

Package ‘NetFrac’

October 15, 2019

Type Package

Title Netfrac - Community distance calculation within sequence similarity networks

Version 0.1.0

Author Xing, Henry; Kembel, Steven W.; Makarenkov, Vladimir

Maintainer Henry Xing <xing.henry@courrier.uqam.ca>

Description The NetFrac package allows the users to calculate the eight following network distances, NetUniFrac, Motifs, Spp, Spep, Spelp, Spinp, and the Transfer distance (two variants), between species communities present in a given sequence similarity network. The shortest path in a phylogenetic tree is traditionally used to estimate evolutionary distances between species. Moreover, the shortest path concept can be extended to sequence similarity networks. The five following distances (NetUniFrac, Spp, Spep, Spelp, and Spinp) can be seen as network analogues of the traditional UniFrac distance used to assess distances between species communities in a phylogenetic tree, whereas the Transfer distance is designed for estimating the rate and direction of gene transfers, or species dispersal, between different phylogenetic, or ecological, species communities.

Library dependencies:

library(igraph)

library(plyr)

library(MLmetrics)

library(doParallel)

library(SDDE)

library(ape)

library(netresponse)

library(Rcpp)

License GNU General Public License

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

R topics documented:

dist_paths	2
multicore	2
NetFrac	3
NetUnifrac	4
reconnect	5

reconnect_btw	5
set_color	6
subgroup_graph	6
Transfer	7

Index	8
--------------	----------

dist_paths	<i>Shortest path distances</i>
------------	--------------------------------

Description

Takes as input the network as well as the species communities to be compared and gives as output the distance matrix between species communities. It is better to use the NetFrac function if the global distances are needed.

Usage

```
dist_paths(x, col1, col2, distance = "paths", paths = "single", info = NULL, type = "graph", maxcores
```

Arguments

x	The igraph object to be analyzed
col1	Community one
col2	Community two
distance	Distances that are implemented in the NetFrac. See also the NetFrac function
paths	Take into account all possible shortest paths or only one
info	Information about the tree if the input is a tree
type	Tree or network
maxcores	Number of parallel cores to be used for calculation (one by default)

See Also

[NetFrac](#)

multicore	<i>multicore</i>
-----------	------------------

Description

Uses parallel cores for bigger networks.

Usage

```
multicore(nc = 1)
```

Arguments

nc	number of cores
----	-----------------

Description

Uses the `dist_paths` function to calculate distances between different species communities present in a sequence similarity network. It takes as input an `igraph` object and returns as output distance matrices between species communities.

Usage

```
NetFrac(x, distances = "Unifrac", paths = "single", maxcores = 1)
```

Arguments

<code>x</code>	The network(or tree) that will be analyzed. The network must be in the <code>igraph</code> format, the edges being accessible with <code>E(x)</code> and vertices with <code>V(x)</code> . The communities are under <code>V(x)\$tax</code> and the branch weights under <code>E(x)\$weight</code> . These functions are applicable with <code>igraph</code> objects only.
<code>distances</code>	Distance(s) to calculate: Transfer, Transfer2, Spaths, UniFrac and Motifs. The transfer, paths and Unifrac distances can also be calculated for trees.
<code>paths</code>	this parameter is used to decide whether all the shortest paths between network nodes should be calculated ("all"), or only one of them ("single"). The last option can significantly reduce the time of computation. For Motifs distance, decide the size of the motifs (i.e. 2 or 3).
<code>maxcores</code>	uses the <code>multicore</code> function to set up the number of cores to be used

Details

The Transfer and Transfer 2 distance are based on the number of paths that are not unique to one community. The Spaths distances are different shortest path distances that use different calculation parameters, namely Spp (Shortest path proportion), Spép (Shortest path edge proportion), Spelp (Shortest path edge length proportion) and Spinp (Shortest path internal node proportion). NetUniFrac is a proportion of edges whose nodes belong to the same species community, and Motifs on the motifs present in a network.

