

Package ‘rDNA’

July 16, 2016

Version 1.31

Date 2016-07-16

Title R Bindings for the Discourse Network Analyzer

Author Philip Leifeld

Maintainer Philip Leifeld <philip.leifeld@glasgow.ac.uk>

Description Control the Java software Discourse Network Analyzer (DNA) from within R. Network matrices, statement frequency time series and attributes of actors can be transferred directly into R.

Depends R (>= 2.14), rJava (>= 0.5-0)

Suggests network, sna

SystemRequirements Java (>= 1.6)

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-07-16 21:00:07

R topics documented:

dna.attributes	2
dna.categories	3
dna.density	4
dna.gui	5
dna.init	6
dna.network	7
dna.timeseries	10
rDNA	12
Index	14

dna.attributes

Get actor attributes

Description

Transfer the attributes of persons or organizations to R.

Usage

```
dna.attributes(infile, organizations = TRUE, verbose = TRUE)
```

Arguments

infile	The input .dna file as a string (i.e., enclosed in quotation marks). If the file is not in the current working directory, specify the path together with the file name. Include the file suffix. Example: sample.dna.
organizations	If TRUE, the attributes of organizations will be returned. If FALSE, the attributes of persons will be returned.
verbose	If true, details about the data import and its progress will be printed. If false, these information will be suppressed.

Details

Pulls the attributes of persons or organizations as a matrix into R. The matrix has three columns: type (the type of actor), alias (the alias/description variable in DNA), note (the notes field) and color (the RGB color used in DNA).

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA](#) [dna.init](#) [dna.network](#) [dna.categories](#)

Examples

```
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/sample.dna",
  destfile = "sample.dna", mode = "wb")
dna.init("dna-1.31.jar")
attributes <- dna.attributes("sample.dna", organizations = TRUE)
```

dna.categories	<i>Get list of categories</i>
----------------	-------------------------------

Description

Transfer a vector of all categories in a DNA file to R.

Usage

```
dna.categories(infile, verbose=TRUE)
```

Arguments

infile	The input .dna file as a string (i.e., enclosed in quotation marks). If the file is not in the current working directory, specify the path together with the file name. Include the file suffix. Example: sample.dna.
verbose	If true, details about the data import and its progress will be printed. If false, these information will be suppressed.

Details

Puts all categories found in a .dna file in a list and imports them as a vector into R.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA dna.init dna.network dna.attributes](#)

Examples

```
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/sample.dna",
  destfile = "sample.dna", mode = "wb")
dna.init("dna-1.31.jar")
categories <- dna.categories("sample.dna")
```

dna.density	<i>Compute density for matrices</i>
-------------	-------------------------------------

Description

Compute the within- and between-block density for weighted or binary network matrices.

Usage

```
dna.density(network.matrix, partitions = "", weighted = FALSE,
            verbose = FALSE)
```

Arguments

network.matrix	A co-occurrence or adjacency network which is saved as a matrix object. It should have the correct row and column labels assigned, otherwise partitions will not work.
partitions	An optional argument where the affiliation of row and column vertices to groups can be specified. The vertex types or groups of the row labels in the network.matrix can be submitted as a one-column matrix or data.frame with meaningful row names. The row and column names of the network.matrix are looked up in the partitions if specified. Alternatively, partitions can be given as a vector (see example). If the partitions do not have row names, the row names of the network matrix will be assumed. The output will be a group x group density table for the different partitions.
weighted	A boolean variable indicating whether the weight of an edge should be counted when the density is calculated.
verbose	A boolean variable indicating whether the results shall also be printed to the screen.

Details

This method computes the density of network matrices. It works for weighted or binary networks, and it can return within- and between-block density if actor types or the affiliation of actors to theoretical groups is given. For example, in a network of political actors, each actor may have a certain actor type: interest group, governmental actor, party etc. If the type of each row and column actor can be looked up in the one-column matrix given by the optional partitions argument, the result of the dna.density method will be a type x type matrix containing the densities within each actor type and between the various actor types. Within- and between block density for theoretical blocks is elsewhere referred to as node-mixing.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[dna.attributes](#) [dna.network](#)

Examples

```
# download files and initialize DNA:
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/sample.dna",
  destfile = "sample.dna", mode = "wb")
dna.init("dna-1.31.jar")

# create network matrix
nw <- dna.network("sample.dna")

# compute (binary) density (without partitions)
dna.density(nw)

# create vector with groups (assuming they correspond to rows in the matrix)
groups <- c("Con", "Con", "Con", "Pro", "Pro", "Pro", "Con")

# compute (weighted) density by partition
dna.density(nw, partitions = groups, weighted = TRUE)
```

 dna.gui

Start DNA GUI

Description

Start the GUI of the Java software Discourse Network Analyzer.

