Package ‘dendroNetwork’

July 25, 2025

Type Package

Title Create Networks of Dendrochronological Series using Pairwise Similarity

Version 0.5.5

Maintainer Ronald Visser <`r.m.visser@saxion.nl`>

Depends R (>= 4.1.0)

Imports dplR (>= 1.7.2), igraph, stringr, reshape2, RCy3, dplyr, RColorBrewer, tidyr, foreach, lifecycle, doParallel, stats, grDevices

Description Creating dendrochronological networks based on the similarity between tree-ring series or chronologies. The package includes various functions to compare tree-ring curves building upon the 'dplR' package. The networks can be used to visualise and understand the relations between tree-ring curves. These networks are also very useful to estimate the provenance of wood as described in Visser (2021) <DOI:10.5334/jcaa.79> or wood-use within a structure/context/site as described in Visser and Vorst (2022) <DOI:10.1163/27723194-bja10014>.

License GPL (>= 3)

biocViews Visualization, GraphAndNetwork, ThirdPartyClient, Network

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), DiagrammeR

Config/testthat/edition 3

VignetteBuilder knitr

URL `https://github.com/ropensci/dendroNetwork`, `https://docs.ropensci.org/dendroNetwork/`

BugReports `https://github.com/ropensci/dendroNetwork/issues`

NeedsCompilation no

Author Ronald Visser [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6966-1729>),
Angelino Salentino [ctb] (ORCID:
<https://orcid.org/0000-0002-4763-3943>),
Andy Bunn [ctb] (ORCID: <https://orcid.org/0000-0001-9027-2162>),
Kaija Gahm [rev] (Kaija Gahm reviewed the package for rOpenSci, see
https://github.com/ropensci/software-review/issues/627#issuecomment-1993235710),
Zachary Gajewski [rev] (Zachary Gajewski reviewed the package for
rOpenSci, see
https://github.com/ropensci/software-review/issues/627#issuecomment-2037644081)

Repository CRAN

Date/Publication 2025-07-25 15:20:02 UTC

Contents

`clique_community_names` *Clique Percolation Method (with node names)*

Description

Function to determine communities in a network using clique percolation method (Palla et al., 2005). Communities are created based on cliques. Cliques are subsets of a network that can be considered complete (sub)networks. The size of the cliques to be used to community detection is part of the input of the function.

Usage

```
clique_community_names(g, k = 3)
```

Arguments

`g`	network object (igraph)
`k`	clique size to be used. The default is set to smallest possible size (3)

Value

a dataframe with node names and community name. The community is named as CPM_Kk_number_of_community with k replaced by the value of k.

Author(s)

Angelo Salatino

Ronald Visser

References

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043), 814-818.

Code adapted from source: https://github.com/angelosalatino/CliquePercolationMethod-R/blob/master/clique.community.R

Examples

```
hol_sim <- sim_table(hol_rom)
g_hol <- dendro_network(hol_sim)
clique_community_names(g_hol, k = 3)
```

`clique_community_names_par`

Clique Percolation Method (with node names)

Description

Function to determine communities in a network using clique percolation method (Palla et al., 2005). Communities are created based on cliques. Cliques are subsets of a network that can be considered complete (sub)networks. The size of the cliques to be used to community detection is part of the input of the function. This function uses parallelisation and should be used for larger networks.

Usage

```
clique_community_names_par(g, k = 3, n_core = 4)
```

Arguments

`g`	network object (igraph)
`k`	clique size to be used, default set to smallest possible size (3)
`n_core`	number of cores to be used for parrallisation, defaults to 4

Value

a dataframe with node names and community name. The community is named as CPM_Kk_number_of_community with k replaced by the value of k.

Author(s)

Angelo Salatino

Ronald Visser

References

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. Nature, 435(7043), 814-818.

