

Package ‘bibliometrix’

May 7, 2026

Type Package

Title Comprehensive Science Mapping Analysis

Version 5.3.0

Description Tool for quantitative research in scientometrics and bibliometrics.

It implements the comprehensive workflow for science mapping analysis proposed in Aria M. and Cuccurullo C. (2017) <[doi:10.1016/j.joi.2017.08.007](https://doi.org/10.1016/j.joi.2017.08.007)>.

'bibliometrix' provides various routines for importing bibliographic data from 'SCOPUS',

'Clarivate Analytics Web of Science' (<<https://www.webofknowledge.com/>>),

'Digital Science Dimensions' (<<https://www.dimensions.ai/>>),

'OpenAlex' (<<https://openalex.org/>>),

'Cochrane Library' (<<https://www.cochranelibrary.com/>>),

'Lens' (<<https://lens.org/>>),

and 'PubMed' (<<https://pubmed.ncbi.nlm.nih.gov/>>)

databases, performing bibliometric analysis

and building networks for co-citation, coupling, scientific collaboration and co-word analysis.

License GPL-3

URL <https://www.bibliometrix.org>,

<https://github.com/massimoaria/bibliometrix>,

<https://www.k-synth.com>

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

LazyData true

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, grDevices, bibliometrixData, contentanalysis (>=

1.0.0), dimensionsR, dplyr, ca, forcats, ggplot2, ggrepel,

igraph, Matrix, methods, plotly, openalexR, openxlsx, pubmedR,

purrr, readr, readxl, rscopus, shiny, shinycssloaders (>=

1.1.0), SnowballC, stringdist, stringi, stringr, tibble, tidyR,

tidytext, visNetwork

Suggests knitr, ragg, rmarkdown, testthat (>= 3.0.0), wordcloud2

RoxygenNote 7.3.3

NeedsCompilation no

Config/testthat/edition 3

Author Massimo Aria [cre, aut, cph] (ORCID:
<https://orcid.org/0000-0002-8517-9411>),
 Corrado Cuccurullo [aut] (ORCID:
<https://orcid.org/0000-0002-7401-8575>)

Maintainer Massimo Aria <aria@unina.it>

Repository CRAN

Date/Publication 2026-04-09 05:10:50 UTC

Contents

bibliometrix-package	3
applyCitationMatching	7
applyReferenceMatching	7
assignEvolutionColors	10
authorBio	12
authorProdOverTime	13
biblioAnalysis	14
biblioNetwork	16
biblioshiny	18
bibttag	19
bradford	20
citations	21
cocMatrix	22
collabByRegionPlot	24
conceptualStructure	27
convert2df	29
countries	31
couplingMap	32
customTheme	34
dominance	34
duplicatedMatching	35
fieldByYear	36
findAuthorWorks	38
get_authors_summary	39
Hindex	40
histNetwork	41
histPlot	43
idByAuthor	44
keywordAssoc	45
KeywordGrowth	46
lifeCycle	47
loadBookDataset	48
localCitations	49
logo	50

lotka	50
ltwa	51
mergeDbSources	52
mergeKeywords	53
metaTagExtraction	54
missingData	55
net2Pajek	56
net2VOSviewer	57
networkPlot	58
networkStat	61
normalizeCitationScore	62
normalizeSimilarity	63
normalize_citations	65
plot.bibliodendrogram	67
plot.bibliometrix	68
plotThematicEvolution	69
print_author_works_summary	70
readFiles	70
retrievalByAuthorID	71
rpys	73
sourceGrowth	74
splitCommunities	75
stopwords	76
summary.bibliometrix	76
summary.bibliometrix_netstat	77
tableTag	78
termExtraction	79
thematicEvolution	82
thematicMap	84
threeFieldsPlot	86
timeslice	87
trim	88
trim.leading	88
trimES	89
Index	90