Usage

```
dna.gui(memory = 1024)
```

Arguments

memory	The amount of memory to allocate for DNA. By default, 1024 MB will be allocated. This should be enough for most .dna files. If your computer is low on memory, you may set it to 512 or 256. If your file is very large and your computer has enough RAM, you can allocate 2048 MB or more.
--------	---

Details

The method starts the user interface of the Java software Discourse Network Analyzer (DNA). All changes in the GUI have to be saved manually and will not be transferred back to R.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA](#) [dna.init](#) [dna.network](#) [dna.attributes](#)

Examples

```
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
dna.init("dna-1.31.jar")
## Not run:
dna.gui()

## End(Not run)
```

dna.init

Initialize rDNA

Description

Establish the connection between R and DNA.

Usage

```
dna.init(dna.jar.file)
```

Arguments

`dna.jar.file` The path and file name of the dna-1.31.jar file. If DNA is located in the current working directory, this can be "dna-1.31.jar".

Details

This method must be called once before rDNA can pull any data from DNA. Hand over the path and file name of the dna-1.31.jar file to the method.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA](#) [dna.gui](#) [dna.network](#) [dna.attributes](#)

Examples

```
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
dna.init("dna-1.31.jar")
```

dna.network

Get network data

Description

Convert a .dna file into a sociomatrix and import it into R.

Usage

```
dna.network(infile, algorithm = "cooccurrence",
  agreement = "combined", start.date = "01.01.1900",
  stop.date = "31.12.2099", two.mode.type = "oc",
  one.mode.type = "organizations", via = "categories",
  ignore.duplicates = TRUE, include.isolates = FALSE,
  normalization = FALSE, window.size = 100, step.size = 1,
  alpha = 100, lambda = 0.1, ignore.agreement = FALSE,
  exclude.persons = c(""), exclude.organizations = c(""),
  exclude.categories = c(""), invert.persons = FALSE,
  invert.organizations = FALSE, invert.categories = FALSE,
  verbose = TRUE)
```

Arguments

infile	The input .dna file as a string (i.e., enclosed in quotation marks). If the file is not in the current working directory, specify the path together with the file name. Include the file suffix. Example: sample.dna.
algorithm	The algorithm which should be used to create the network. Refer to the DNA manual at http://www.philipleifeld.de for details. Possible values are: affiliation (for a two-mode network of actors and concepts), cooccurrence (for an actor or concept co-occurrence/one-mode network), timewindow (for the time window algorithm) and attenuation (for the attenuation algorithm).
agreement	The agreement pattern to be used. Must be one of the following: yes, no, combined or conflict.
start.date	Only statements after this date will be retained. The start date is a character string of the form dd.mm.yyyy, where dd is the two-digit day, mm the two-digit month and yyyy the four-digit year.
stop.date	Only statements before this date will be retained. The stop date is a character string of the form dd.mm.yyyy, where dd is the two-digit day, mm the two-digit month and yyyy the four-digit year.

<code>two.mode.type</code>	If the affiliation algorithm is selected, this argument determines the vertex classes to be used. The following values are possible: <code>oc</code> (which stands for organizations x categories), <code>pc</code> (persons x categories) and <code>po</code> (persons x organizations).
<code>one.mode.type</code>	If the cooccurrence algorithm, the timewindow algorithm or the attenuation algorithm is selected, this argument specifies the vertex class to be used. The following values are possible: <code>organizations</code> (which stands for organizations x organizations), <code>persons</code> (persons x persons) or <code>categories</code> (categories x categories).
<code>via</code>	If the <code>one.mode.type</code> argument is active (i.e., the cooccurrence, timewindow or attenuation algorithm is used), this argument specifies via which variable a co-occurrence network is created. For example, if an organizations x organizations network is created, organizations can either be connected via their shared persons or categories. Valid values are thus <code>organizations</code> , <code>persons</code> and <code>categories</code> , but not the vertex type used in the <code>one.mode.type</code> argument.
<code>ignore.duplicates</code>	A boolean variable indicating whether two statements with the same actor, category, agreement pattern and date should be counted separately during network creation. For example, if a speaker re-iterates the same concepts in the same way over and over again in the same article, each of these statements increases the edge weight between this speaker and other speakers using the same argument if <code>ignore.duplicates</code> is switched off (i.e., set to <code>FALSE</code>).
<code>include.isolates</code>	If several time slices are exported, usually the network matrices will have different dimensions. If the <code>include.isolates</code> argument is set to <code>TRUE</code> , all actors - even if they are inactive in the current time slice - are included in the matrix. This guarantees that several time slices have the same dimensions and the same order of actors.
<code>normalization</code>	Some actors make statements more frequently than others, and this behavior is caused by their institutional position. These actors are likely to be at the center of the network. If <code>normalization</code> is set to <code>TRUE</code> , DNA tries to correct for institutional positions by dividing edge weights by the average total number of statements of both actors involved in an edge. For more details, please refer to the DNA manual at http://www.philipleifeld.de .
<code>window.size</code>	If the timewindow algorithm is used, the <code>window.size</code> argument controls the size of the time window. Integer values are possible. Recommended values are somewhere between 10 and 2000 days, depending on the theory and the dataset.
<code>step.size</code>	If the timewindow algorithm is used, the <code>step.size</code> argument controls the rate at which the time window moves, i.e., the number of days by which the window is moved at each step. Using 1 day is recommended. For non-overlapping time windows, use the same value as in the <code>window.size</code> argument.
<code>alpha</code>	If the timewindow algorithm is used and <code>normalization=TRUE</code> , the <code>step.size</code> argument provides a constant by which edge values are multiplied. This is useful because normalized edge weights in the time window algorithm may become quite small.