Code adapted from source: https://github.com/angelosalatino/CliquePercolationMethod-R/blob/master/clique.community.op

Examples

```
## Not run:
hol_sim <- sim_table(hol_rom)
g_hol <- dendro_network(hol_sim, r_threshold = 0.4, sgc_threshold = 0.4)
clique_community_names_par(g_hol, k = 3, n_core = 2)

## End(Not run)
```

`cor_mat_overlap` *Pearson correlation matrix*

Description

Function that creates a Pearson correlation matrix of two rwl objects compared. If the same rwl-object is passed to the function, the correlation between all series is calculated. In addition, the number of overlapping tree-rings is part of the output. The results can be used to calculate the Students' t value.

Usage

```
cor_mat_overlap(x, y, minoverlap)
```

Arguments

`x`	rwl object of tree-ring series
`y`	rwl object of tree-ring series
`minoverlap`	the correlation will only be calculated if the number of overlapping tree-rings is equal or larger than this value

Details

The function is an adaptation of the function cor.with.limit.R() from https://github.com/AndyBunn/dplR/blob/master/R/rwi.st
The new function is optimized and also outputs the number of overlapping rings.

Value

a list with two matrices: one with the correlation values and one with the number of overlapping tree rings for each correlation value. The matrices have row names and column names of the compared tree-ring curves

Author(s)

Andy Bunn

Ronald Visser

Examples

```
rwl_object1 <- hol_rom
rwl_object2 <- hol_rom
cor_mat_overlap(rwl_object1, rwl_object2, 50)
cor_mat_overlap(rwl_object1, rwl_object1, 50)
```

`cyto_clean_styles` *Cleaning nearly all styles in cytoscape session and import simple styles.*

Description

Cleaning nearly all styles in cytoscape session and import simple styles.

Usage

```
cyto_clean_styles()
```

Value

Cytoscape cleaned of styles and only styles with white and grey nodes.

Examples

```
## Not run:
cyto_clean_styles()

## End(Not run)
```

`cyto_create_cpm_style` *Create CPM style in Cytoscape*

Description

Function to create a style in Cytoscape to visualise the communities in a network using clique percolation method (CPM: Palla et al., 2005). See also: find_all_cpm_com() Each node is filled with the colour of the community. If a node is part of several communities a pie chart is used to show the various community colours. The function uses a graph as input and the number of cliques (default = 3). The style can be specified or automatically named based on the name of the network an the number of cliques. Before starting this function, Cytoscape must be up and running!

Usage

```
cyto_create_cpm_style(graph_input, k = 3, com_k = NULL, style_name = "auto")
```

Arguments

`graph_input`	the graph with the CPM communities
`k`	clique size for the visualisation. This should be an integer with the value 3 or higher
`com_k`	data_frame with the communities for the specific clique size (two columns: node and com_name). This is the result of clique_community_names_par() or clique_community_names()
`style_name`	name of the output style in Cytoscape. If set to "auto", the style is derived from the name of the network and value for k

Value

The style applied in Cytoscape, no objects in R as return.

Examples

```
## Not run:
data(hol_rom)
sim_table_hol <- sim_table(hol_rom)
g_hol <- dendro_network(sim_table_hol)
hol_com_cpm_k3 <- clique_community_names(g_hol, k = 3)
cyto_create_graph(g_hol)
cyto_create_cpm_style(g_hol, k = 3, com_k = hol_com_cpm_k3)

## End(Not run)
```

`cyto_create_gn_style` *Create Girvan-Newman communities style in Cytoscape*

Description

Function to create a style in Cytoscape to visualise the communities in a network using the Girvan-Newman method for community detection. Each node is filled with a separate colour for each community. Before starting this function, Cytoscape must be up and running!

Usage

```
cyto_create_gn_style(graph_input, gn_coms = NULL, style_name = "auto")
```

Arguments

`graph_input`	the graph with the CPM communities
`gn_coms`	GN communities in graph_input. This is the result of gn_names(). If this is not given this will be calculated in the function
`style_name`	name of the output style in Cytoscape. If set to "auto", the style is derived from the name of the network and value for k

Value

The style applied in Cytoscape, no objects in R as return.

