

rDNA

A package to control Discourse Network Analyzer from R

Philip Leifeld

2010-11-17

Contents

1	Introduction	1
2	Requirements	1
3	Methods	2
3.1	dna.init	2
3.2	dna.gui	2
3.3	dna.network	2
3.4	dna.attributes	4
4	Examples	5

1 Introduction

rDNA is an R package which allows the user to control the Java software Discourse Network Analyzer (DNA).¹ DNA is a content analysis software for the extraction of social network data of actors and/or their concepts. rDNA facilitates the import of these network data directly from DNA into R. There is no need to save the data to a CSV file in between. This is very handy in situations where the data export should be integrated into an R script, e.g. in the appendix of a publication or during data analysis.

2 Requirements

rDNA requires the R package **rJava** and a Java Runtime Engine (at least version 1.6 or higher) running on the operating system.

¹Download the Discourse Network Analyzer from <http://www.philipleifeld.de>.

3 Methods

3.1 dna.init

Usage

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

Description

This method connects the rDNA package to DNA.

Arguments

dna.jar.file This is the name of the DNA JAR file (including its path). Make sure you put it into quotation marks. If the JAR file is in the current working directory, no path is needed. Example: `dna.init("dna-1.23.jar")`.

3.2 dna.gui

Usage

```
dna.gui()
```

Description

This method has no arguments. It starts the user interface of DNA. Files are *not* automatically transferred to R, so make sure you save your work in a .dna file.

3.3 dna.network

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,  
exclude.persons=c(""), exclude.organizations=c(""),  
exclude.categories=c(""))
```

Description

Extract a network from a .dna file and return it to R as a matrix object.

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.

two.mode.type If the **affiliation** algorithm is selected, this argument determines the vertex classes to be used. The following values are possible: **oc** (which stands for organizations x categories), **pc** (persons x categories) and **po** (persons x organizations).

one.mode.type If the **cooccurrence**, **timewindow** or **attenuation** algorithm is selected, this argument specifies the vertex class to be used. The following values are possible: **organizations** (which stands for organizations x organizations), **persons** (persons x persons) or **categories** (categories x categories).

via If the **one.mode.type** argument is active (i.e., the **cooccurrence** algorithm, the **timewindow** algorithm 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 **persons**, **organizations** and **categories**, but **not** the vertex type used in the **one.mode.type** argument.

ignore.duplicates 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 **ignore.duplicates** is switched off (i.e., set to **FALSE**).

include.isolates If several time slices are exported, usually the network matrices will have different dimensions. If the **include.isolates** argument is set to **TRUE**, 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.

normalization 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 **normalization** is set to **TRUE**, 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>.

window.size If the **timewindow** algorithm is used, the **window.size** 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.

step.size If the **timewindow** algorithm is used, the **step.size** 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 **window.size** argument.

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.

3.4 dna.attributes

Usage

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

Description

Pull attribute data of actors as found in the bottom bar of DNA into R. Works with persons and organizations. The resulting matrix has four 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).

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.

4 Examples

The following code demonstrates how rDNA can be used in conjunction with clustering methods and the statnet package.

```
# download files and initialize DNA:
download.file("http://www.philipleifeld.de/cms/upload/Downloads/
  dna-1.23.jar", destfile="dna-1.23.jar")
download.file("http://www.philipleifeld.de/cms/upload/Downloads/
  sample.dna", destfile="sample.dna")
library(rDNA)
dna.init("dna-1.23.jar")

# plot a congruence network using the statnet package:
congruence <- dna.network("sample.dna", exclude.categories=
  "There should be legislation to regulate emissions.")
library(statnet)
congruence.nw <- network(congruence)
plot(congruence.nw, displaylabels=TRUE, label.cex=0.6, pad=0.8)

# 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,]
distances <- dist(affiliation, method="binary")
clustering <- hclust(distances)
plot(clustering)

# open the GUI of DNA in order to manually work on the data
dna.gui()
```