parker index. Second, the package allows to apply these measures on probability distributions resulting from random walks between the parts of multipartite graphs. By defining an initial distribution on a given part of the graph and a path to follow between the different parts, the probability of the walker's position within the final part is then computed, thus providing a particular instance of diversity to measure.

Depends R (>= 3.2.3), stats, slam, data.tree

License GPL-3 | file LICENSE

Encoding UTF-8

RoxygenNote 6.0.1

Repository https://lamarche-perrin.r-universe.dev

RemoteUrl https://github.com/lamarche-perrin/triversity

RemoteRef HEAD

RemoteSha 554552e6fa5c326e1b261b30f15fc4d45479f23e

Contents

get_all_diversities 2
get_distribution_from_path 3
get_diversity_from_distribution 4
get_diversity_from_path 5

get_diversity_from_transition 6
get_multipartite 7
get_transition_from_path 9
tripartite_example 10
triversity 10

Index **12**

`get_all_diversities` *Compute all diversity flows associated to different random walks between the parts of a multipartite graph.*

Description

`get_all_diversities` builds on `get_diversity_from_path` to compute the diversity values associated to all possible paths of a given `length` between the different parts of the input multipartite `graph`.

Usage

```
get_all_diversities(graph, length = 2, cycles = TRUE, type = c("mean",
  "collective"), order = NULL, measure = NULL)
```

Arguments

`graph`	A multipartite graph obtained by calling the `get_multipartite` function.
`length`	A positive integer giving the maximal length of the computed paths.
`cycles`	A boolean value indicating if the diversity associated to paths containing cycles should be computed.
`type`	A vector specifying the types of diversity to compute, among `'mean'` and `'collective'`. See `get_diversity_from_path`.
`order`	A vector of positive floats (possibly including `Inf`) giving the orders of the diversity measures to be computed. If neither `order` nor `measure` is specified, a predefined list of 8 diversity measures is computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. Possible values are `richness`, `entropy`, `herfindahl`, and `bergerparker`.

Value

A dataframe summarising all the computed diversities.

Examples

```
data (tripartite_example)
graph <- get_multipartite (data=tripartite_example)
get_all_diversities (graph)
```

`get_distribution_from_path` *Compute the probability distribution associated to a random walk following a path between the parts of a multipartite graph.*

Description

`get_distribution_from_path` computes the probability distribution of a random walk following a given `path` between the different parts of the input multipartite `graph`. It starts at a given part with an initial probability distribution, then randomly follows the links of the graph between the different parts according to the input `path`, then stops at the last specified part.

Usage

```
get_distribution_from_path(graph, path, initial_distribution = NULL,
  initial_node = NULL)
```

Arguments

`graph`	A multipartite graph obtained by calling the `get_multipartite` function.
`path`	A vector of character strings giving the path that the random walk should follow between the different parts of the input multipartite `graph`. This path can be as long as wanted, with eventual cycles, and each string it contains should refer to a label in `graph$parts`.
`initial_distribution`	(optional) A vector of floats in [`0`,`1`] and summing to `1` giving the probability distribution to start with at the first part of the input `path`. It should hence contain as many values as there are nodes in the corresponding part. If not specified, this distribution is assumed uniform.
`initial_node`	(optional) A character string giving the label of a node in the first part of the input `path`. This node is then considered to have probability one, thus being equivalent to specifying an `initial_distribution` with only zeros except for one node. If not specified, no such node is defined and the initial distribution is then assumed uniform.

Value

A vector of floats in [`0`,`1`] and summing to `1` giving the probability distribution of the random walk when arriving at the last part, after having followed the input `path` within the different parts of the graph.

Examples

```
data (tripartite_example)
graph <- get_multipartite (data=tripartite_example)

path <- c(2,1,2,3)
```

```
as.matrix (get_distribution_from_path (graph, path))
as.matrix (get_distribution_from_path (graph, path, initial_distribution=c(1/3,0,0,2/3)))
as.matrix (get_distribution_from_path (graph, path, initial_node='i2'))
```

get_diversity_from_distribution

Compute the diversity of a probability distribution.

