Package ‘bibliometrix’

September 16, 2022

Type Package

Title Comprehensive Science Mapping Analysis

Version 4.0.1


License GPL-3


BugReports https://github.com/massimoaria/bibliometrix/issues

LazyData FALSE

Encoding UTF-8

Depends R (>= 3.3.0)


Suggests knitr, markdown, openalexR, testthat (>= 3.0.0), shinycssloaders, visNetwork, wordcloud2

RoxygenNote 7.2.1

NeedsCompilation no

Config/testthat/edition 3

Author Massimo Aria [cre, aut, cph] (https://orcid.org/0000-0002-8517-9411), Corrado Cuccurullo [aut] (https://orcid.org/0000-0002-7401-8575)
Maintainer Massimo Aria <aria@unina.it>
Repository CRAN
Date/Publication 2022-09-16 16:56:03 UTC

R topics documented:

bibliometrix-package .................................................. 3
authorProdOverTime ..................................................... 6
biblioAnalysis .......................................................... 7
biblioNetwork ............................................................ 8
bibioshiny ............................................................... 11
bibtag ................................................................. 12
bradford .............................................................. 12
citations ............................................................... 13
cocMatrix ............................................................... 14
centralization .......................................................... 16
cor2df ................................................................. 19
countries ............................................................... 20
couplingMap ............................................................ 21
dominance ............................................................... 23
duplicatedMatching ..................................................... 24
degreeByYear ............................................................ 25
degreeIndex ............................................................. 26
degreePlot ............................................................... 28
degreeStat ............................................................... 29
degreeByAuthor .......................................................... 30
degreeByAuthor .......................................................... 31
degreeByArticle .......................................................... 32
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
degreeBySubmission .................................................... 33
fieldByYear ............................................................... 25
Hindex ................................................................. 26
histNetwork ............................................................. 28
histPlot ............................................................... 29
idByAuthor .............................................................. 30
keywordAssoc ............................................................ 31
KeywordGrowth ........................................................... 32
localCitations ............................................................ 33
logo ................................................................. 35
lotka ................................................................. 35
mergeDbSources ........................................................ 36
metaTagExtraction ....................................................... 37
net2VOSSviewer .......................................................... 38
networkPlot ............................................................. 39
networkStat ............................................................. 42
normalizeCitationScore .................................................. 43
normalizeSimilarity ...................................................... 44
plot.bibliometrix .......................................................... 46
plotThematicEvolution ................................................... 47
readFiles ............................................................... 48
retrievalByAuthorID ....................................................... 48
rpy2 ................................................................. 50
sourceGrowth ........................................................... 51
splitCommunities ......................................................... 52
stopwords ............................................................... 53
summary.bibliometrix ..................................................... 54
**Description**


**Details**

**INSTALLATION**
- Stable version from CRAN:
  ```r```
  install.packages("bibliometrix")
  ```
- Or development version from GitHub:
  ```r```
  install.packages("devtools") devtools::install_github("massimoaria/bibliometrix")
  ```
- Load "bibliometrix"
  ```r```
  library('bibliometrix')
  ```
**DATA LOADING AND CONVERTING**

The export file can be imported and converted by R using the function `convert2df*`:
```r```
file <- "https://www.bibliometrix.org/datasets/savedrecs.txt"
M <- convert2df(file, dbsource = "wos", format = "bibtex")
```
*convert2df* creates a bibliographic data frame with cases corresponding to manuscripts and variables to Field Tag in the original export file. Each manuscript contains several elements, such as authors' names, title, keywords and other information. All these elements constitute the bibliographic attributes of a document, also called metadata. Data frame columns are named using the standard Clarivate Analytics WoS Field Tag codify.

**BIBLIOMETRIC ANALYSIS**
The first step is to perform a descriptive analysis of the bibliographic data frame. The function `biblioAnalysis` calculates main bibliometric measures using this syntax:

```r
results <- biblioAnalysis(M, sep = ";");
```

The function `biblioAnalysis` returns an object of class "bibliometrix".

To summarize main results of the bibliometric analysis, use the generic function `summary`. It displays main information about the bibliographic data frame and several tables, such as annual scientific production, top manuscripts per number of citations, most productive authors, most productive countries, total citation per country, most relevant sources (journals) and most relevant keywords. `summary` accepts two additional arguments. `k` is a formatting value that indicates the number of rows of each table. `pause` is a logical value (TRUE or FALSE) used to allow (or not) pause in screen scrolling. Choosing k=10 you decide to see the first 10 Authors, the first 10 sources, etc.

```r
S <- summary(object = results, k = 10, pause = FALSE)
```

Some basic plots can be drawn using the generic function plot:

```r
plot(x = results, k = 10, pause = FALSE)
```

**BIBLIOGRAPHIC NETWORK MATRICES**

Manuscript’s attributes are connected to each other through the manuscript itself: author(s) to journal, keywords to publication date, etc. These connections of different attributes generate bipartite networks that can be represented as rectangular matrices (Manuscripts x Attributes). Furthermore, scientific publications regularly contain references to other scientific works. This generates a further network, namely, co-citation or coupling network. These networks are analyzed in order to capture meaningful properties of the underlying research system, and in particular to determine the influence of bibliometric units such as scholars and journals.

`biblioNetwork` function

The function `biblioNetwork` calculates, starting from a bibliographic data frame, the most frequently used networks: Coupling, Co-citation, Co-occurrences, and Collaboration. `biblioNetwork` uses two arguments to define the network to compute: - `analysis` argument can be "co-citation", "coupling", "collaboration", or "co-occurrences". - `network` argument can be "authors", "references", "sources", "countries", "universities", "keywords", "author_keywords", "titles" and "abstracts".

i.e. the following code calculates a classical co-citation network:

```r
NetMatrix <- biblioNetwork(M, analysis = "co-citation", network = "references", sep = ",;")
```

**VISUALIZING BIBLIOGRAPHIC NETWORKS**

All bibliographic networks can be graphically visualized or modeled. Using the function `networkPlot`, you can plot a network created by `biblioNetwork` using R routines.

The main argument of `networkPlot` is `type`. It indicates the network map layout: circle, kamada-kawai, mds, etc.

In the following, we propose some examples.

```r
### Country Scientific Collaboration

# Create a country collaboration network
M <- metaTagExtraction(M, Field = "AU_CO", sep = ",;")
NetMatrix <- biblioNetwork(M, analysis = "collaboration", network = "countries", sep = ",;")
# Plot the network
```
### Co-Citation Network

# Create a co-citation network
NetMatrix <- biblioNetwork(M, analysis = "co-citation", network = "references", sep = ";")

# Plot the network
net = networkPlot(NetMatrix, n = 30, Title = "Co-Citation Network", type = "fruchterman", size = T, remove.multiple = FALSE, labelsize = 0.7, edgesize = 5)

### Keyword co-occurrences

# Create keyword co-occurrences network
NetMatrix <- biblioNetwork(M, analysis = "co-occurrences", network = "keywords", sep = ";")

# Plot the network
net = networkPlot(NetMatrix, normalize = "association", weighted = T, n = 30, Title = "Keyword Co-occurrences", type = "fruchterman", size = T, edgesize = 5, labelsize = 0.7)

**CO-WORD ANALYSIS: THE CONCEPTUAL STRUCTURE OF A FIELD**

The aim of the co-word analysis is to map the conceptual structure of a framework using the word co-occurrences in a bibliographic collection. The analysis can be performed through dimensionality reduction techniques such as Multidimensional Scaling (MDS), Correspondence Analysis (CA) or Multiple Correspondence Analysis (MCA). Here, we show an example using the function `conceptualStructure` that performs a CA or MCA to draw a conceptual structure of the field and K-means clustering to identify clusters of documents which express common concepts. Results are plotted on a two-dimensional map. `conceptualStructure` includes natural language processing (NLP) routines (see the function `termExtraction`) to extract terms from titles and abstracts. In addition, it implements the Porter’s stemming algorithm to reduce inflected (or sometimes derived) words to their word stem, base or root form.

# Conceptual Structure using keywords (method="MCA")
CS <- conceptualStructure(M, field="ID", method = "MCA", minDegree = 4, clust = 4, k.max = 8, stemming = FALSE, labelsize = 10, documents = 10)

**HISTORICAL DIRECT CITATION NETWORK**

The historiographic map is a graph proposed by E. Garfield to represent a chronological network map of most relevant direct citations resulting from a bibliographic collection. The function `histNetwork` generates a chronological direct citation network matrix which can be plotted using `histPlot`:

# Create a historical citation network
histResults <- histNetwork(M, sep = ";")

# Plot a historical co-citation network
net <- histPlot(histResults, size = 10)

**Author(s)**

NA

Maintainer: NA
### References


---

### authorProdOverTime

**Top-Authors’ Productivity over Time**

**Description**

It calculates and plots the author production (in terms of number of publications) over the time.

**Usage**

```
authorProdOverTime(M, k = 10, graph = TRUE)
```

**Arguments**

- `M` is a bibliographic data frame obtained by `convert2df` function.
- `k` is a integer. It is the number of top authors to analyze and plot. Default is `k = 10`.
- `graph` is logical. If TRUE the function plots the author production over time graph. Default is `graph = TRUE`.

**Value**

The function `authorProdOverTime` returns a list containing two objects:

- `dfAU` is a data frame
- `dfpapersAU` is a data frame
- `graph` a ggplot object
biblioAnalysis

See Also

biblioAnalysis function for bibliometric analysis
summary method for class 'bibliometrix'

Examples

data(scientometrics, package = "bibliometrixData")
res <- authorProdOverTime(scientometrics, k=10)
print(res$dfAU)
plot(res$graph)

Description

It performs a bibliometric analysis of a dataset imported from SCOPUS and Clarivate Analytics Web of Science databases.

Usage

biblioAnalysis(M, sep = ";")

Arguments

M is a bibliographic data frame obtained by the converting function convert2df. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics Web of Science file.

sep is the field separator character. This character separates strings in each column of the data frame. The default is sep = ";".

Value

biblioAnalysis returns an object of class "bibliometrix".

The functions summary and plot are used to obtain or print a summary and some useful plots of the results.