lambda	If the attenuation algorithm is used, lambda provides the decay constant for the exponential decay function. The default value of 0.1 attributes relatively high weight to statements which are made within approximately five to ten days.
ignore.agreement	This argument is only used if algorithm="attenuation" is set. When using the attenuation algorithm, ignore.agreement=TRUE specifies that the agreement variable should be ignored completely. For example, if the initial statement is positive and another actor uses the same concept in a negative way, an edge is established nevertheless – even if agreement="combined" is set, which would normally distinguish between positive and negative relations and add them up). The partial edge value is subject to the exponential decay function using constant lambda.
exclude.persons	Specify a list of persons to be excluded from the network. For example, c("person 1", "person 2"). Note that the names must appear exactly as they are used on the dataset.
exclude.organizations	Specify a list of organizations to be excluded from the network. For example, c("organization 1", "organization 2"). Note that the names must appear exactly as they are used on the dataset.
exclude.categories	Specify a list of categories to be excluded from the network. For example, c("category 1", "category 2"). Note that the concept names must appear exactly as they are used on the dataset.
invert.persons	Reverse the selection of persons. If TRUE, the persons specified in the exclude.persons argument will be included, not excluded. No other persons will be included.
invert.organizations	Reverse the selection of organizations. If TRUE, the organizations specified in the exclude.organizations argument will be included, not excluded. No other organizations will be included.
invert.categories	Reverse the selection of categories. If TRUE, the categories specified in the exclude.categories argument will be included, not excluded. No other categories will be included.
verbose	If true, details about the data import and its progress will be printed. If false, these information will be suppressed.

Details

Specify an input .dna file, specify options for generating a network, and transfer the network as a matrix object into R.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA](#) [dna.init](#) [dna.attributes](#) [dna.categories](#)

Examples

```
# download files and initialize DNA:
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/sample.dna",
  destfile = "sample.dna", mode = "wb")
dna.init("dna-1.31.jar")

## Not run:
# plot a congruence network using the statnet package:
congruence <- dna.network("sample.dna", exclude.categories =
  "There should be legislation to regulate emissions.")
library("network")
congruence.nw <- network(congruence)
plot(congruence.nw, displaylabels = TRUE, label.cex = 0.6, pad = 0.8)

## End(Not run)

# do a hierarchical cluster analysis with an affiliation network:
affiliation.yes <- dna.network("sample.dna", algorithm = "affiliation",
  agreement = "yes", include.isolates = TRUE)
affiliation.no <- dna.network("sample.dna", algorithm = "affiliation",
  agreement = "no", include.isolates = TRUE)
affiliation <- cbind(affiliation.yes, affiliation.no)
affiliation <- affiliation[rowSums(affiliation) > 0, ] #remove isolates
distances <- dist(affiliation, method = "binary")
clustering <- hclust(distances)
plot(clustering)
```

`dna.timeseries`

Compute time series statistics

Description

Compute actor-wise statement frequency time series statistics.

Usage

```
dna.timeseries(infile, persons = FALSE, time.unit = "month",
  ignore.duplicates = "article", separate.actors = TRUE,
  start.date = "first", stop.date = "last",
  include.persons = "all", include.organizations = "all",
  include.categories = "all", invert.persons = FALSE,
  invert.organizations = FALSE, invert.categories = FALSE,
  agreement = "combined", verbose = TRUE)
```