Description

`get_diversity_from_distribution` computes diversity values associated to an input probability `distribution`. The implemented diversity measures all belong to the parametrized family of "True Diversity" measures. They can either be specified by their diversity `order` in [`0`,`Inf`[or by their `measure` name when it corresponds to classical instances such as the richness, the Shannon entropy, the Herfindahl-Hirschman index, or the Berger-Parker index.

Usage

```
get_diversity_from_distribution(distribution, order = NULL, measure = NULL)
```

Arguments

`distribution`	A vector of floats in [`0`,`1`] and summing to `1` giving the probability distribution whose diversity is measured.
`order`	A vector of positive floats (possibly including `Inf`) giving the orders of the diversity measures to be computed. If neither `order` nor `measure` is specified, a predefined list of 8 diversity measures is computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. Possible values are `richness`, `entropy`, `herfindahl`, and `bergerparker`.

Value

A vector of positive floats giving the diversity values of the input probability distribution.

Examples

```
distribution <- c (1/4, 1/2, 1/12, 2/12)

get_diversity_from_distribution (distribution)
get_diversity_from_distribution (distribution, order=c(0,Inf), measure='entropy')
```

`get_diversity_from_path`

Compute the diversity associated to a random walk following a path between the parts of a multipartite graph.

Description

`get_diversity_from_path` computes diversity values of the probability distribution of a random walk following a `path` between the different parts of the input multipartite `graph`. It starts at a given part with an initial probability distribution, then randomly follows the links of the graph between the different parts according to the input `path`, then stops at the last specified part. The implemented diversity measures all belong to the parametrized family of "True Diversity" measures. They can either be specified by their diversity `order` in `[0,Inf[` or by their `measure` name when it corresponds to classical instances such as the richness, the Shannon entropy, the Herfindahl-Hirschman index, or the Berger-Parker index.

Usage

```
get_diversity_from_path(graph, path, type = "individual",
  mean_distribution = NULL, initial_distribution = NULL,
  initial_node = NULL, order = NULL, measure = NULL)
```

Arguments

`graph`	A multipartite graph obtained by calling the `get_multipartite` function.
`path`	A vector of character strings giving the path that the random walk should follow between the different parts of the input multipartite `graph`. This path can be as long as wanted, with eventual cycles, and each string it contains should refer to a label in `graph$parts`.
`type`	Either `'individual'`, to separately compute all individual diversities, `'mean'`, to compute their geometric mean, or `'collective'`, to compute the overall diversity.
`mean_distribution`	(optional, only when `type` == `'mean'`) A vector of floats in `[0,1]` and summing to 1 giving the probability distribution that is used to weight the diversity values when computing their geometric means. It should hence contain as many values as there are rows in the input `transition`. If not specified, this distribution is assumed uniform.
`initial_distribution`	(optional, only when `type` == `'collective'`) A vector of floats in `[0,1]` and summing to 1 giving the probability distribution to start with at the first part of the input `path`. It should hence contain as many values as there are nodes in the corresponding part. If not specified, this distribution is assumed uniform.
`initial_node`	(optional, only when `type` == `'collective'`) A character string giving the label of a node in the first part of the input `path`. This node is then considered to have probability one, thus being equivalent to specifying an `initial_distribution`

	with only zeros except for one node. If not specified, no such node is defined and the initial distribution is then assumed uniform.
`order`	A vector of positive floats (possibly including `Inf`) giving the orders of the diversity measures to be computed. If neither `order` nor `measure` is specified, a predefined list of 8 diversity measures is computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. Possible values are `richness`, `entropy`, `herfindahl`, and `bergerparker`.

Value

A matrix (or a vector) of positive floats giving the individual diversity values of the random walks following the input `path` (or their mean, or the collective diversity).

Examples

```
data (tripartite_example)
graph <- get_multipartite (data=tripartite_example)
path <- c(1,2,3)


get_diversity_from_path (graph, path, 'individual', measure=c('entropy','herfindahl'))
get_diversity_from_path (graph, path, 'individual', order=c(0,1,Inf))

# Mean of individual diversities
get_diversity_from_path (graph, path, 'mean')
get_diversity_from_path (graph, path, 'mean', mean_distribution=c(1/3,2/3))

# Collective diversities
get_diversity_from_path (graph, path, 'collective')
get_diversity_from_path (graph, path, 'collective', initial_distribution=c(0.75,0.25))
```

---

get_diversity_from_transition

Compute the diversity of a transition matrix, that is the individual diversities of its rows or their mean diversity.