An object of class "bibliometrix" is a list containing the following components:

- Articles the total number of manuscripts
- Authors the authors' frequency distribution
- AuthorsFrac the authors' frequency distribution (fractionalized)
- FirstAuthors corresponding author of each manuscript
- nAUperPaper the number of authors per manuscript
- Appearances the number of author appearances
- nAuthors the number of authors
AuMultiAuthoredArt  the number of authors of multi-authored articles
MostCitedPapers    the list of manuscripts sorted by citations
Years              publication year of each manuscript
FirstAffiliation   the affiliation of the first author
Affiliations       the frequency distribution of affiliations (of all co-authors for each paper)
Aff_frac           the fractionalized frequency distribution of affiliations (of all co-authors for each paper)
CO                 the affiliation country of the first author
Countries          the affiliation countries’ frequency distribution
CountryCollaboration Intra-country (SCP) and intercountry (MCP) collaboration indices
TotalCitation      the number of times each manuscript has been cited
TCperYear          the yearly average number of times each manuscript has been cited
Sources            the frequency distribution of sources (journals, books, etc.)
DE                 the frequency distribution of authors’ keywords
ID                 the frequency distribution of keywords associated to the manuscript by SCOPUS and Clarivate Analytics

See Also
convert2df to import and convert an WoS or SCOPUS Export file in a bibliographic data frame.
summary to obtain a summary of the results.
plot to draw some useful plots of the results.

Examples
## Not run:
data(management, package = "bibliometrixData")
results <- biblioAnalysis(management)
summary(results, k = 10, pause = FALSE)
## End(Not run)

biblioNetwork

Creating Bibliographic networks

Description

biblioNetwork creates different bibliographic networks from a bibliographic data frame.

Usage

biblioNetwork(
  M,
  analysis = "coupling",
  network = "authors",
  n = NULL,
)
Arguments

M is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file.

analysis is a character object. It indicates the type of analysis can be performed. analysis argument can be "collaboration", "coupling", "co-occurrences" or "co-citation". Default is analysis = "coupling".

network is a character object. It indicates the network typology. The network argument can be "authors", "references", "sources", "countries", "keywords", "author_keywords", "titles", or "abstracts". Default is network = "authors".

n is an integer. It indicates the number of items to select. If N = NULL, all items are selected.

sep is the field separator character. This character separates strings in each column of the data frame. The default is sep = ";".

short is a logical. If TRUE all items with frequency<2 are deleted to reduce the matrix size.

shortlabel is logical. IF TRUE, reference labels are stored in a short format. Default is shortlabel = TRUE.

remove.terms is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is remove.terms = NULL.

synonyms is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is synonyms = NULL.

Details

The function `biblioNetwork` can create a collection of bibliographic networks following the approach proposed by Batagelj & Cerinsek (2013) and Aria & cuccurullo (2017).

Typical networks output of `biblioNetwork` are:

#### Collaboration Networks

- Authors collaboration (analysis = "collaboration", network = "authors")
- University collaboration (analysis = "collaboration", network = universities")
- Country collaboration (analysis = "collaboration", network = "countries")

#### Co-citation Networks

- Authors co-citation (analysis = "co-citation", network = "authors")
biblioNetwork

- Reference co-citation (analysis = "co-citation", network = "references")
- Source co-citation (analysis = "co-citation", network = "sources")

##### Coupling Networks
- Manuscript coupling (analysis = "coupling", network = "references")
- Authors coupling (analysis = "coupling", network = "authors")
- Source coupling (analysis = "coupling", network = "sources")
- Country coupling (analysis = "coupling", network = "countries")

##### Co-occurrences Networks
- Authors co-occurrences (analysis = "co-occurrences", network = "authors")
- Source co-occurrences (analysis = "co-occurrences", network = "sources")
- Keyword co-occurrences (analysis = "co-occurrences", network = "keywords")
- Author-Keyword co-occurrences (analysis = "co-occurrences", network = "author_keywords")
- Title content co-occurrences (analysis = "co-occurrences", network = "titles")
- Abstract content co-occurrences (analysis = "co-occurrences", network = "abstracts")

References:

Value

It is a squared network matrix. It is an object of class dgMatrix of the package Matrix.

See Also
convert2df to import and convert a SCOPUS and Thomson Reuters’ ISI Web of Knowledge export file in a data frame.
cocMatrix to compute a co-occurrence matrix.
biblioAnalysis to perform a bibliometric analysis.

Examples

# EXAMPLE 1: Authors collaboration network

# data(scientometrics, package = "bibliometrixData")

# NetMatrix <- biblioNetwork(scientometrics, analysis = "collaboration",
# network = "authors", sep = ";")

# net <- networkPlot(NetMatrix, n = 30, type = "kamada", Title = "Collaboration",labels=0.5)

# EXAMPLE 2: Co-citation network
```r
data(scientometrics, package = "bibliometrixData")

NetMatrix <- biblioNetwork(scientometrics, analysis = "co-citation",
                          network = "references", sep = ";")

net <- networkPlot(NetMatrix, n = 30, type = "kamada", Title = "Co-Citation", labelsize=0.5)
```

---

**biblioshiny**  
*Shiny UI for bibliometrix package*

**Description**

biblioshiny performs science mapping analysis using the main functions of the bibliometrix package.

**Usage**

```r
biblioshiny(
  host = "127.0.0.1",
  port = NULL,
  launch.browser = TRUE,
  maxUploadSize = 200
)
```

**Arguments**

- **host**: The IPv4 address that the application should listen on. Defaults to the shiny.host option, if set, or "127.0.0.1" if not.
- **port**: is the TCP port that the application should listen on. If the port is not specified, and the shiny.port option is set (with options(shiny.port = XX)), then that port will be used. Otherwise, use a random port.
- **launch.browser**: If true, the system’s default web browser will be launched automatically after the app is started. Defaults to true in interactive sessions only. This value of this parameter can also be a function to call with the application’s URL.
- **maxUploadSize**: is a integer. The max upload file size argument. Default value is 200 (megabyte)

**Examples**

```r
#biblioshiny()
```
**Description**

Data frame containing a list of tags and corresponding: WoS, SCOPUS and generic bibtex fields; and Dimensions.ai csv and xlsx fields.

**Format**

A data frame with 44 rows and 6 variables:

<table>
<thead>
<tr>
<th>TAG</th>
<th>Tag Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCOPUS</td>
<td>Scopus bibtex fields</td>
</tr>
<tr>
<td>ISI</td>
<td>WOS/ISI bibtex fields</td>
</tr>
<tr>
<td>GENERIC</td>
<td>Generic bibtex fields</td>
</tr>
<tr>
<td>DIMENSIONS_OLD</td>
<td>DIMENSIONS csv/xlsx old fields</td>
</tr>
<tr>
<td>DIMENSIONS</td>
<td>DIMENSIONS csv/xlsx fields</td>
</tr>
</tbody>
</table>

**bradford**

*Bradford’s law*

**Description**

It estimates and draws the Bradford’s law source distribution.

**Usage**

`bradford(M)`

**Arguments**

`M` is a bibliographic dataframe.

**Details**

Bradford’s law is a pattern first described by *(Samuel C. Bradford, 1934)* that estimates the exponentially diminishing returns of searching for references in science journals.

One formulation is that if journals in a field are sorted by number of articles into three groups, each with about one-third of all articles, then the number of journals in each group will be proportional to `1:n:n^2`.

Reference:

Value

The function bradford returns a list containing the following objects:

- **table**: a dataframe with the source distribution partitioned in the three zones
- **graph**: the source distribution plot in ggplot2 format

See Also

- **biblioAnalysis** function for bibliometric analysis
- **summary** method for class 'bibliometrix'

Examples

```r
## Not run:
file <- 'https://www.bibliometrix.org/datasets/bibliometrics_articles.txt'
M <- convert2df(file = file, dbsource = "isi", format = "bibtex")
BR <- bradford(M)
## End(Not run)
```

citations

**Citation frequency distribution**

Description

It calculates frequency distribution of citations.

Usage

```r
citations(M, field = "article", sep = ";")
```

Arguments

- **M**: is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics Web of Science file.
- **field**: is a character. It can be "article" or "author" to obtain frequency distribution of cited citations or cited authors (only first authors for WoS database) respectively. The default is `field = "article"`.
- **sep**: is the field separator character. This character separates citations in each string of CR column of the bibliographic data frame. The default is `sep = ";"`.

Value

an object of class "list" containing the following components:
cocMatrix

Cited the most frequent cited manuscripts or authors
Year the publication year (only for cited article analysis)
Source the journal (only for cited article analysis)

See Also

biblioAnalysis function for bibliometric analysis.
summary to obtain a summary of the results.
plot to draw some useful plots of the results.

Examples

## EXAMPLE 1: Cited articles

data(scientometrics, package = "bibliometrixData")

CR <- citations(scientometrics, field = "article", sep = ";")

CR$Cited[1:10]
CR$Year[1:10]
CR$Source[1:10]

## EXAMPLE 2: Cited first authors

data(scientometrics)

CR <- citations(scientometrics, field = "author", sep = ";")

CR$Cited[1:10]

cocMatrix

Co-occurrence matrix

Description

cocMatrix computes co-occurrences between elements of a Tag Field from a bibliographic data frame. Manuscript is the unit of analysis.

Usage

cocMatrix(
M,
Field = "AU",
type = "sparse",
n = NULL,
sep = ";",
binary = TRUE,
)
cocMatrix

```r
short = FALSE,
remove.terms = NULL,
synonyms = NULL
)
```

Arguments

- **M** is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original WoS or SCOPUS file.
- **Field** is a character object. It indicates one of the field tags of the standard ISI WoS Field Tag codify. Field can be equal to one of these tags:
  - AU Authors
  - SO Publication Name (or Source)
  - JI ISO Source Abbreviation
  - DE Author Keywords
  - ID Keywords associated by WoS or SCOPUS database
  - CR Cited References
for a complete list of filed tags see: Field Tags used in bibliometrix

- **type** indicates the output format of co-occurrences:
  - `type = "matrix"` produces an object of class matrix
  - `type = "sparse"` produces an object of class dgMatrix of the package Matrix. "sparse" argument generates a compact representation of the matrix.

- **n** is an integer. It indicates the number of items to select. If `N = NULL`, all items are selected.
- **sep** is the field separator character. This character separates strings in each column of the data frame. The default is `sep = ";"`.
- **binary** is a logical. If TRUE each cell contains a 0/1. if FALSE each cell contains the frequency.
- **short** is a logical. If TRUE all items with frequency<2 are deleted to reduce the matrix size.
- **remove.terms** is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is `remove.terms = NULL`.
- **synonyms** is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is `synonyms = NULL`.

Details

This co-occurrence matrix can be transformed into a collection of compatible networks. Through matrix multiplication you can obtain different networks. The function follows the approach proposed by Batagelj & Cerinsek (2013) and Aria & cuccurullo (2017).
References:

Value
a co-occurrence matrix with cases corresponding to manuscripts and variables to the objects extracted from the Tag Field.

See Also
convert2df to import and convert an ISI or SCOPUS Export file in a data frame.
biblioAnalysis to perform a bibliometric analysis.
biblioNetwork to compute a bibliographic network.