Examples

```
## Not run:
data(hol_rom)
sim_table_hol <- sim_table(hol_rom)
g_hol <- dendro_network(sim_table_hol)
g_hol_gn <- gn_names(g_hol)
cyto_create_graph(g_hol)
cyto_create_gn_style(g_hol, gn_coms = g_hol_gn)

## End(Not run)
```

`cyto_create_graph` *Create networks in Cytoscape*

Description

Function to create a network in cytoscape (https://cytoscape.org/) Cytoscape must be running before executing this function.

Usage

```
cyto_create_graph(
  graph_input,
  network_name = substitute(graph_input),
  collection_name = "default",
  style_name = "default",
  CPM_table = NULL,
  GN_table = NULL
)
```

Arguments

`graph_input`	igraph network used to create network in Cytoscape
`network_name`	name of the network in Cytoscape, defaults to the name of variable that is the network in R
`collection_name`	name of the collection in Cytoscape (default = default)
`style_name`	name of the style in Cytoscape (default = default)
`CPM_table`	table with the name of the nodes in the first column and the CPM-communities in other columns. This is the result of find_all_cpm_com()
`GN_table`	two column table with the name of the nodes in the first column and the Girvan-Newman-communities in other columns

Value

a graph in Cytoscape

Examples

```
## Not run:
data(hol_rom)
sim_table_hol <- sim_table(hol_rom)
g_hol <- dendro_network(sim_table_hol)
hol_com_cpm_all <- find_all_cpm_com(g_hol)
g_hol_gn <- gn_names(g_hol)
cyto_create_graph(g_hol, CPM_table = hol_com_cpm_all, GN_table = g_hol_gn)

## End(Not run)
```

`dendro_network` *Create dendrochronologial networks*

Description

Function to create dendrochronological networks based on the similarity. The input for this function is a similarity table created with the sim_table() function. The thresholds are set to the defaults (Visser 2021).

Usage

```
dendro_network(
  sim_table,
  r_threshold = 0.5,
  sgc_threshold = 0.7,
  p_threshold = 1e-04,
  corr_type = "r_hol"
)
```

Arguments

`sim_table`	a table of similarities created by the sim_table() function
`r_threshold`	all correlations equal or above this value are taken into account for the creation of edges. If you want all positive correlations included set this to 0.
`sgc_threshold`	all sgc-values equal or above this value are taken into account for the creation of edges.
`p_threshold`	all probabilities of exceedence equal or below this value are taken into account for the creation of edges.
`corr_type`	choose between the correlation based on wuchswerte (r_hol: Hollstein 1980) or the correlation without extra transformation (r)

Value

A simplified network of tree-ring material with the edges defined by the similarity.

References

Visser, RM. 2021a Dendrochronological Provenance Patterns. Network Analysis of Tree-Ring Material Reveals Spatial and Economic Relations of Roman Timber in the Continental North-Western Provinces. Journal of Computer Applications in Archaeology 4(1): 230–253. DOI: https://doi.org/10.5334/jcaa.79.

Examples

```
data(hol_rom)
sim_table_hol <- sim_table(hol_rom)
g_hol <- dendro_network(sim_table_hol)
plot(g_hol)
g_hol_r_low <- dendro_network(sim_table_hol, r_threshold = 0.3,
                                sgc_threshold = 0.6, corr_type = "r")
plot(g_hol_r_low)
```

`find_all_cpm_com` *Finding all CPM communities in a network/graph Function to determine all CPM-communities in a network (or graph) using the clique percolation method (Palla et al 2005).*

Description

The CPM makes use of complete graphs or cliques in the network. A clique is a group of nodes were all of them are connected. A clique of 3 nodes has 3 edges, while a clique of 4 nodes has already 6 edges. Communities in CPM consist of a large group of connected cliques.

Usage

```
find_all_cpm_com(graph_input, n_core = 0)
```

Arguments

`graph_input`	the graph for find all CPM communities in
`n_core`	this defaults to 0, if a number larger than 1 is given the community detection is performed using parallel computing

Value

data frame with at least two columns. The first column are the node names and the further columns represent the CPM-communities, with 1 denoting the membership in a community.