bibliometrix-package *Comprehensive Science Mapping Analysis*

Description

Tool for quantitative research in scientometrics and bibliometrics. It implements the comprehensive workflow for science mapping analysis proposed in Aria M. and Cuccurullo C. (2017) <doi:10.1016/j.joi.2017.08.007>. 'bibliometrix' provides various routines for importing bibliographic data from 'SCOPUS', 'Clarivate Analytics Web of Science' (<<https://www.webofknowledge.com/>>), 'Digital Science Dimensions' (<<https://www.dimensions.ai/>>), 'OpenAlex' (<<https://openalex.org/>>), 'Cochrane Library' (<<https://www.cochranelibrary.com/>>), 'Lens' (<<https://lens.org/>>), and 'PubMed'

(<<https://pubmed.ncbi.nlm.nih.gov/>>) databases, performing bibliometric analysis and building networks for co-citation, coupling, scientific collaboration and co-word analysis.

Details

INSTALLATION

- Stable version from CRAN:

```
install.packages("bibliometrix")
```

- Or development version from GitHub:

```
install.packages("devtools") devtools::install_github("massimoaria/bibliometrix")
```

- Load "bibliometrix"

```
library('bibliometrix')
```

DATA LOADING AND CONVERTING

The export file can be imported and converted by R using the function `*convert2df*`:

```
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:

```
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.

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

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

```
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:

```
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.

```
### 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
```

```
net=networkPlot(NetMatrix, n = dim(NetMatrix)[1], Title = "Country Collaboration", type = "circle", size=TRUE, remove.multiple=FALSE,labels=0.8)
```

```
### 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, labels=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,labels=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, stem-
ming=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)

Massimo Aria [cre, aut, cph] (ORCID: <<https://orcid.org/0000-0002-8517-9411>>), Corrado Cuccurullo [aut] (ORCID: <<https://orcid.org/0000-0002-7401-8575>>)

Maintainer: Massimo Aria <aria@unina.it>

References

Aria, M. & Cuccurullo, C. (2017). `*bibliometrix*`: An R-tool for comprehensive science mapping analysis, **Journal of Informetrics**, 11(4), pp 959-975, Elsevier, DOI: 10.1016/j.joi.2017.08.007 (<https://doi.org/10.1016/j.joi.2017.08.007>).

Cuccurullo, C., Aria, M., & Sarto, F. (2016). Foundations and trends in performance management. A twenty-five years bibliometric analysis in business and public administration domains, **Scientometrics**, DOI: 10.1007/s11192-016-1948-8 (<https://doi.org/10.1007/s11192-016-1948-8>).

Cuccurullo, C., Aria, M., & Sarto, F. (2015). Twenty years of research on performance management in business and public administration domains. Presentation at the `*Correspondence Analysis and Related Methods conference (CARME 2015)*` in September 2015 (https://www.bibliometrix.org/documents/2015Carne_cuc

Sarto, F., Cuccurullo, C., & Aria, M. (2014). Exploring healthcare governance literature: systematic review and paths for future research. **Mecosan** (https://www.francoangeli.it/Riviste/Scheda_Rivista.aspx?IDarticolo=5278

Cuccurullo, C., Aria, M., & Sarto, F. (2013). Twenty years of research on performance management in business and public administration domains. In **Academy of Management Proceedings** (Vol. 2013, No. 1, p. 14270). Academy of Management (<https://doi.org/10.5465/AMBPP.2013.14270abstract>).

applyCitationMatching *applyCitationMatching (Deprecated)*

Description

Deprecated in bibliometrix 6.0. This function has been renamed to [applyReferenceMatching](#). Please use `applyReferenceMatching()` instead.

Usage