Examples
# EXAMPLE 1: Articles x Authors co-occurrence matrix
data(scientometrics, package = "bibliometrixData")
WA <- cocMatrix(scientometrics, Field = "AU", type = "sparse", sep = ";")

# EXAMPLE 2: Articles x Cited References co-occurrence matrix
# data(scientometrics, package = "bibliometrixData")
# WCR <- cocMatrix(scientometrics, Field = "CR", type = "sparse", sep = ";")

# EXAMPLE 3: Articles x Cited First Authors co-occurrence matrix
# data(scientometrics, package = "bibliometrixData")
# scientometrics <- metaTagExtraction(scientometrics, Field = "CR_AU", sep = ";")
# WCR <- cocMatrix(scientometrics, Field = "CR_AU", type = "sparse", sep = ",")

case study

conceptualStructure

Creating and plotting conceptual structure map of a scientific field

Description
The function conceptualStructure creates a conceptual structure map of a scientific field performing Correspondence Analysis (CA), Multiple Correspondence Analysis (MCA) or Metric Multidimensional Scaling (MDS) and Clustering of a bipartite network of terms extracted from keyword, title or abstract fields.
Usage

conceptualStructure(
  M,
  field = "ID",
  ngrams = 1,
  method = "MCA",
  quali.supp = NULL,
  quanti.supp = NULL,
  minDegree = 2,
  clust = "auto",
  k.max = 5,
  stemming = FALSE,
  labelsize = 10,
  documents = 2,
  graph = TRUE,
  remove.terms = NULL,
  synonyms = NULL
)

Arguments

M is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original ISI or SCOPUS file.

field is a character object. It indicates one of the field tags of the standard ISI WoS Field Tag codify. field can be equal to one of these tags:

- ID Keywords Plus associated by ISI or SCOPUS database
- DE Author's keywords
- ID_TM Keywords Plus stemmed through the Porter's stemming algorithm
- DE_TM Author's Keywords stemmed through the Porter's stemming algorithm
- TI Terms extracted from titles
- AB Terms extracted from abstracts

ngrams is an integer between 1 and 3. It indicates the type of n-gram to extract from texts. An n-gram is a contiguous sequence of n terms. The function can extract n-grams composed by 1, 2, 3 or 4 terms. Default value is `ngrams=1`.

method is a character object. It indicates the factorial method used to create the factorial map. Use method="CA" for Correspondence Analysis, method="MCA" for Multiple Correspondence Analysis or method="MDS" for Metric Multidimensional Scaling. The default is method="MCA"

quali.supp is a vector indicating the indexes of the categorical supplementary variables. It is used only for CA and MCA.

quanti.supp is a vector indicating the indexes of the quantitative supplementary variables. It is used only for CA and MCA.

minDegree is an integer. It indicates the minimum occurrences of terms to analize and plot. The default value is 2.
conceptualStructure

clust is an integer or a character. If clust="auto", the number of cluster is chosen automatically, otherwise clust can be an integer between 2 and 8.

k.max is an integer. It indicates the maximum number of cluster to keep. The default value is 5. The max value is 20.

stemming is logical. If TRUE the Porter's Stemming algorithm is applied to all extracted terms. The default is stemming = FALSE.

labels is an integer. It indicates the label size in the plot. Default is labels=10

documents is an integer. It indicates the number of documents per cluster to plot in the factorial map. The default value is 2. It is used only for CA and MCA.

graph is logical. If TRUE the function plots the maps otherwise they are saved in the output object. Default value is TRUE

remove.terms is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is remove.terms = NULL.

synonyms is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is synonyms = NULL.

Value

It is an object of the class list containing the following components:

net bipartite network
res Results of CA, MCA or MDS method
km.res Results of cluster analysis
graph_terms Conceptual structure map (class "ggplot2")
graph_documents_Contrib Factorial map of the documents with the highest contributes (class "ggplot2")
graph_docuemnts_TC Factorial map of the most cited documents (class "ggplot2")

See Also
termExtraction to extract terms from a textual field (abstract, title, author’s keywords, etc.) of a bibliographic data frame.
biblioNetwork to compute a bibliographic network.
cocMatrix to compute a co-occurrence matrix.
biblioAnalysis to perform a bibliometric analysis.

Examples

# EXAMPLE Conceptual Structure using Keywords Plus

data(scientometrics, package = "bibliometrixData")

CS <- conceptualStructure(scientometrics, field="ID", method="CA",
stemming=FALSE, minDegree=3, k.max = 5)
convert2df

Import and Convert bibliographic export files and API objects.

Description

It converts a SCOPUS, Clarivate Analytics WoS, Dimensions, Lens.org, PubMed and COCHRANE Database export files or pubmedR and dimensionsR JSON/XML objects into a data frame, with cases corresponding to articles and variables to Field Tags as used in WoS.

Usage

convert2df(file, dbsource = "wos", format = "plaintext")

Arguments

file

a character array containing a sequence of object names coming from:

a) Clarivate Analytics WoS (in plaintext `.txt`, Endnote Desktop `.ciw`, or bibtex formats `.bib`);
b) SCOPUS (exclusively in bibtex format `.bib`);
c) Digital Science Dimensions (in csv `.csv` or excel `.xlsx` formats);
d) Lens.org (in csv `.csv`);
e) an object of the class pubmedR (package pubmedR) containing a collection obtained from a query performed with pubmedR package;
f) an object of the class dimensionsR (package dimensionsR) containing a collection obtained from a query performed with dimensionsR package.

dbsource

is a character indicating the bibliographic database. dbsource can be "isi", "wos", "scopus", "dimensions" or "pubmed". Default is dbsource = "isi".

format

is a character indicating the format of the SCOPUS and Clarivate Analytics WoS export file. format can be "api", "bibtex", "plaintext", "endnote", "csv" or "excel". Default is format = "plaintext".

Value

a data frame with cases corresponding to articles and variables to Field Tags in the original export file.

Le We have three files downloaded from Web of Science in plaintext format, file will be:

file <- c("filename1.txt", "filename2.txt", "filename3.txt")

data frame columns are named using the standard Clarivate Analytics WoS Field Tag codify. The main field tags are:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>Authors</td>
</tr>
<tr>
<td>TI</td>
<td>Document Title</td>
</tr>
<tr>
<td>SO</td>
<td>Publication Name (or Source)</td>
</tr>
<tr>
<td>JI</td>
<td>ISO Source Abbreviation</td>
</tr>
<tr>
<td>DT</td>
<td>Document Type</td>
</tr>
<tr>
<td>DE</td>
<td>Authors’ Keywords</td>
</tr>
<tr>
<td>ID</td>
<td>Keywords associated by SCOPUS or WoS database</td>
</tr>
</tbody>
</table>
countries

<table>
<thead>
<tr>
<th>AB</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Author Address</td>
</tr>
<tr>
<td>RP</td>
<td>Reprint Address</td>
</tr>
<tr>
<td>CR</td>
<td>Cited References</td>
</tr>
<tr>
<td>TC</td>
<td>Times Cited</td>
</tr>
<tr>
<td>PY</td>
<td>Year</td>
</tr>
<tr>
<td>SC</td>
<td>Subject Category</td>
</tr>
<tr>
<td>UT</td>
<td>Unique Article Identifier</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
</tbody>
</table>

for a complete list of field tags see: Field Tags used in bibliometrix

Examples

# Example:
# Import and convert a Web of Science collection form an export file in plaintext format:

## Not run:
files <- 'https://www.bibliometrix.org/datasets/wos_plaintext.txt'
M <- convert2df(file = files, dbsource = 'wos', format = "plaintext")
## End(Not run)

---

countries  

Index of Countries.

Description

Data frame containing a normalized index of countries.
Data are used by `biblioAnalysis` function to extract Country Field of Cited References and Authors.

Format

A data frame with 197 rows and 4 variables:

- **countries**  country names
- **continent**  continent names
- **Longitude**  country centroid longitude
- **Latitude**   country centroid latitude
couplingMap

Description

It performs a coupling network analysis and plots community detection results on a bi-dimensional map (Coupling Map).

Usage

couplingMap(
  M,
  analysis = "documents",
  field = "CR",
  n = 500,
  label.term = NULL,
  ngrams = 1,
  impact.measure = "local",
  minfreq = 5,
  community.repulsion = 0.1,
  stemming = FALSE,
  size = 0.5,
  n.labels = 1,
  repel = TRUE,
  cluster = "walktrap"
)

Arguments

M is a bibliographic dataframe.

analysis is the textual attribute used to select the unit of analysis. It can be analysis = c("documents", "authors", "sources").

field is the textual attribute used to measure the coupling strength. It can be field = c("CR", "ID", "DE", "TI", "AB").

n is an integer. It indicates the number of units to include in the analysis.

label.term is a character. It indicates which content metadata have to use for cluster labeling. It can be label.term = c("ID", "DE", "TI", "AB"). If label.term = NULL cluster items will be use for labeling.

ngrams is an integer between 1 and 4. It indicates the type of n-gram to extract from texts. An n-gram is a contiguous sequence of n terms. The function can extract n-grams composed by 1, 2, 3 or 4 terms. Default value is ngrams=1.

impact.measure is a character. It indicates the impact measure used to rank cluster elements (documents, authors or sources). It can be impact.measure = c("local", "global"). With impact.measure = "local", couplingMap calculates elements impact using the Normalized Local Citation Score while using codeimpact.measure =
"global", the function uses the Normalized Global Citation Score to measure elements impact.

- **minfreq**: is a integer. It indicates the minimum frequency (per thousand) of a cluster. It is a number in the range (0,1000).
- **community.repulsion**: is a real. It indicates the repulsion force among network communities. It is a real number between 0 and 1. Default is community.repulsion = 0.1.
- **stemming**: is logical. If it is TRUE the word (from titles or abstracts) will be stemmed (using the Porter’s algorithm).
- **size**: is numerical. It indicates the size of the cluster circles and is a number in the range (0.01,1).
- **n.labels**: is integer. It indicates how many labels associate to each cluster. Default is n.labels = 1.
- **repel**: is logical. If it is TRUE ggplot uses geom_label_repel instead of geom_label.
- **cluster**: is a character. It indicates the type of cluster to perform among ("optimal", "louvain","leiden", "infomap","edge_betweenness","walktrap", "spinglass", "leading_eigen", "fast_greedy").

**Details**

The analysis can be performed on three different units: documents, authors or sources and the coupling strength can be measured using the classical approach (coupled by references) or a novel approach based on unit contents (keywords or terms from titles and abstracts).

The x-axis measures the cluster centrality (by Callon’s Centrality index) while the y-axis measures the cluster impact by Mean Normalized Local Citation Score (MNLCS). The Normalized Local Citation Score (NLCS) of a document is calculated by dividing the actual count of local citing items by the expected citation rate for documents with the same year of publication.