References

Palla, G, Derenyi, I, Farkas, I and Vicsek, T. 2005 Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043): 814–818. DOI: https://doi.org/10.1038/nature03607.

Examples

```
data(hol_rom)
sim_table_hol <- sim_table(hol_rom)
g_hol <- dendro_network(sim_table_hol)
hol_com_cpm_all <- find_all_cpm_com(g_hol)
```

`gn_names` *Community detection using the Girvan-Newman algorithm*

Description

Function to determine the communities in a network using the Girvan-Newman algorithm. This function uses the cluster_edge_betweenness() function from the iGraph package, but creates a more user-friendly output that includes the names of the nodes.

Usage

```
gn_names(g)
```

Arguments

`g`	input graph or network that is used for community detection

Details

The GN-algorithm uses the edge betweenness to determine communities. Edge betweenness is the number of shortest paths between pairs of nodes running over that edge. The higher the edge betweenness, the more communities an edge connects. Therefore these are removed in this algorithm to detect communities.

Value

the names of the nodes in the various communities

References

Girvan, M and Newman, MEJ. 2002 Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99(12): 7821–7826. DOI: https://doi.org/10.1073/pnas.122653799.

Newman, MEJ and Girvan, M. 2004 Finding and evaluating community structure in networks. Physical Review E 69(2): 026113. DOI: https://doi.org/10.1103/PhysRevE.69.026113.

Examples

```
hol_sim <- sim_table(hol_rom)
g_hol <- dendro_network(hol_sim)
gn_names(g_hol)
```

`hol_rom` *Roman tree-ring site chronologies from Hollstein*

Description

Dendrochronological site chronologies published by Hollstein (1972, 1980). These 52 (pre) Roman site chronologies date between 502 BC and AD 550. This selection of (pre)Roman site-chronologies are also used by Visser (2021) and made available as Visser (2022).

Usage

```
data(hol_rom)
```

Format

An object of class `"rwl"`.

References

Hollstein, E. 1972 Dendrochronologische Datierung von Hölzern aus Wederath (Belginum). Trierer Zeitschrift 35: 123–125.

Hollstein, E. 1980. Mitteleuropäische Eichenchronologie. Trierer Dendrochronologische Forschungen zur Archäologie und Kunstgeschichte. Trierer Grabungen und Forschungen 11. Mainz am Rhein: Verlag Philipp von Zabern.

Visser, RM. 2021 Dendrochronological Provenance Patterns. Network Analysis of Tree-Ring Material Reveals Spatial and Economic Relations of Roman Timber in the Continental North-Western Provinces. Journal of Computer Applications in Archaeology 4(1): 230–253. DOI: https://doi.org/10.5334/jcaa.79.

Visser, RM. 2022 Dendrochronological Provenance Patterns. Code and Data of Network Analysis of Tree-Ring Material. DOI: https://doi.org/10.5281/zenodo.7157744.

Examples

```
data(hol_rom)
sim_table(hol_rom, last_digit_radius = TRUE)
```

`RING_Visser_2021` *Roman tree-ring site chronologies*

Description

Dendrochronological site chronologies published by Visser (2021, 2022). These (pre) Roman site chronologies date between 520 BC and AD 663 and are based on the material of RING (full references to the source data can be found in the supplementary data of Visser 2021) The series are named based on their location, species and type of standardisation For example: ABC_Q1M or ABC_Q1C consist of the same material from the site ABC (Abcoude), species Q(uercus), chronology 1 and standardisation C(ofecha) and M(eans). See Visser(2021) for more explanation.

Usage

```
data(RING_Visser_2021)
```

Format

An object of class `"rwl"`.

References

Visser, RM. 2021 Dendrochronological Provenance Patterns. Network Analysis of Tree-Ring Material Reveals Spatial and Economic Relations of Roman Timber in the Continental North-Western Provinces. Journal of Computer Applications in Archaeology 4(1): 230–253. DOI: https://doi.org/10.5334/jcaa.79.