Examples

```
net_a = load_network("exemples/network_a.txt", "exemples/network_a_tax.txt")
NetFrac(net_a)
NetFrac(net_a, "paths", "all")
```

Output:

```
$`Spp`
      A      B
A 0.000000 0.555556
B 0.555556 0.000000
```

```
$Spép
      A      B
A 0.000000 0.6296296
```

```
B 0.6296296 0.0000000
```

```
$Spelp
```

```
      A      B
A 0.0000000 0.6296296
B 0.6296296 0.0000000
```

```
$Spinp
```

```
      A      B
A 0.0000000 0.9907407
B 0.9907407 0.0000000
```

```
$Transfer
```

```
      A      B
A 0.0000000 0.6666667
B 0.3333333 0.0000000
```

NetUnifrac	<i>NetUnifrac</i>
------------	-------------------

Description

Calculates the network version of the UniFrac distance.

Usage

```
NetUnifrac(igraph, tax1 = "", tax2 = "", weight = TRUE)
```

Arguments

igraph	The igraph object
tax1	The first community
tax2	The second community
weight	Use weights of the edges if existing

Examples

```
net_a = load_network("examples/network_a.txt", "examples/network_a_tax.txt")
NetUnifrac(net_a)
```

reconnect	<i>Connect the network using a similarity matrix</i>
-----------	--

Description

Creates a fully connected network using a similarity matrix between nodes.

Usage

```
reconnect(graph, matrice = "")
```

Arguments

graph	The disconnected igraph object
matrice	The similarity matrix containing pairwise distances between the nodes

See Also

[reconnect_btw](#)

reconnect_btw	<i>Connect the network a graph using betweenness centrality</i>
---------------	---

Description

Creates a fully connected network using the betweenness centrality principle.

Usage

```
reconnect_btw(graph)
```

Arguments

graph	The disconnected igraph object
-------	--------------------------------

See Also

[reconnect](#)

set_color	<i>Set colors for the communities in the network</i>
-----------	--

Description

Visualizes the network by adding different colors to different species communities, when plotting either with R own color list or a provided color list.

Usage

```
set_color(graph, colors_list = "")
```

Arguments

graph	The igraph object used to define colors. The nodes accessed with V() should have a \$tax attribute which links them to a community, so a different color can be assigned to each community.
colors_list	Optional; the colors can be defined in a list with the community names defined with names(). Otherwise, the function will return the numbers already associated to R colors.

Examples

```
net_a = load_network("exemples/network_a.txt", "exemples/network_a_tax.txt")
net_a = set_color(net_a)
plot(net_a)
```

subgroup_graph	<i>Create a subgraph based on chosen species communities</i>
----------------	--

Description

Creates another igraph object containing only the nodes from some chosen species communities (two or more), and the edges connecting them.

Usage

```
subgroup_graph(graph, groups)
```

Arguments

graph	The igraph object
groups	The communities that should be isolated, indicated as a list of two items

Examples

```
CAT_env = load_network("exemples/CAT_edges_env.csv", "exemples/CAT_node_env.csv")
CAT_env1 = set_color(subgroup_graph(CAT_env, c("host", "ubiquitous")))
plot(CAT_env1)
```

Transfer*Algorithm for a fast calculation of the Transfer distance*

Description

The transfer distance can be used to calculate the proportion of species/sequences of community X that have been affected by horizontal gene transfers from species/sequences of community Y. The Transfer distance is assymmetric. If the threshold α ($0 < \alpha < 1$) is satisfied, the distance is incremented. The operation is repeated for every node of the network. The result is a pairwise matrix reporting the Transfer distance between each pair of species communities.

Usage

```
Transfer(graph)
```

Arguments

graph	The igraph object
-------	-------------------

Examples

```
net_a = load_network("examples/network_a.txt", "examples/network_a_tax.txt")
transfer2(net_a)
```

Index

***Topic \textasciitildekw1**

dist_paths, [2](#)

multicore, [2](#)

NetFrac, [3](#)

***Topic \textasciitildekw2**

dist_paths, [2](#)

multicore, [2](#)

NetFrac, [3](#)

dist_paths, [2](#)

multicore, [2](#)

NetFrac, [2](#), [3](#)

NetUnifrac, [4](#)

reconnect, [5](#), [5](#)

reconnect_btw, [5](#), [5](#)

set_color, [6](#)

subgroup_graph, [6](#)

Transfer, [7](#)