**Value**

- a list containing:
  - **map**: The coupling map as ggplot2 object
  - **clusters**: Centrality and Density values for each cluster.
  - **data**: A list of units following in each cluster
  - **nclust**: The number of clusters
  - **NCS**: The Normalized Citation Score dataframe
  - **net**: A list containing the network output (as provided from the networkPlot function)

**See Also**

- [biblioNetwork](#) function to compute a bibliographic network.
- [cocMatrix](#) to compute a bibliographic bipartite network.
- [networkPlot](#) to plot a bibliographic network.
dominance

Examples

```r
## Not run:
data(management, package = "bibliometrixData")
res <- couplingMap(management, analysis = "authors", field = "CR", n = 250, impact.measure="local",
                   minfreq = 3, size = 0.5, repel = TRUE)
plot(res$map)
## End(Not run)
```

dominance

Authors’ dominance ranking

Description

It calculates the authors’ dominance ranking from an object of the class 'bibliometrix' as proposed by Kumar & Kumar, 2008.

Usage

```r
dominance(results, k = 10)
```

Arguments

- `results` is an object of the class 'bibliometrix' for which the analysis of the authors’ dominance ranking is desired.
- `k` is an integer, used for table formatting (number of authors). Default value is 10.

Value

The function dominance returns a data frame with cases corresponding to the first k most productive authors and variables to typical field of a dominance analysis.

the data frame variables are:

- **Author**
- **Dominance Factor**
- **Tot Articles**
- **Single Authored**
- **Multi Authored**
- **First Authored**
- **Rank by Articles**
- **Rank by DF**

See Also

- `biblioAnalysis` function for bibliometric analysis
summary method for class 'bibliometrix'

Examples

data(scientometrics, package = "bibliometrixData")
results <- biblioAnalysis(scientometrics)
DF=dominance(results)
DF

duplicatedMatching

Searching of duplicated records in a bibliographic database

Description

Search duplicated records in a dataframe.

Usage

duplicatedMatching(M, Field = "TI", exact = FALSE, tol = 0.95)

Arguments

- **M** is the bibliographic data frame.
- **Field** is a character object. It indicates one of the field tags used to identify duplicated records. Field can be equal to one of these tags: TI (title), AB (abstract), UT (manuscript ID).
- **exact** is logical. If exact = TRUE the function searches duplicates using exact matching. If exact=FALSE, the function uses the restricted Damerau-Levenshtein distance to find duplicated documents.
- **tol** is a numeric value giving the minimum relative similarity to match two manuscripts. Default value is tol = 0.95. To use the restricted Damerau-Levenshtein distance, exact argument has to be set as FALSE.

Details

A bibliographic data frame is obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file. The function identifies duplicated records in a bibliographic data frame and deletes them. Duplicate entries are identified through the restricted Damerau-Levenshtein distance. Two manuscripts that have a relative similarity measure greater than tol argument are stored in the output data frame only once.

Value

the value returned from duplicatedMatching is a data frame without duplicated records.
### fieldByYear

It calculates the median year for each item of a field tag.

#### Description

It calculates the median year for each item of a field tag.

#### Usage

```r
fieldByYear(
  M,
  field = "ID",
  timespan = NULL,
  min.freq = 2,
  n.items = 5,
  labelsize = NULL,
  remove.terms = NULL,
  synonyms = NULL,
  dynamic.plot = FALSE,
  graph = TRUE
)
```

#### Arguments

- **M**: is a bibliographic data frame obtained by `convert2df` function.
- **field**: is a character object. It indicates one of the field tags of the standard ISI WoS Field Tag codify.

---

**fieldByYear**  
*Field Tag distribution by Year*

---

**Examples**

```r
data(scientometrics, package = "bibliometrixData")
M=rbind(scientometrics[1:20,],scientometrics[10:30,])
newM <- duplicatedMatching(M, Field = "TI", exact=FALSE, tol = 0.95)
dim(newM)
```
timespan is a vector with the min and max year. If it is NULL, the analysis is performed on the entire period. Default is timespan = NULL.

min.freq is an integer. It indicates the min frequency of the items to include in the analysis.

n.items is an integer. It indicates the maximum number of items per year to include in the plot.

labels is deprecated argument. It will be removed in the next update.

remove.terms is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is remove.terms = NULL.

synonyms is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is synonyms = NULL.

dynamic.plot is a logical. If TRUE plot aesthetics are optimized for plotly package.

graph is logical. If TRUE the function plots Field Tag distribution by Year graph. Default is graph = TRUE.

Value

The function fieldByYear returns a list containing three objects:

- df is a data frame
- df_graph is a data frame with data used to build the graph
- graph a ggplot object

See Also

- biblioAnalysis function for bibliometric analysis
- summary method for class 'bibliometrix'

Examples

```r
data(management, package = "bibliometrixData")
timespan=c(2005,2015)
res <- fieldByYear(management, field = "ID", timespan = timespan,
                  min.freq = 5, n.items = 5, graph = TRUE)
```

---

**Hindex**

**h-index calculation**

Description

It calculates the authors’ h-index and its variants.

Usage

```r
Hindex(M, field = "author", elements = NULL, sep = ";", years = 10)
```
Arguments

- **M** is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file.

- **field** is character. It can be equal to c("author", "source"). field indicates if H-index have to be calculated for a list of authors or for a list of sources. Default value is `field = "author"`.

- **elements** is a character vector. It contains the authors' names list or the source list for which you want to calculate the H-index. When the field is "author", the argument has the form C("SURNAME1 N","SURNAME2 N",...), in other words, for each author: surname and initials separated by one blank space. If elements=NULL, the function calculates impact indices for all elements contained in the data frame. i.e for the authors SEMPRONIO TIZIO CAIO and ARIA MASSIMO elements argument is elements = c("SEMPRONIO TC", "ARIA M").

- **sep** is the field separator character. This character separates authors in each string of AU column of the bibliographic data frame. The default is sep = ";";.

- **years** is an integer. It indicates the number of years to consider for Hindex calculation. Default is 10.

Value

an object of class "list". It contains two elements: H is a data frame with h-index, g-index and m-index for each author; CitationList is a list with the bibliographic collection for each author.

See Also

- `convert2df` to import and convert an WoS or SCOPUS Export file in a bibliographic data frame.
- `biblioAnalysis` function for bibliometric analysis.
- `summary` to obtain a summary of the results.
- `plot` to draw some useful plots of the results.

Examples

```R
### EXAMPLE 1: ###
data(scientometrics, package = "bibliometrixData")
authors <- c("SMALL H", "CHEN DZ")
Hindex(scientometrics, field = "author", elements = authors, sep = ";")$H
Hindex(scientometrics, field = "source", elements = "SCIENTOMETRICS", sep = ";")$H

### EXAMPLE 2: Garfield h-index###
data(garfield, package = "bibliometrixData")
```
indices=Hindex(garfield, field = "author", elements = "GARFIELD E", years=Inf, sep = ";")

# h-index, g-index and m-index of Eugene Garfield
indices$H

# Papers and total citations
head(indices$CitationList[[1]])

---

**histNetwork**

*Historical co-citation network*

**Description**

histNetwork creates a historical citation network from a bibliographic data frame.

**Usage**

```r
histNetwork(M, min.citations = 1, sep = ";", network = TRUE, verbose = TRUE)
```

**Arguments**

- **M** is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics Web of Science file.
- **min.citations** is a positive integer. It sets the minimum number of citations for the documents included in the analysis. It can be greater than or equal to 1. The default is `min.citations = 1`.
- **sep** is the field separator character. This character separates strings in CR column of the data frame. The default is `sep = ";"`.
- **network** is logical. If TRUE, function calculates and returns also the direct citation network. If FALSE, the function returns only the local citation table.
- **verbose** is logical. If TRUE, results are printed on screen.

**Value**

histNetwork returns an object of class "list" containing the following components:

- **NetMatrix** the historical co-citation network matrix
- **histData** the set of n most cited references
- **M** the bibliographic data frame

**See Also**

- `convert2df` to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.
- `summary` to obtain a summary of the results.
plot to draw some useful plots of the results.
biblioNetwork to compute a bibliographic network.

Examples

```r
## Not run:
data(management, package = "bibliometrixData")

histResults <- histNetwork(management, min.citations = 0, sep = ";")
## End(Not run)
```

### histPlot

**Plotting historical co-citation network**

**histPlot** plots a historical co-citation network.

**Usage**

```r
histPlot(
  histResults,
  n = 20,
  size = 5,
  labelsize = 5,
  title_as_label = FALSE,
  label = "short",
  verbose = TRUE
)
```

**Arguments**

- `histResults` is an object of class "list" containing the following components:
  - `NetMatrix` the historical citation network matrix
  - `Degree` the min degree of the network
  - `histData` the set of n most cited references
  - `M` the bibliographic data frame

- `n` is integer. It defines the number of vertices to plot.
- `size` is an integer. It defines the point size of the vertices. Default value is 5.
- `labelsize` is an integer. It indicates the label size in the plot. Default is `labelsize=5`
- `title_as_label` is a logical. DEPRECATED
- `label` is a string. It defines the label in the plot. Default is "short"
idByAuthor

Get Complete Author Information and ID from Scopus

Description

Uses SCOPUS API author search to identify author identification information.

Usage

idByAuthor(df, api_key)

Arguments

df is a dataframe composed of three columns:

<table>
<thead>
<tr>
<th>lastname</th>
<th>author's last name</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstname</td>
<td>author's first name</td>
</tr>
<tr>
<td>affiliation</td>
<td>Part of the affiliation name (university name, city, etc.)</td>
</tr>
</tbody>
</table>
When affiliation is not specified, the field df$affiliation have to be NA. i.e. df[2,1:3]<-c("cuccurullo","corrado", NA)

Value

a data frame with cases corresponding to authors and variables to author’s information and ID got from SCOPUS.

See Also

retrievalByAuthorID for downloading the complete author bibliographic collection from SCOPUS

Examples

```r
## Request a personal API Key to Elsevier web page https://dev.elsevier.com/sc_apis.html
# api_key="your api key"

## create a data frame with the list of authors to get information and IDs
# i.e. df[1,1:3]<-c("aria","massimo","naples")
# df[2,1:3]<-c("cuccurullo","corrado", NA)

## run idByAuthor function
# authorsID <- idByAuthor(df, api_key)
```
KeywordGrowth

sep is the field separator character. This character separates keywords in each string of ID and DE columns of the bibliographic data frame. The default is sep = ";".
n is an integer. It indicates the number of authors’ keywords to associate to each keyword plus. The default is n = 10.
excludeKW is a character vector. It contains authors’ keywords to exclude from the analysis.

Value

an object of class "list".

