triversity

October 4, 2017

`get_conditional_diversity_from_distribution`
Get the conditional diversity values associated to a probability distribution and a transition matrix.

Usage

```
get_conditional_diversity_from_distribution(distribution, transition,
  order = NULL, measure = NULL)
```

Arguments

`distribution`	The probability distribution to measure.
`transition`	The probability transition matrix to use.
`order`	A vector of positive floats (possibly including `Inf`) describing the orders of the diversity measures to be computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. The following possible values are `'richness'`, `'entropy'`, `'herfindahl'`, and `'bergerparker'`.

Value

The diversity of the input distribution.

---

get_distribution_from_path

Get the probability distribution corresponding to a path in the tripartite graph

---

Usage

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

Arguments

`tripartite`	A tripartite graph obtained by the `get_tripartite`.
`path`	The path to follow in the graph to compute the distribution.
`initial_distribution`	A vector of floats describing the initial distribution to start with at the first level of the path. It should sum to 1 and contains as many values as there are nodes in the graph's level specified by the first step of the path.
`initial_node`	The name of a node in the first level of the path.

Value

The resulting probability distribution, that is a vector of floats which sum to 1.

---

get_diversity_from_distribution

Get the diversity value associated to a probability distribution.

Usage

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

Arguments

`distribution`	A vector of floats describing the probability distribution to measure.
`order`	A vector of positive floats (possibly including `Inf`) describing the orders of the diversity measures to be computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. The following possible values are `'richness'`, `'entropy'`, `'herfindahl'`, and `'bergerparker'`.

Value

A vector containing the diversity values of the input distribution.

`get_diversity_from_path` *Get diversity values associated to the distribution obtained through a path in a tripartite graph.*

Usage

```
get_diversity_from_path(tripartite, path, conditional_path = NULL,
  initial_distribution = NULL, initial_node = NULL, order = NULL,
  measure = NULL)
```

Arguments

`tripartite`	A tripartite graph obtained by the `get_tripartite`.
`path`	The path to follow to build the probability distribution
`conditional_path`	A vector of integers (among `1, 2, 3`) giving an eventual path to follow before computing and aggregating the diversity measures.
`initial_distribution`	A vector of floats describing the initial distribution to start with at the first level of the path. It should sum to 1 and contains as many values as there are nodes in the graph's level specified by the first step of the path.
`initial_node`	The name of a node in the first level of the path.
`order`	A vector of positive floats (possibly including `Inf`) describing the orders of the diversity measures to be computed.
`measure`	A vector of strings giving the names of the diversity measures to compute. The following possible values are `'richness'`, `'entropy'`, `'herfindahl'`, and `'bergerparker'`.

Value

The diversity of the resulting probability distribution

`get_transition_from_path` *Get the probability transition matrix corresponding to a path in a tripartite graph.*

Usage

```
get_transition_from_path(tripartite, path)
```

Arguments

`tripartite`	A tripartite graph obtained by `get_tripartite`.
`path`	The path to follow in the graph to compute the transition matrix.

Details

Note that the tripartite graph structure implemented in this package store in memory any computed transition matrix to avoid redundant computation in the future. Hence, the first execution of `get_transition_from_path` (or any other function that is built on it), can be much slower than latter calls. The transition matrices are stored in a `data.tree` (see `tripartite$transitions`).

Value

The resulting transition matrix, that is a matrix of floats which rows sum to one.

`get_tripartite` *Get a properly-structured tripartite graph from raw data.*

Description

`get_tripartite` returns a properly-structured tripartite graph from a file or from a dataframe. The structure of the input data and of the resultung data structure is detailed below.

Usage

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

Arguments

`filename`	The path to the file containing raw data to build the tripartite graph. This input file should have at least four columns, separated by spaces. Each row describe a link between two nodes belonging to two different levels of the tripartite graph. The first column indicates the level of the first node (any integer among `1`, `2`, or `3`) and the second column indicates its name (any character string). Similarly, the third and fourth columns indicate the level and the name of the second node. A fifth column can be added to indicate the eventual weights of the links (any integer or float value).
`data`	A `data.frame` containing the raw data to build the tripartite graph. This `data.frame` should have the same structure than the one described above when using an input file: four columns indicating (in the same order) the levels and the names of the two nodes constituting the link, and an optional fifth column fot its weight.

Value

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

The resulting data structure describe the different levels of the tripartite graph, as well as its transition probabilities (encoded as sparse float matrices) each following a given paths from one level to another. These transition matrices are then used by functions such as `get_distribution_from_path` to compute a distribution from a given path traveling through the different levels of the graph.

Once the `value` returned by `get_tripartite`:

- `value$nodes` is a list of vectors contraining the names of the nodes constituting the three levels of the tripartite graph (resp. `value$nodes$level1`, `value$nodes$level2`, and `value$nodes$level3`).
- `value$transitions` is a `data.tree` which nodes each contains a transition matrix. For example, `value$transitions$level1$level2$mat` is the transition matrix from level 1 to level 2.

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

Description

`triversity` is an R package for the computation of diversity measures on tripartite graphs. It first implements a parametrised family of such diversity measures applying 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. This package then allows to apply such measures on probability distributions resulting from random walks on tripartite graphs. By defining an initial distribution at a given level in the graph, a path within the three levels, and eventually a conditional step, the probability of the walker's position within the final level is then computed, thus providing a particular instance of diversity index.

Details

This package has been developed by researchers of the Complex Networks team, (http://www.complexnetworks.fr/) within the Computer Science Laboratory of Paris 6 (https://www.lip6.fr/), for the AlgoDiv project (http://algodiv.huma-num.fr/), founded by the French National Agency of Research (http://www.agence-nationale-recherche.fr/) under grant ANR-15-CE38-0001.

Contact: Robin Lamarche-Perrin <Robin.Lamarche-Perrin@lip6.fr>

See also my webpage: https://www-complexnetworks.lip6.fr/~lamarche/

List of main collaborators:

- Lionel Tabourier
- Fabien Tarissan
- Rapha\"el Fournier S'niehotta
- R\'emy Cazabet

Index