```
applyCitationMatching(M, threshold = 0.9, method = "jw", min_chars = 20)
```

Arguments

M	A bibliometrix data frame, typically created by convert2df . Must contain the columns: <ul style="list-style-type: none">• SR: Short reference identifier for each document• CR: Cited references field (citations separated by semicolons)• DB: (Optional) Database source identifier for format detection
threshold	Numeric value between 0 and 1 indicating the similarity threshold for matching citations. Default is 0.85. See normalize_citations for details on selecting appropriate thresholds.
method	String distance method to use for fuzzy matching. Options include: <ul style="list-style-type: none">• "jw" (default): Jaro-Winkler distance, optimized for bibliographic strings• "lv": Levenshtein distance• Other methods supported by stringdistmatrix
min_chars	Minimum characters for valid citations (default: 20)

Value

Same as [applyReferenceMatching](#).

applyReferenceMatching

Apply reference normalization to bibliometrix data frame

Description

This is a convenience wrapper function that applies [normalize_citations](#) to a bibliometrix data frame (typically loaded with [convert2df](#)). It extracts citations from the CR field, performs normalization and matching, and returns comprehensive results including per-paper citation lists and summary statistics.

Usage

```
applyReferenceMatching(M, threshold = 0.9, method = "jw", min_chars = 20)
```

Arguments

M	A bibliometrix data frame, typically created by convert2df . Must contain the columns: <ul style="list-style-type: none">• SR: Short reference identifier for each document• CR: Cited references field (citations separated by semicolons)• DB: (Optional) Database source identifier for format detection
threshold	Numeric value between 0 and 1 indicating the similarity threshold for matching citations. Default is 0.85. See normalize_citations for details on selecting appropriate thresholds.
method	String distance method to use for fuzzy matching. Options include: <ul style="list-style-type: none">• "jw" (default): Jaro-Winkler distance, optimized for bibliographic strings• "lv": Levenshtein distance• Other methods supported by stringdistmatrix
min_chars	Minimum characters for valid citations (default: 20)

Details

The function automatically handles the new Scopus citation format (where the year appears at the end in parentheses) by converting it to the classic format before processing.

The function performs the following steps:

1. Splits the CR field by semicolons to extract individual citations
2. Detects and converts new Scopus format citations to classic format
3. Trims whitespace from each citation
4. Applies [normalize_citations](#) to identify duplicate citations
5. Links normalized citations back to source documents (SR)
6. Generates summary statistics and reconstructs normalized CR fields

The normalized CR field can be used to replace the original CR field in subsequent bibliometric analyses, ensuring that citation counts and network analyses are not inflated by duplicate citations with minor formatting differences.

Value

A list with four elements:

full_data A data frame with columns:

- SR: Source document identifier
- CR: Original citation string
- CR_canonical: Canonical (normalized) citation
- cluster_id: Unique cluster identifier

- `n_cluster`: Size of the citation cluster
- `first_author`, `year`, `journal`, `volume`: Extracted metadata

summary A data frame summarizing citation frequencies with columns:

- `CR_canonical`: The canonical citation for each cluster
- `n`: Total number of times this work was cited
- `n_variants`: Number of different formatting variants found
- `variants_example`: Sample of variant formats (up to 3 examples)

Sorted by citation frequency (`n`) in descending order.

matched_citations Complete output from [normalize_citations](#), useful for detailed analysis of the matching process.

CR_normalized A data frame with columns:

- `SR`: Source document identifier
- `CR`: Reconstructed CR field with normalized citations (semicolon-separated)
- `n_references`: Number of unique references after normalization

This can be merged back with `M` to replace the original CR field.

References

Aria, M. & Cuccurullo, C. (2017). *bibliometrix*: An R-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4), 959-975.

See Also

[normalize_citations](#) for the underlying normalization algorithm [citations](#) for citation analysis
[localCitations](#) for local citation analysis

Examples

```
## Not run:
# Load bibliometric data
file <- "https://www.bibliometrix.org/datasets/savedrecs.txt"
M <- convert2df(file, dbsource = "wos", format = "plaintext")

# Apply reference normalization
results <- applyReferenceMatching(M, threshold = 0.85)

# View top cited works (after normalization)
head(results$summary, 20)

# See how many variants were found for the top citation
top_citation <- results$summary$CR_canonical[1]
variants <- subset(results$full_data, CR_canonical == top_citation)
unique(variants$CR)