Arguments

<code>infile</code>	The input .dna file as a string (i.e., enclosed in quotation marks). If the file is not in the current working directory, specify the path together with the file name. Include the file suffix. Example: <code>sample.dna</code> .
<code>persons</code>	Specify as a boolean value whether persons should be exported. If set to <code>false</code> , organizations are exported.
<code>time.unit</code>	If set to <code>month</code> , statement frequencies are computed per month. If set to <code>year</code> , the number of statements per year is computed. The value <code>total</code> causes DNA to aggregate all frequencies per actor into one column.
<code>ignore.duplicates</code>	A duplicate statement is a statement with the same person, organization, category and agreement pattern as another statement. By default, a duplicate statement is omitted if it is found within the same article (option <code>article</code>). Other options are <code>month</code> (i.e., duplicate statements are omitted if they occur in the same month) and <code>off</code> (i.e., all duplicates are retained).
<code>separate.actors</code>	If this boolean argument is set to <code>true</code> , there will be one actor per row in the resulting matrix. If <code>false</code> , the frequencies of all actors will be aggregated into a single row.
<code>start.date</code>	Only statements after this date will be retained. The start date is a character string of the form <code>dd.mm.yyyy</code> , where <code>dd</code> is the two-digit day, <code>mm</code> the two-digit month and <code>yyyy</code> the four-digit year. Alternatively, the value <code>first</code> will use the first date found in the .dna file.
<code>stop.date</code>	Only statements before this date will be retained. The stop date is a character string of the form <code>dd.mm.yyyy</code> , where <code>dd</code> is the two-digit day, <code>mm</code> the two-digit month and <code>yyyy</code> the four-digit year. Alternatively, the value <code>last</code> will use the last date found in the .dna file.
<code>include.persons</code>	Specify a list of persons to be included in the statistics. For example, <code>c("person 1", "person 2")</code> . Note that the names must appear exactly as they are used on the dataset. Alternatively, the string value <code>all</code> will include all persons in the .dna file.
<code>include.organizations</code>	Specify a list of organizations to be included in the statistics. For example, <code>c("organization 1", "organization 2")</code> . Note that the names must appear exactly as they are used on the dataset. Alternatively, the string value <code>all</code> will include all organizations in the .dna file.
<code>include.categories</code>	Specify a list of categories to be included in the statistics. For example, <code>c("category 1", "category 2")</code> . Note that the concept names must appear exactly as they are used on the dataset. Alternatively, the value <code>all</code> will include all categories in the .dna file.
<code>invert.persons</code>	Reverse the selection of persons. If <code>TRUE</code> , the persons specified in the <code>include.persons</code> argument will be excluded, not included. All other persons will be included.
<code>invert.organizations</code>	Reverse the selection of organizations. If <code>TRUE</code> , the organizations specified in the <code>include.organizations</code> argument will be excluded, not included. All other organizations will be included.

<code>invert.categories</code>	Reverse the selection of categories. If TRUE, the categories specified in the <code>include.categories</code> argument will be excluded, not included. All other categories will be included.
<code>agreement</code>	Can be combined, yes or no. If yes, only positive statements will be counted. If no, only negative statements will be counted. If combined, all kinds of statements are counted.
<code>verbose</code>	If true, details about the data import and its progress will be printed. If false, these information will be suppressed.

Details

An important piece of information in actor-based content analysis is the question how active each actor or actor group is over time. This function generates a matrix of statement frequencies of all actors, aggregated per actor or as a total value and either per year, per month or overall.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[rDNA dna.init dna.network dna.categories](#)

Examples

```
download.file("http://www.philipleifeld.de/cms/upload/Downloads/dna-1.31.jar",
  destfile = "dna-1.31.jar", mode = "wb")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/sample.dna",
  destfile = "sample.dna", mode = "wb")
dna.init("dna-1.31.jar")

time.series <- dna.timeseries("sample.dna")
```

rDNA

Control DNA from R

Description

Control the Java software Discourse Network Analyzer from within R

Details

A package that controls the Java software Discourse Network Analyzer (DNA) from within R. Network matrices, statement frequency time series and attributes of actors can be transferred directly into R. Requires the package **rJava**. Make sure you execute [dna.init](#) before running any other rDNA command.

To display citation information, execute `citation("rDNA")`.

Author(s)

Philip Leifeld (<http://www.philipleifeld.com>)

See Also

[dna.init](#) [dna.gui](#) [dna.attributes](#) [dna.network](#) [dna.timeseries](#) [dna.categories](#) [dna.density](#)

Index

`dna.attributes`, [2](#), [3](#), [5](#), [6](#), [9](#), [13](#)
`dna.categories`, [2](#), [3](#), [9](#), [12](#), [13](#)
`dna.density`, [4](#), [13](#)
`dna.gui`, [5](#), [6](#), [13](#)
`dna.init`, [2](#), [3](#), [6](#), [6](#), [9](#), [12](#), [13](#)
`dna.network`, [2](#), [3](#), [5](#), [6](#), [7](#), [12](#), [13](#)
`dna.timeseries`, [10](#), [13](#)

`rDNA`, [2](#), [3](#), [6](#), [9](#), [12](#), [12](#)
`rDNA-package (rDNA)`, [12](#)