Description

`get_diversity_from_transition` either computes the individual diversity values associated to the rows of the input `transition` matrix, or the geometric means of these values, optionally weighting them according to the input `mean_distribution`. This hence allows to compute conditional diversity values associated to a transition matrix.

Usage

```
get_diversity_from_transition(transition, type = "individual",
  mean_distribution = NULL, order = NULL, measure = NULL)
```

Arguments

`transition`	A matrix of floats in [`0`,`1`], with all lines summing to `1`, giving the transition matrix from which the individual diversity values are computed.
`type`	Either `'individual'`, to separately compute all individual diversities, or `'mean'`, to compute their geometric mean.
`mean_distribution`	(optional, only when `type == 'mean'`) A vector of floats in [`0`,`1`] and summing to `1` giving the probability distribution that is used to weight the diversity values when computing their geometric means. It should hence contain as many values as there are rows in the input `transition`. If not specified, this distribution is assumed uniform.
`order`	A vector of positive floats (possibly including `Inf`) giving the orders of the diversity measures to be computed. If neither `order` nor `measure` is specified, a predefined list of 8 diversity measures is computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. Possible values are `richness`, `entropy`, `herfindahl`, and `bergerparker`.

Value

A matrix (or a vector) of positive floats giving the individual diversity values (or their geometric mean) of the row of the input `transition` matrix.

Examples

```
transition <- matrix (c (1/3, 1/3, 1/3, 0.9, 0.1, 0), nrow=2, ncol=3, byrow=TRUE)

get_diversity_from_transition (transition, type='individual', order=c(0,Inf), measure='entropy')
get_diversity_from_transition (transition, type='mean')
get_diversity_from_transition (transition, type='mean', mean_distribution=c(1/4,3/4))
```

`get_multipartite` *Build a properly-structured multipartite graph from raw data.*

Description

`get_multipartite` builds a properly-structured multipartite graph from a file or from a `data.frame` containing raw data. This object can then be used by the other functions of this package. The structure of the input data and of the resulting structure is detailed below.

Usage

```
get_multipartite(filename = NULL, data = NULL)
```

Arguments

`filename`	A character string giving the path to the file containing the raw data. The input file should contain at least four columns, separated by spaces. Each row gives a link between two nodes belonging to two different parts of the multipartite graph. The first column gives the label of the part of the first node and the second column gives the label of this node (any character string). Similarly, the third and fourth columns give the labels of the second part and of the second node. The file can eventually contain a fifth column giving the weights of the links (any positive integer or float value).
`data`	A `data.frame` containing the raw data. This `data.frame` should have the same structure than the one described above for the case of an input file: four columns indicating the labels of the parts and of the nodes constituting the link, and an optional fifth column for its weight. At least `filename` or `data` should be specified, but not both at the same time.

Value

A properly-structured multipartite graph that can be used by the other functions of the `triversity` package.

The resulting object encodes the labels of the parts and of the nodes in the multipartite graph, as well as the transition matrices of random walks following different paths between parts (encoded as sparse matrices of floats in `[0,1]`, with all rows summing to `1`). These transition matrices are then used by functions such as `get_distribution_from_path` to compute the probability distributions of such random walks, or such as `get_diversity_from_path` to compute the diversity of these distributions.

Assuming the object returned by `get_multipartite` is stored in the `graph` variable, then:

- `graph$parts` is a vector of character strings giving the labels of the different parts in the multipartite graph;
- `graph$nodes` is a list of vectors of character strings giving the labels of the nodes constituting the different parts of the multipartite graph;
- `graph$edges` is a `data.tree` whose nodes each contains the transition matrix of the corresponding random walk.

See Also

Package `slam` for the sparse encoding of transition matrices and package `data.tree` for the tree structure storing these matrices.

Examples

```
data (tripartite_example)
graph <- get_multipartite (data=tripartite_example)

graph$parts

part1 <- graph$parts[1]
graph$nodes[[part1]]
```

```
part2 <- graph$parts[2]
as.matrix (graph$edges$Climb(part1,part2)$matrix)
```

---

get_transition_from_path

Compute the transition matrix of a random walk following a path between the parts of a multipartite graph.