See Also

convert2df to import and convert a WoS or SCOPUS Export file in a bibliographic data frame.
biblioAnalysis function for bibliometric analysis.
summary to obtain a summary of the results.
plot to draw some useful plots of the results.

Examples

data(scientometrics, package = "bibliometrixData")
KWlist <- keywordAssoc(scientometrics, sep = ";", n = 10, excludeKW = NA)

# list of first 10 Keywords plus
names(KWlist)

# list of first 10 authors' keywords associated to the first Keyword plus
KWlist[[1]][1:10]

---

KeywordGrowth | Yearly occurrences of top keywords/terms

Description

It calculates yearly occurrences of top keywords/terms.

Usage

KeywordGrowth(
  M,
  Tag = "ID",
  sep = ";",
  top = 10,
  cdf = TRUE,
  remove.terms = NULL,
  synonyms = NULL
)
localCitations

Arguments

- **M** is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original WoS or SCOPUS file.

- **Tag** is a character object. It indicates one of the keyword field tags of the standard ISI WoS Field Tag codify (ID or DE) or a field tag created by `termExtraction` function (TI_TM, AB_TM, etc.).

- **sep** is the field separator character. This character separates strings in each keyword column of the data frame. The default is `sep = ";"`. 

- **top** is a numeric. It indicates the number of top keywords to analyze. The default value is 10.

- **cdf** is a logical. If TRUE, the function calculates the cumulative occurrences distribution.

- **remove.terms** is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is `remove.terms = NULL`.

- **synonyms** is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is `synonyms = NULL`.

Value

- an object of class `data.frame`

Examples

```r
data(scientometrics, package = "bibliometrixData")
topKW=KeywordGrowth(scientometrics, Tag = "ID", sep = ";", top=5, cdf=TRUE)
topKW

# Plotting results
## Not run:
install.packages("reshape2")
library(reshape2)
library(ggplot2)
DF=melt(topKW, id="Year")
ggplot(DF,aes(Year,value, group=variable, color=variable))+geom_line

## End(Not run)
```

localCitations

Description

It calculates local citations (LCS) of authors and documents of a bibliographic collection.
Usage

localCitations(M, fast.search = FALSE, sep = ";", verbose = FALSE)

Arguments

M is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file.

fast.search is logical. If true, the function calculates local citations only for 25 percent top cited documents.

sep is the field separator character. This character separates citations in each string of CR column of the bibliographic data frame. The default is sep = ";".

verbose is a logical. If TRUE, results are printed on screen.

Details

Local citations measure how many times an author (or a document) included in this collection have been cited by the documents also included in the collection.

Value

an object of class "list" containing author local citations and document local citations.

See Also

citations function for citation frequency distribution.
biblioAnalysis function for bibliometric analysis.
summary to obtain a summary of the results.
plot to draw some useful plots of the results.

Examples

data(scientometrics, package = "bibliometrixData")
CR <- localCitations(scientometrics, sep = ";")
CR$Authors[1:10,]
CR$Papers[1:10,]
Description

The matrix contains the rgb format of the bibliometrix official logo.

Format

A matrix with 927 rows and 800 columns.

Description

It estimates Lotka’s law coefficients for scientific productivity (Lotka A.J., 1926).

Usage

`lotka(results)`

Arguments

- `results` is an object of the class 'bibliometrix' for which the analysis of the authors’ dominance ranking is desired.

Details


Value

The function `lotka` returns a list of summary statistics of the Lotka’s law estimation of an object of class `bibliometrix`.

the list contains the following objects:

- **Beta** Beta coefficient
- **C** Constant coefficient
- **R2** Goodness of Fit
- **fitted** Fitted Values
- **p.value** Pvalue of two-sample Kolmogorov-Smirnov test between the empirical and the theoretical Lotka’s Law distribution
- **AuthorProd** Authors’ Productivity frequency table
mergeDbSources

Merge bibliographic data frames from SCOPUS and WoS

Description
Merge bibliographic data frames from different databases (WoS and SCOPUS) into a single one.

Usage
mergeDbSources(..., remove.duplicated = TRUE)

Arguments
... are the bibliographic data frames to merge.
remove.duplicated is logical. If TRUE duplicated documents will be deleted from the bibliographic collection.

Details
bibliographic data frames are obtained by the converting function convert2df. The function merges data frames identifying common tag fields and duplicated records.

Value
the value returned from mergeDbSources is a bibliographic data frame.

See Also
convert2df to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.
biblioAnalysis function for bibliometric analysis.
summary to obtain a summary of the results.
plot to draw some useful plots of the results.
Examples

```r
dataset(isiCollection, package = "bibliometrixData")
data(scopusCollection, package = "bibliometrixData")
M <- mergeDbSources(isiCollection, scopusCollection, remove.duplicated=TRUE)
dim(M)
```

---

**metaTagExtraction**  
*Meta-Field Tag Extraction*

---

**Description**

It extracts other field tags, different from the standard WoS/SCOPUS codify.

**Usage**

```r
metaTagExtraction(M, Field = "CR_AU", sep = ";", aff.disamb = TRUE)
```

**Arguments**

- **M** is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original WoS or SCOPUS file.
- **Field** is a character object. New tag extracted from aggregated data is specified by this string. Field can be equal to one of these tags:
  - "CR_AU"  - First Author of each cited reference
  - "CR_SO"  - Source of each cited reference
  - "AU_CO"  - Country of affiliation for co-authors
  - "AU1_CO" - Country of affiliation for the first author
  - "AU_UN"  - University of affiliation for each co-author and the corresponding author (AU1_UN)
  - "SR"     - Short tag of the document (as used in reference lists)
- **sep** is the field separator character. This character separates strings in each column of the data frame. The default is sep = ";".
- **aff.disamb** is a logical. If TRUE and Field="AU_UN", then a disambiguation algorithm is used to identify and match scientific affiliations (univ, research centers, etc.). The default is aff.disamb=TRUE.
Value

the bibliometric data frame with a new column containing data about new field tag indicated in the argument Field.

See Also

convert2df for importing and converting bibliographic files into a data frame.
biblioAnalysis function for bibliometric analysis

Examples

# Example 1: First Authors for each cited reference

data(scientometrics, package = "bibliometrixData")
scientometrics <- metaTagExtraction(scientometrics, Field = "CR_AU", sep = ";";"

unlist(strsplit(scientometrics$CR_AU[1], ";;"))

#Example 2: Source for each cited reference

data(scientometrics)
scientometrics <- metaTagExtraction(scientometrics, Field = "CR_SO", sep = ";;"

unlist(strsplit(scientometrics$CR_SO[1], ";;"))

#Example 3: Affiliation country for co-authors

data(scientometrics)
scientometrics <- metaTagExtraction(scientometrics, Field = "AU_CO", sep = ";;"

scientometrics$AU_CO[1:10]

net2VOSviewer

Open a bibliometrix network in VosViewer

Description

net2VOSviewer plots a network created with networkPlot using VOSviewer by Nees Jan van Eck and Ludo Waltman.

Usage

net2VOSviewer(net, vos.path = NULL)

Arguments

net is an object created by networkPlot function.
vos.path is a character indicating the full path where VOSviewer.jar is located.
Details

The function `networkPlot` can plot a bibliographic network previously created by `biblioNetwork`. The network map can be plotted using internal R routines or using VOSviewer by Nees Jan van Eck and Ludo Waltman.

Value

It write a .net file that can be open in VOSviewer

See Also

`biblioNetwork` to compute a bibliographic network.

`networkPlot` to create and plot a network object

Examples

```r
# EXAMPLE

# VOSviewer.jar have to be present in the working folder

# data(scientometrics, package = "bibliometrixData")

# NetMatrix <- biblioNetwork(scientometrics, analysis = "co-citation",
# network = "references", sep = ";")

# net <- networkPlot(NetMatrix, n = 30, type = "kamada", Title = "Co-Citation", labelsize=0.5)

# net2VOSviewer(net)
```

Description

`networkPlot` plots a bibliographic network.

Usage

`networkPlot(`
  NetMatrix,
  normalize = NULL,
  n = NULL,
  degree = NULL,
  Title = "Plot",
  type = "auto",
  label = TRUE,
  labelsize = 1,
)
label.cex = FALSE,
label.color = FALSE,
label.n = NULL,
halo = FALSE,
cluster = "leading_eigen",
community.repulsion = 0.1,
vos.path = NULL,
size = 3,
size.cex = FALSE,
curved = FALSE,
noloops = TRUE,
remove.multiple = TRUE,
remove.isolates = FALSE,
weighted = NULL,
edgesize = 1,
edges.min = 0,
alpha = 0.5,
verbose = TRUE)

Arguments

NetMatrix is a network matrix obtained by the function biblioNetwork.

normalize is a character. It can be "association", "jaccard", "inclusion", "salton" or "equivalence" to obtain Association Strength, Jaccard, Inclusion, Salton or Equivalence similarity index respectively. The default is type = NULL.

n is an integer. It indicates the number of vertices to plot.

degree is an integer. It indicates the min frequency of a vertex. If degree is not NULL, n is ignored.

Title is a character indicating the plot title.

type is a character object. It indicates the network map layout:

type="auto" Automatic layout selection
  type="circle" Circle layout
  type="sphere" Sphere layout
  type="mds" Multidimensional Scaling layout
  type="fruchterman" Fruchterman-Reingold layout
  type="kamada" Kamada-Kawai layout

label is logical. If TRUE vertex labels are plotted.

label.size is an integer. It indicates the label size in the plot. Default is label.size=1

label.cex is logical. If TRUE the label size of each vertex is proportional to its degree.

label.color is logical. If TRUE, for each vertex, the label color is the same as its cluster.

label.n is an integer. It indicates the number of vertex labels to draw.

halo is logical. If TRUE communities are plotted using different colors. Default is halo=FALSE
cluster is a character. It indicates the type of cluster to perform among ("none", "optimal", "louvain", "leiden", "infomap", "edge_betweenness", "walktrap", "spinglass", "leading_eigen", "fast_greedy").

community.repulsion is a real. It indicates the repulsion force among network communities. It is a real number between 0 and 1. Default is community.repulsion = 0.1.

vos.path is a character indicating the full path where VOSviewer.jar is located.

size is integer. It defines the size of each vertex. Default is size=3.

size.cex is logical. If TRUE the size of each vertex is proportional to its degree.

curved is a logical or a number. If TRUE edges are plotted with an optimal curvature. Default is curved=FALSE. Curved values are any numbers from 0 to 1.

noloops is logical. If TRUE loops in the network are deleted.

remove.multiple is logical. If TRUE multiple links are plotted using just one edge.

remove.isolates is logical. If TRUE isolates vertices are not plotted.

weighted This argument specifies whether to create a weighted graph from an adjacency matrix. If it is NULL then an unweighted graph is created and the elements of the adjacency matrix gives the number of edges between the vertices. If it is a character constant then for every non-zero matrix entry an edge is created and the value of the entry is added as an edge attribute named by the weighted argument. If it is TRUE then a weighted graph is created and the name of the edge attribute will be weight.

edgesize is an integer. It indicates the network edge size.

edges.min is an integer. It indicates the min frequency of edges between two vertices. If edge.min=0, all edges are plotted.

alpha is a number. Legal alpha values are any numbers from 0 (transparent) to 1 (opaque). The default alpha value usually is 0.5.

verbose is a logical. If TRUE, network will be plotted. Default is verbose = TRUE.