Visser, RM. 2022 Dendrochronological Provenance Patterns. Code and Data of Network Analysis of Tree-Ring Material. DOI: https://doi.org/10.5281/zenodo.7157744.

Examples

```
data(RING_Visser_2021)
sim_table(RING_Visser_2021, last_digit_radius = TRUE)
```

`sim_table` *Similarity table of tree-ring curves for the creation of a network*

Description

Function to calculate various similarity measures for the creation of dendrochronological networks as described by Visser (2021a). The function results in a list of all similarities of all comparisons between the tree-ring series in trs1 (or between trs1 and trs2). The resulting list includes the overlap, correlation (both with and without Hollstein-transformation), the t-value based on these correlations, Synchronous Growth Changes (SGC), Semi Synchronous Growth Changes (SSGC), and the related probability of exceedence (p).The last three (SGC, SSGC and p) are explained in Visser (2021b).

Usage

```
sim_table(trs1,
          trs2=NULL,
          min_overlap=50,
          last_digit_radius=FALSE)
```

Arguments

`trs1`	Rwl object with first tree-ring series to be compared with trs2. A rwl object is a data.frame with series or tree-ring widths as columns and years as rows. This object is created or read by using the dplR-package
`trs2`	Optional second rwl object with second tree-ring series to be compared with trs1. Use this is you have two different datasets that you want to compare. Otherwise all series in trs1 are pair wise compared.
`min_overlap`	If the overlap of the compared series is longer or equal than this minimal value, the similarities will be calculated for the comparison
`last_digit_radius`	Set this to TRUE if the last digit of a series name is the radius of the tree-ring series

Value

The resulting list includes the names of the compared series, overlap, correlation (both with and without Hollstein-transformation), t-value based on these correlations, SGC, SSGC and the related probability of exceedence.

References

Visser, RM. 2021a Dendrochronological Provenance Patterns. Network Analysis of Tree-Ring Material Reveals Spatial and Economic Relations of Roman Timber in the Continental North-Western Provinces. Journal of Computer Applications in Archaeology 4(1): 230–253. DOI: https://doi.org/10.5334/jcaa.79.

Visser, RM. 2021b On the similarity of tree-ring patterns: Assessing the influence of semi-synchronous growth changes on the Gleichläufigkeitskoeffizient for big tree-ring data sets. Archaeometry 63(1): 204–215. DOI: https://doi.org/10.1111/arcm.12600.

Examples

```
data(hol_rom)
sim_table(hol_rom)
sim_table(hol_rom, last_digit_radius = TRUE)
sim_table(hol_rom, min_overlap = 25)
sim_table(hol_rom, min_overlap = 100, last_digit_radius = TRUE)
```

`t_value` *Students' t value*

Description

Function to determine the Students' t-value using the correlation and the number of compare values (tree-rings)

Usage

```
t_value(r, n)
```

Arguments

`r`	correlation value.
`n`	number of overlapping tree-rings/compared values.

Value

Students' t value as a numeric.

Examples

```
t_value(0.5, 100)
```

`wuchswerte` *Wuchswerte (Hollstein 1980)*

Description

Function to normalize tree-ring values according to Hollsteins transformation to Wuchswerte (Hollstein 1980, 14-15).

Usage

```
wuchswerte(x)
```

Arguments

`x`	tree-ring series

Value

tree-ring series normalized according to Hollstein (1980, 14-15)

References

Hollstein, E. 1980. Mitteleuropäische Eichenchronologie. Trierer Dendrochronologische Forschungen zur Archäologie und Kunstgeschichte. Trierer Grabungen und Forschungen 11. Mainz am Rhein: Verlag Philipp von Zabern.

Examples

```
data(hol_rom)
wuchswerte(hol_rom)
# to convert a rwl object into wuchwerte use:
dplR::as.rwl(apply(hol_rom, 2, wuchswerte))
```

Index