Description

`get_transition_from_path` computes the transition matrix of a random walk following a `path` between the different parts of the input multipartite `graph`.

Usage

```
get_transition_from_path(graph, path)
```

Arguments

`graph`	A multipartite graph obtained by calling the `get_multipartite` function.
`path`	A vector of character strings giving the path that the random walk should follow between the different parts of the input multipartite `graph`. This path can be as long as wanted, with eventual cycles, and each string it contains should refer to a label in `graph$parts`.

Details

Note that the multipartite graph structure implemented in this package stores in memory any computed transition matrix to avoid redundant computation in the future. Hence, the latter execution of `get_transition_from_path`, or of any other function that builds on it, can be much faster than the first call. The transition matrices are stored within a `data.tree` in the input `graph` variable (see `graph$edges`).

Value

A matrix of floats in `[0,1]`, with all lines summing to `1`, giving the transition matrix of the random walk following the input `path`.

Examples

```
data (tripartite_example)
graph <- get_multipartite (data=tripartite_example)

as.matrix (get_transition_from_path (graph, path=c(2,1,2,3)))
```

`tripartite_example` *An example of dataframe encoding a small tripartite graph.*

Description

`tripartite_example` is a `data.frame` containing raw data that encodes a small tripartite graph. It has the proper format to be loaded by `get_multipartite`. It contains four columns. Each row gives a link between two nodes belonging to two different parts of the tripartite graph. The first column gives the label if the part of the first node and the second column gives the label of this node (any character string). Similarly, the third and fourth columns give the labels of the second part and of the second node. A fifth column could eventually be added to give the weights of the links (any positive integer or float value).

Usage

```
data(tripartite_example)
```

Format

An object of class `data.frame` with 11 rows and 4 columns.

Examples

```
data (tripartite_example)
head (tripartite_example)

graph <- get_multipartite (data=tripartite_example)
graph
```

`triversity` *Compute diversity measures on multipartite graphs.*

Description

`triversity` is an R package for the computation of diversity measures on multipartite graphs. First, it implements a parametrized family of such diversity measures which apply on probability distributions. Sometimes called "True Diversity", this family contains famous measures such as the richness, the Shannon entropy, the Herfindahl-Hirschman index, and the Berger-Parker index. Second, the package allows to apply these measures on probability distributions resulting from random walks between the parts of multipartite graphs. By defining an initial distribution on a given part of the graph and a path to follow between the different parts, the probability of the walker's position within the final part is then computed, thus providing a particular instance of diversity to measure.

This package has been developed by researchers of the Complex Networks team, located within the Computer Science Laboratory of Paris 6 (LIP6), for the AlgoDiv project founded by the French National Agency of Research (ANR) under grant ANR-15-CE38-0001.

Links:

- AlgoDiv project: `http://algodiv.huma-num.fr/`
- Complex Networks team: `http://www.complexnetworks.fr/`
- LIP6: `https://www.lip6.fr/`
- ANR: `http://www.agence-nationale-recherche.fr/`

Contact:

- Robin Lamarche-Perrin: `<Robin.Lamarche-Perrin@lip6.fr>`
  See also my webpage: `https://www-complexnetworks.lip6.fr/~lamarche/`

List of main collaborators:

- Robin Lamarche-Perrin (CNRS, ISC-PIF, LIP6)
- Lionel Tabourier (UPMC, LIP6)
- Fabien Tarissan (CNRS, ISP, LIP6)
- Raphaël Fournier S'niehotta (CNAM, CÉDRIC)
- Rémy Cazabet (UdL, LIRIS)

Index

∗ **datasets**
    tripartite_example, 10

get_all_diversities, 2
get_distribution_from_path, 3, *8*
get_diversity_from_distribution, 4
get_diversity_from_path, *2*, 5, *8*
get_diversity_from_transition, 6
get_multipartite, *2, 3*, *5*, 7, *9*, *10*
get_transition_from_path, 9

tripartite_example, 10
triversity, 10
triversity-package *(*triversity*)*, 10