Details

The function networkPlot can plot a bibliographic network previously created by biblioNetwork.

Value

It is a list containing the following elements:

- graph a network object of the class igraph
- cluster_obj a communities object of the package igraph
- cluster_res a data frame with main results of clustering procedure.
networkStat

See Also

biblioNetwork to compute a bibliographic network.
net2VOSviewer to export and plot the network with VOSviewer software.
cocMatrix to compute a co-occurrence matrix.
biblioAnalysis to perform a bibliometric analysis.

Examples

```r
# EXAMPLE Keyword co-occurrence network

data(management, package = "bibliometrixData")

NetMatrix <- biblioNetwork(management, analysis = "co-occurrences",
                          network = "keywords", sep = ";")

net <- networkPlot(NetMatrix, n = 30, type = "auto", Title = "Co-occurrence Network",labelsize=1)
```

Description
	networkStat calculates main network statistics.

Usage

```r
networkStat(object, stat = "network", type = "degree")
```

Arguments

- `object` is a network matrix obtained by the function `biblioNetwork` or an graph object of the class igraph.
- `stat` is a character. It indicates which statistics are to be calculated. `stat = "network"` calculates the statistics related to the network; `stat = "all"` calculates the statistics related to the network and the individual nodes that compose it. Default value is `stat = "network"`.
- `type` is a character. It indicates which centrality index is calculated. `type` values can be `c("degree", "closeness", "betweenness","eigenvector","pagerank","hub","authority", "all")`. Default is "degree".

Details

The function `networkStat` can calculate the main network statistics from a bibliographic network previously created by `biblioNetwork`. 
normalizeCitationScore

Value

It is a list containing the following elements:

- graph: a network object of the class igraph
- network: a communities: a list with the main statistics of the network
- vertex: a data frame with the main measures of centrality and prestige of vertices.

See Also

biblioNetwork to compute a bibliographic network.
cocMatrix to compute a co-occurrence matrix.
biblioAnalysis to perform a bibliometric analysis.

Examples

# EXAMPLE Co-citation network

# to run the example, please remove # from the beginning of the following lines
# data(scientometrics, package = "bibliometrixData")

# NetMatrix <- biblioNetwork(scientometrics, analysis = "co-citation",
# network = "references", sep = ";")

# netstat <- networkStat(NetMatrix, stat = "all", type = "degree")

normalizeCitationScore

Calculate the normalized citation score metric

Description

It calculates the normalized citation score for documents, authors and sources using both global and
local citations.

Usage

normalizeCitationScore(M, field = "documents", impact.measure = "local")

Arguments

- M: a bibliographic data frame obtained by convert2df function.
- field: is a character. It indicates the unit of analysis on which calculate the NCS. It
can be equal to field = c("documents", "authors", "sources"). Default is
field = "documents".
impact.measure is a character. It indicates the impact measure used to rank cluster elements (documents, authors or sources). It can be `impact.measure = c("local", "global")`.

With `impact.measure = "local"`, `normalizeCitationScore` calculates elements impact using the Normalized Local Citation Score while using code `impact.measure = "global"`, the function uses the Normalized Global Citation Score to measure elements impact.

**Details**

The document Normalized Citation Score (NCS) of a document is calculated by dividing the actual count of citing items by the expected citation rate for documents with the same year of publication.

The MNCS of a set of documents, for example the collected works of an individual, or published on a journal, is the average of the NCS values for all the documents in the set.

The NGCS is the NCS calculated using the global citations (total citations that a document received considering the whole bibliographic database).

The NLCS is the NCS calculated using the local citations (total citations that a document received from a set of documents included in the same collection).

**Value**

a dataframe.

**Examples**

```r
## Not run:
data(management, package = "bibliometrixData")
NCS <- normalizeCitationScore(M, field = "authors", impact.measure = "local")
## End(Not run)
```

---

**normalizeSimilarity**  
*Calculate similarity indices*

**Description**

It calculates a relative measure of bibliographic co-occurrences.

**Usage**

`normalizeSimilarity(NetMatrix, type = "association")`

**Arguments**

- **NetMatrix** is a coupling matrix obtained by the network functions `biblioNetwork` or `cocMatrix`.
- **type** is a character. It can be "association", "jaccard", "inclusion", "salton" or "equivalence" to obtain Association Strength, Jaccard, Inclusion, Salton or Equivalence similarity index respectively. The default is `type = "association"`. 
normalizeSimilarity

details

couplingSimilarity calculates Association strength, Inclusion, Jaccard or Salton similarity from a co-occurrence bibliographic matrix.

The association strength is used by Van Eck and Waltman (2007) and Van Eck et al. (2006). Several works refer to the measure as the proximity index, while Leydesdorff (2008) and Zitt et al. (2000) refer to it as the probabilistic affinity (or activity) index.

The inclusion index, also called Simpson coefficient, is an overlap measure used in information retrieval.

The Jaccard index (or Jaccard similarity coefficient) gives us a relative measure of the overlap of two sets. It is calculated as the ratio between the intersection and the union of the reference lists (of two manuscripts).

The Salton index, instead, relates the intersection of the two lists to the geometric mean of the size of both sets. The square of Salton index is also called Equivalence index.

The indices are equal to zero if the intersection of the reference lists is empty.

references


value

a similarity matrix.

see also

biblioNetwork function to compute a bibliographic network.
cocMatrix to compute a bibliographic bipartite network.

examples

data(scientometrics, package = "bibliometrixData")
NetMatrix <- biblioNetwork(scientometrics, analysis = "co-occurrences",
                   network = "keywords", sep = ";")
plot.bibliometrix

Description

plot method for class 'bibliometrix'

Usage

## S3 method for class 'bibliometrix'
plot(x, 
...)

Arguments

x is the object for which plots are desired.
... can accept two arguments:
k is an integer, used for plot formatting (number of objects). Default value is 10.
pause is a logical, used to allow pause in screen scrolling of results. Default value is pause = FALSE.

Value

The function plot returns a list of plots of class ggplot2.

See Also

The bibliometric analysis function biblioAnalysis.
summary to compute a list of summary statistics of the object of class bibliometrix.

Examples

data(scientometrics, package = "bibliometrixData")
results <- biblioAnalysis(scientometrics)
plot(results, k = 10, pause = FALSE)
plotThematicEvolution  Plot a Thematic Evolution Analysis

Description
It plots a Thematic Evolution Analysis performed using the `thematicEvolution` function.

Usage
```r
plotThematicEvolution(Nodes, Edges, measure = "inclusion", min.flow = 0)
```

Arguments
- **Nodes**: is a list of nodes obtained by `thematicEvolution` function.
- **Edges**: is a list of edges obtained by `thematicEvolution` function.
- **measure**: is a character. It can be `measure=("inclusion","stability", "weighted")`.
- **min.flow**: is numerical. It indicates the minimum value of measure to plot a flow.

Value
a sankeyPlot

See Also
- `thematicMap` function to create a thematic map based on co-word network analysis and clustering.
- `thematicMap` function to perform a thematic evolution analysis.
- `networkPlot` to plot a bibliographic network.

Examples
```r
## Not run:
data(scientometrics, package = "bibliometrixData")
years=c(2000)
nexus <- thematicEvolution(scientometrics,field="ID",years=years,n=100,minFreq=2)
plotThematicEvolution(nexus$Nodes,nexus$Edges)
## End(Not run)
```
readFiles

DEPRECATED: Load a sequence of ISI or SCOPUS Export files into a large character object

Description

The function readFiles is deprecated. You can import and convert your export files directly using the function convert2df.

Usage

readFiles(...) 

Arguments

... is a sequence of names of files downloaded from WOS.(in plain text or bibtex format) or SCOPUS Export file (exclusively in bibtex format).

Value

a character vector of length the number of lines read.

See Also

convert2df for converting SCOPUS of ISI Export file into a dataframe

Examples

# WoS or SCOPUS Export files can be read using \code{readFiles} function:
# largechar <- readFiles('filename1.txt','filename2.txt','filename3.txt')

# filename1.txt, filename2.txt and filename3.txt are ISI or SCOPUS Export file
# in plain text or bibtex format.
# D <- readFiles('https://www.bibliometrix.org/datasets/bibliometrics_articles.txt')

retrievalByAuthorID

Get Author Content on SCOPUS by ID

Description

Uses SCOPUS API search to get information about documents on a set of authors using SCOPUS ID.
Usage
	retrievalByAuthorID(id, api_key, remove.duplicated = TRUE, country = TRUE)

Arguments

id is a vector of characters containing the author’s SCOPUS IDs. SCOPUS IDs can be obtained using the function idByAuthor.

api_key is a character. It contains the Elsevier API key. Information about how to obtain an API key Elsevier API website

remove.duplicated is logical. If TRUE duplicated documents will be deleted from the bibliographic collection.

country is logical. If TRUE authors’ country information will be downloaded from SCOPUS.

Value

a list containing two objects: (i) M which is a data frame with cases corresponding to articles and variables to main Field Tags named using the standard ISI WoS Field Tag codify. M includes the entire bibliographic collection downloaded from SCOPUS. The main field tags are:

- AU Authors
- TI Document Title
- SO Publication Name (or Source)
- DT Document Type
- DE Authors’ Keywords
- ID Keywords associated by SCOPUS or ISI database
- AB Abstract
- C1 Author Address
- RP Reprint Address
- TC Times Cited
- PY Year
- UT Unique Article Identifier
- DB Database

(ii) authorDocuments which is a list containing a bibliographic data frame for each author.

LIMITATIONS: Currently, SCOPUS API does not allow to download document references. As consequence, it is not possible to perform co-citation analysis (the field CR is empty).

See Also

idByAuthor for downloading author information and SCOPUS ID.

Examples

## Request a personal API Key to Elsevier web page https://dev.elsevier.com/sc_apis.html
## api_key="your api key"

## create a data frame with the list of authors to get information and IDs
# i.e. df[1,1:3] <- c("aria","massimo","naples")
#       df[2,1:3] <- c("cuccurullo","corrado","naples")

## run idByAuthor function
#
# authorsID <- idByAuthor(df, api_key)
#

## extract the IDs
#
# id <- authorsID[,3]
#

## create the bibliographic collection
#
# res <- retrievalByAuthorID(id, api_key)
#
# M <- res$M # the entire bibliographic data frame
# M <- res$authorDocuments # the list containing a bibliographic data frame for each author

---

**r pys**

*Reference Publication Year Spectroscopy*

---

**Description**

r pys computes a Reference Publication Year Spectroscopy for detecting the Historical Roots of Research Fields. The method was introduced by Marx et al., 2014.

**Usage**

r pys(M, sep = ";", timespan = NULL, graph = T)

**Arguments**

- **M** is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original ISI or SCOPUS file.
- **sep** is the cited-references separator character. This character separates cited-references in the CR column of the data frame. The default is `sep = ";"`.
- **timespan** is a numeric vector `c(min year,max year)`. The default value is `NULL` (the entire timespan is considered).
- **graph** is a logical. If `TRUE` the function plot the spectroscopy otherwise the plot is created but not drawn down.
sourceGrowth

Details
Reference:

Value
a list containing the spectroscopy (class ggplot2) and two dataframes with the number of citations per year and the list of the cited references for each year, respectively.

See Also
csv2df to import and convert an ISI or SCOPUS Export file in a data frame.
biblioAnalysis to perform a bibliometric analysis.
biblioNetwork to compute a bibliographic network.

Examples

```r
data(scientometrics, package = "bibliometrixData")
res <- rpys(scientometrics, sep=";", graph = TRUE)
```

<table>
<thead>
<tr>
<th>sourceGrowth</th>
<th>Number of documents published annually per Top Sources</th>
</tr>
</thead>
</table>

Description
It calculates yearly published documents of the top sources.

Usage

```
sourceGrowth(M, top = 5, cdf = TRUE)
```

Arguments

- **M**
  is a data frame obtained by the converting function csv2df. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original ISI or SCOPUS file.

- **top**
  is a numeric. It indicates the number of top sources to analyze. The default value is 5.

- **cdf**
  is a logical. If TRUE, the function calculates the cumulative occurrences distribution.
splitCommunities

Splitting Network communities

Description

networkPlot Create a network plot with separated communities.

Usage

splitCommunities(graph, n = NULL)

Arguments

graph is a network plot obtained by the function networkPlot.

n is an integer. It indicates the number of vertices to plot for each community.

Details

The function splitCommunities splits communities in separated subnetworks from a bibliographic network plot previously created by networkPlot.

Value

It is a network object of the class igraph

Examples

data(scientometrics, package = "bibliometrixData")
topSO=sourceGrowth(scientometrics, top=1, cdf=TRUE)
topSO

# Plotting results
## Not run:
install.packages("reshape2")
library(reshape2)
library(ggplot2)
DF=melt(topSO, id='Year')
ggplot(DF,aes(Year,value, group=variable, color=variable))+geom_line()

## End(Not run)
stopwords

See Also

biblioNetwork to compute a bibliographic network.

networkPlot to plot a bibliographic network.

net2VOSviewer to export and plot the network with VOSviewer software.

cocMatrix to compute a co-occurrence matrix.

biblioAnalysis to perform a bibliometric analysis.

Examples

# EXAMPLE Keyword co-occurrence network
data(management, package = "bibliometrixData")

NetMatrix <- biblioNetwork(management, analysis = "co-occurrences",
network = "keywords", sep = ";")

net <- networkPlot(NetMatrix, n = 30, type = "auto",
                   Title = "Co-occurrence Network", labelsize=1, verbose=FALSE)

graph <- splitCommunities(net$graph, n = 30)

stopwords

List of English stopwords.

Description

A character vector containing a complete list of English stopwords
Data are used by biblioAnalysis function to extract Country Field of Cited References and Authors.

Format

A character vector with 665 rows.
Description

summary method for class 'bibliometrix'

Usage

```r
## S3 method for class 'bibliometrix'
summary(object, ...)
```

Arguments

- `object` is the object for which a summary is desired.
- `...` can accept two arguments:
  - `k` integer, used for table formatting (number of rows). Default value is 10.
  - `pause` logical, used to allow pause in screen scrolling of results. Default value is `pause = FALSE`.
  - `width` integer, used to define screen output width. Default value is `width = 120`.
  - `verbose` logical, used to allow screen output. Default is TRUE.

Value

The function `summary` computes and returns a list of summary statistics of the object of class `bibliometrics`. The list contains the following objects:

- `MainInformation` Main Information about Data
- `AnnualProduction` Annual Scientific Production
- `AnnualGrowthRate` Annual Percentage Growth Rate
- `MostProdAuthors` Most Productive Authors
- `MostCitedPapers` Top manuscripts per number of citations
- `MostProdCountries` Corresponding Author’s Countries
- `TCperCountries` Total Citation per Countries
- `MostRelSources` Most Relevant Sources
- `MostRelKeywords` Most Relevant Keywords

See Also

- `biblioAnalysis` function for bibliometric analysis
- `plot` to draw some useful plots of the results.

Examples

```r
data(scientometrics, package = "bibliometrixData")```
results <- biblioAnalysis(scientometrics)
summary(results)

---

**Summary**

Summary method for class 'bibliometrix_netstat'

**Usage**

```r
## S3 method for class 'bibliometrix_netstat'
summary(object, ...)
```

**Arguments**

- `object` is the object for which a summary is desired.
- `...` can accept two arguments:
  - `k` integer, used for table formatting (number of rows). Default value is 10.

**Value**

The function `summary` computes and returns on display several statistics both at network and vertex level.

**Examples**

# to run the example, please remove # from the beginning of the following lines
#data(scientometrics, package = "bibliometrixData")

```r
#NetMatrix <- biblioNetwork(scientometrics, analysis = "collaboration",
# network = "authors", sep = ":")
#netstat <- networkStat(NetMatrix, stat = "all", type = "degree")
#summary(netstat)
```
**tableTag**

Tabulate elements from a Tag Field column

**Description**

It tabulates elements from a Tag Field column of a bibliographic data frame.

**Usage**

```r
tableTag(
  M,
  Tag = "CR",
  sep = ",;",
  ngrams = 1,
  remove.terms = NULL,
  synonyms = NULL
)
```

**Arguments**

- `M` is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original WoS or SCOPUS file.
- `Tag` is a character object. It indicates one of the field tags of the standard ISI WoS Field Tag codify.
- `sep` is the field separator character. This character separates strings in each column of the data frame. The default is `sep = ",;".`
- `ngrams` is an integer between 1 and 3. It indicates the type of n-gram to extract from titles or abstracts.
- `remove.terms` is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is `remove.terms = NULL`.
- `synonyms` is a character vector. Each element contains a list of synonyms, separated by ",;", that will be merged into a single term (the first word contained in the vector element). The default is `synonyms = NULL`.

**Details**

`tableTag` is an internal routine of main function `biblioAnalysis`.

**Value**

an object of class `table`
Term extraction tool from textual fields of a manuscript

Description

It extracts terms from a text field (abstract, title, author’s keywords, etc.) of a bibliographic data frame.

Usage

```r
termExtraction(
  M,
  Field = "TI",
  ngrams = 1,
  stemming = FALSE,
  language = "english",
  remove.numbers = TRUE,
  remove.terms = NULL,
  keep.terms = NULL,
  synonyms = NULL,
  verbose = TRUE
)
```

Arguments

- **M** is a data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to articles and variables to Field Tag in the original WoS or SCOPUS file.
- **Field** is a character object. It indicates the field tag of textual data:
  - "TI" Manuscript title
  - "AB" Manuscript abstract
  - "ID" Manuscript keywords plus
  - "DE" Manuscript author’s keywords

  The default is `Field = "TI"`.
- **ngrams** is an integer between 1 and 3. It indicates the type of n-gram to extract from texts. An n-gram is a contiguous sequence of n terms. The function can extract n-grams composed by 1, 2, 3 or 4 terms. Default value is `ngrams=1`.  

Examples

```r
data(scientometrics, package = "bibliometrixData")
Tab <- tableTag(scientometrics, Tag = "CR", sep = ";")
Tab[1:10]
```
termExtraction

stemming is logical. If TRUE the Porter Stemming algorithm is applied to all extracted terms. The default is stemming = FALSE.

language is a character. It is the language of textual contents ("english", "german", "italian", "french", "spanish"). The default is language="english".

remove.numbers is logical. If TRUE all numbers are deleted from the documents before term extraction. The default is remove.numbers = TRUE.

remove.terms is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is remove.terms = NULL.

keep.terms is a character vector. It contains a list of compound words "formed by two or more terms" to keep in their original form in the term extraction process. The default is keep.terms = NULL.

synonyms is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is synonyms = NULL.

verbose is logical. If TRUE the function prints the most frequent terms extracted from documents. The default is verbose=TRUE.

Value

the bibliometric data frame with a new column containing terms about the field tag indicated in the argument Field.

See Also

c_convert2df to import and convert an WoS or SCOPUS Export file in a bibliographic data frame.

biblioAnalysis function for bibliometric analysis

Examples

# Example 1: Term extraction from titles

data(scientometrics, package = "bibliometrixData")

# vector of compound words
keep.terms <- c("co-citation analysis", "bibliographic coupling")

# term extraction
scientometrics <- termExtraction(scientometrics, Field = "TI", ngrams = 1,
remove.numbers=TRUE, remove.terms=NULL, keep.terms=keep.terms, verbose=TRUE)

# terms extracted from the first 10 titles
scientometrics$TI_TM[1:10]

#Example 2: Term extraction from abstracts

data(scientometrics)

# term extraction
thematicEvolution <- termExtraction(scientometrics, Field = "AB", ngrams = 2, stemming=TRUE, language="english", remove.numbers=TRUE, remove.terms=NULL, keep.terms=NULL, verbose=TRUE)

# terms extracted from the first abstract
scientometrics$AB_TM[1]

# Example 3: Term extraction from keywords with synonyms
data(scientometrics)

# vector of synonyms
synonyms <- c("citation; citation analysis", "h-index; index; impact factor")

# term extraction
scientometrics <- termExtraction(scientometrics, Field = "ID", ngrams = 1, synonyms=synonyms, verbose=TRUE)

---

thematicEvolution  
**Perform a Thematic Evolution Analysis**

**Description**

It performs a Thematic Evolution Analysis based on co-word network analysis and clustering. The methodology is inspired by the proposal of Cobo et al. (2011).

**Usage**

```r
thematicEvolution(
  M, 
  field = "ID", 
  years, 
  n = 250, 
  minFreq = 2, 
  size = 0.5, 
  ngrams = 1, 
  stemming = FALSE, 
  n.labels = 1, 
  repel = TRUE, 
  remove.terms = NULL, 
  synonyms = NULL, 
  cluster = "leading_eigen" 
)
```
Arguments

- **M** is a bibliographic data frame obtained by the converting function `convert2df`.
- **field** is a character object. It indicates the content field to use. Field can be one of `c("ID","DE","TI","AB")`. Default value is `field="ID"`.
- **years** is a numeric vector of two or more unique cut points.
- **n** is numerical. It indicates the number of words to use in the network analysis.
- **minFreq** is numerical. It indicates the min frequency of words included in to a cluster.
- **size** is numerical. It indicates del size of the cluster circles and is a number in the range (0.01,1).
- **ngrams** is an integer between 1 and 4. It indicates the type of n-gram to extract from texts. An n-gram is a contiguous sequence of n terms. The function can extract n-grams composed by 1, 2, 3 or 4 terms. Default value is `ngrams=1`.
- **stemming** is logical. If it is TRUE the word (from titles or abstracts) will be stemmed (using the Porter’s algorithm).
- **n.labels** is integer. It indicates how many labels associate to each cluster. Default is `n.labels = 1`.
- **repel** is logical. If it is TRUE ggplot uses `geom_label_repel` instead of `geom_label`.
- **remove.terms** is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is `remove.terms = NULL`.
- **synonyms** is a character vector. Each element contains a list of synonyms, separated by ",;", that will be merged into a single term (the first word contained in the vector element). The default is `synonyms = NULL`.
- **cluster** is a character. It indicates the type of cluster to perform among ("optimal", "louvain","leiden", "infomap","edge_betweenness","walktrap", "spinglass", "leading_eigen", "fast_greedy").

Details

`thematicEvolution` starts from two or more thematic maps created by `thematicMap` function.

Reference:

Value

- a list containing:
  - **nets** The thematic nexus graph for each comparison
  - **incMatrix** Some useful statistics about the thematic nexus
**thematicMap**

**See Also**

*thematicMap* function to create a thematic map based on co-word network analysis and clustering.

*cocMatrix* to compute a bibliographic bipartite network.

*networkPlot* to plot a bibliographic network.

**Examples**

```r
## Not run:
data(scientometrics, package = "bibliometrixData")
years=c(2000)

nexus <- thematicEvolution(scientometrics,field="ID", years=years, n=100,minFreq=2)
## End(Not run)
```

---

**thematicMap**  
*Create a thematic map*

**Description**

It creates a thematic map based on co-word network analysis and clustering. The methodology is inspired by the proposal of Cobo et al. (2011).

**Usage**

```r
thematicMap(
  M,
  field = "ID",
  n = 250,
  minfreq = 5,
  ngrams = 1,
  stemming = FALSE,
  size = 0.5,
  n.labels = 1,
  community.repulsion = 0.1,
  repel = TRUE,
  remove.terms = NULL,
  synonyms = NULL,
  cluster = "walktrap"
)
```
thematicMap

Arguments

M is a bibliographic dataframe.

field is the textual attribute used to build up the thematic map. It can be field = c("ID", "DE", "TI", "AB"). biblioNetwork or cocMatrix.

n is an integer. It indicates the number of terms to include in the analysis.

minfreq is an integer. It indicates the minimum frequency (per thousand) of a cluster. It is a number in the range (0,1000).

ngrams is an integer between 1 and 4. It indicates the type of n-gram to extract from texts. An n-gram is a contiguous sequence of n terms. The function can extract n-grams composed by 1, 2, 3 or 4 terms. Default value is ngrams=1.

stemming is logical. If it is TRUE the word (from titles or abstracts) will be stemmed (using the Porter’s algorithm).

size is numerical. It indicates del size of the cluster circles and is a number in the range (0.01,1).

n.labels is integer. It indicates how many labels associate to each cluster. Default is n.labels = 1.

community.repulsion is a real. It indicates the repulsion force among network communities. It is a real number between 0 and 1. Default is community.repulsion = 0.1.

repel is logical. If it is TRUE ggplot uses geom_label_repel instead of geom_label.

remove.terms is a character vector. It contains a list of additional terms to delete from the documents before term extraction. The default is remove.terms = NULL.

synonyms is a character vector. Each element contains a list of synonyms, separated by ";", that will be merged into a single term (the first word contained in the vector element). The default is synonyms = NULL.

cluster is a character. It indicates the type of cluster to perform among ("optimal", "louvain", "leiden", "infomap", "edge_betweenness", "walktrap", "spinglass", "leading_eigen", "fast_greedy").

Details

thematicMap starts from a co-occurrence keyword network to plot in a two-dimesional map the typological themes of a domain.

Reference:

Value

a list containing:
threeFieldsPlot

map The thematic map as ggplot2 object
clusters Centrality and Density values for each cluster.
words A list of words following in each cluster
nclust The number of clusters
net A list containing the network output (as provided from the networkPlot function)

See Also

biblioNetwork function to compute a bibliographic network.
cocMatrix to compute a bibliographic bipartite network.
networkPlot to plot a bibliographic network.

Examples

## Not run:
data(scientometrics, package = "bibliometrixData")
res <- thematicMap(scientometrics, field = "ID", n = 250, minfreq = 5, size = 0.5, repel = TRUE)
plot(res$map)
## End(Not run)

threeFieldsPlot Three Fields Plot

Description

Visualize the main items of three fields (e.g. authors, keywords, journals), and how they are related through a Sankey diagram.

Usage

threeFieldsPlot(M, fields = c("AU", "DE", "SO"), n = c(20, 20, 20))

Arguments

M is a bibliographic data frame obtained by the converting function convert2df. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file.

fields is a character vector. It indicates the fields to analyze using the standard WoS field tags. Default is fields = c("AU", "DE", "SO").

n is a integer vector. It indicates how many items to plot, for each of the three fields. Default is n = c(20, 20, 20)

Value

a sankeyPlot
**timeslice**

**Bibliographic data frame time slice**

**Description**
Divide a bibliographic data frame into time slice

**Usage**
timeslice(M, breaks = NA, k = 5)

**Arguments**
- **M**: is a bibliographic data frame obtained by the converting function `convert2df`. It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analytics WoS file.
- **breaks**: is a numeric vector of two or more unique cut points.
- **k**: is an integer value giving the number of intervals into which the data frame is to be cut. k is used only in case breaks argument is not provided. The default is k = 5.

**Value**
the value returned from split is a list containing the data frames for each sub-period.

**See Also**
- `convert2df` to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.
- `biblioAnalysis` function for bibliometric analysis.
- `summary` to obtain a summary of the results.
- `plot` to draw some useful plots of the results.

**Examples**

```r
#data(scientometrics, package = "bibliometrixData")
#threeFieldsPlot(scientometrics, fields=c("DE","AU","CR"),n=c(20,20,20))

list_df <- timeslice(scientometrics, breaks = c(1995, 2005))

names(list_df)
```


**trim**

*Deleting leading and ending white spaces*

---

**Description**

Deleting leading and ending white spaces from a character object.

**Usage**

```r
trim(x)
```

**Arguments**

`x` is a character object.

**Details**

`tableTag` is an internal routine of bibliometrics package.

**Value**

an object of class character

**Examples**

```r
char <- c(" Alfred", "Mary", " John")
char
trim(char)
```

---

**trim.leading**

*Deleting leading white spaces*

---

**Description**

Deleting leading white spaces from a character object.

**Usage**

```r
trim.leading(x)
```

**Arguments**

`x` is a character object.
Details

`tableTag` is an internal routine of `bibliometrics` package.

Value

an object of class `character`

Examples

```r
ch <- c(" Alfred", "Mary", " John")
char
trim.leading(char)
```

---

**trimES**

*Deleting extra white spaces*

Description

Deleting extra white spaces from a character object.

Usage

`trimES(x)`

Arguments

`x` is a character object.

Details

`tableTag` is an internal routine of `bibliometrics` package.

Value

an object of class `character`

Examples

```r
ch <- c("Alfred BJ", "Mary Beth", "John John")
char
trimES(char)
Index

* Clarivate Analytics Web of Science
  bibliometrix-package, 3
* Science Mapping
  bibliometrix-package, 3
* Scopus
  bibliometrix-package, 3
* citations
  bibliometrix-package, 3
* co-authors
  bibliometrix-package, 3
* co-citations
  bibliometrix-package, 3
* co-occurences
  bibliometrix-package, 3
* co-word analysis
  bibliometrix-package, 3
* collaboration
  bibliometrix-package, 3
* network
  bibliometrix-package, 3
authorProdOverTime, 6
biblioAnalysis, 7, 7, 10, 13, 14, 16, 18, 20, 23, 25–27, 30, 32, 34, 36, 38, 42, 43, 46, 51, 53, 54, 56, 58, 64
bibliometrix(bibliometrix-package), 3
bibliometrix-package, 3
biblioNetwork, 8, 9, 16, 18, 22, 29, 39–45, 51, 53, 62, 63
biblioshiny, 11
bibtag, 12
bradford, 12
citations, 13, 34
cocMatrix, 10, 14, 18, 22, 30, 42–45, 53, 61–63
communities, 41, 43
conceptualStructure, 16
count2df, 6–10, 13, 15–17, 19, 24, 25, 27, 28, 31–34, 36–38, 43, 48, 50, 51, 56–58, 60, 63, 64
countries, 20
couplingMap, 21, 21
dominance, 23
duplicatedMatching, 24
fieldByYear, 25
Hindex, 26
histNetwork, 28, 29, 30
histPlot, 29, 30
idByAuthor, 30, 49
keywordAssoc, 31
KeywordGrowth, 32
localCitations, 33
logo, 35
lotka, 35
Matrix, 10, 15
mergeDbSources, 36
metaTagExtraction, 37
net2VOSviewer, 38, 42, 53
networkPlot, 22, 38, 39, 39, 41, 47, 52, 53, 61, 63
networkStat, 42, 42
normalizeCitationScore, 43, 44
normalizeSimilarity, 44
plot, 7, 8, 14, 25, 27, 29, 32, 34, 36, 54, 64
plot.bibliometrix, 46
plotThematicEvolution, 47
readFiles, 48
retrievalByAuthorID, 31, 48
rpys, 50
sourceGrowth, 51
splitCommunities, 52, 52
stopwords, 53
summary, 7, 8, 13, 14, 24–28, 32, 34, 36, 46, 64
summary.bibliometrix, 54
summary.bibliometrix_netstat, 55

tableTag, 56
termExtraction, 18, 33, 57
thematicEvolution, 47, 59, 60
thematicMap, 47, 60, 61, 61
threeFieldsPlot, 63
timeslice, 64
trim, 65
trim.leading, 65
trimES, 66