# Replace original CR with normalized CR in the data frame
M_normalized <- M %>%
  rename(CR_orig = CR) %>%
  left_join(results$CR_normalized, by = "SR")
```

```

# Compare citation counts before and after normalization
original_citations <- strsplit(M$CR, ";") %>%
  unlist() %>%
  trimws() %>%
  table() %>%
  length()

normalized_citations <- nrow(results$summary)

cat("Original unique citations:", original_citations, "\n")
cat("After normalization:", normalized_citations, "\n")
cat("Duplicates found:", original_citations - normalized_citations, "\n")

# Use normalized data for further analysis
CR_analysis <- citations(M_normalized, field = "article", sep = ";")

## End(Not run)

```

assignEvolutionColors *Assign Colors to Thematic Evolution Nodes Based on Lineages*

Description

This function assigns colors to nodes in a thematic evolution analysis based on their lineages across time periods. Nodes connected by strong edges (above a threshold) receive the same color to visualize thematic continuity. Nodes with the same name across periods that are not strongly connected to other nodes are also colored identically.

Usage

```
assignEvolutionColors(nexus, threshold = 0.5, palette = NULL, alpha = 0.5)
```

Arguments

nexus	A list object returned by <code>thematicEvolution</code> containing: <ul style="list-style-type: none"> • Nodes: data frame with node information (name, group, id, sum, freq, etc.) • Edges: data frame with edge information (from, to, weight measures) • TM: list of thematic maps for each period
threshold	Numeric. The minimum weight value for an edge to be considered a "strong connection" (default: 0.5). Edges with weights \geq threshold will propagate the same color to connected nodes across periods.
palette	Character vector. Optional custom color palette as hex codes. If NULL, uses a default palette of 50+ distinct colors. Colors are assigned sequentially without reuse.
alpha	Numeric. Balancing parameters between inclusion and importance measures. Default value is 0.5.

Details

The function uses a multi-phase algorithm:

1. **Phase 1:** Identifies lineages by following strong connections (weight \geq threshold) from the first period forward. When a node has multiple strong connections, the strongest one determines the lineage.
2. **Phase 1.5:** Assigns the same lineage to nodes with identical names across periods if they are not already part of different strong connections.
3. **Phase 2:** Assigns unique colors from the palette to each identified lineage.
4. **Phase 3:** Assigns unique colors to isolated nodes (those without any lineage).
5. **Phase 4:** Colors edges based on their connected nodes - same color if both nodes share a color, grey otherwise.
6. **Final:** Updates thematic maps with the new color scheme.

Each lineage receives a unique color from the palette. No color is reused across different lineages, ensuring clear visual distinction between independent thematic streams.

Value

Returns the modified nexus object with updated colors:

- Nodes\$color: updated with lineage-based colors
- Edges\$color: edges connecting same-colored nodes receive the node color, others are grey
- TM: thematic maps updated with new cluster colors

See Also

[thematicEvolution](#) to perform thematic evolution analysis.

[plotThematicEvolution](#) to visualize the colored evolution.

Examples

```
## Not run:
data(scientometrics, package = "bibliometrixData")
years <- c(2000, 2010)

nexus <- thematicEvolution(scientometrics, field = "ID",
                           years = years, n = 100, minFreq = 2)

# Use custom threshold and measure
nexus <- assignEvolutionColors(nexus, threshold = 0.6, use_measure = "weighted")

## End(Not run)
```

authorBio

*Retrieve Author Biographical Information from OpenAlex***Description**

This function downloads comprehensive author information from OpenAlex based on a DOI and the numerical position of the author in the co-authors list. It provides detailed biographical data, bibliometric indicators, and affiliation information.

Usage

```
authorBio(
  author_position = 1,
  doi = "10.1016/j.joi.2017.08.007",
  verbose = FALSE,
  return_all_authors = FALSE,
  sleep_time = 1,
  max_retries = 3,
  retry_delay = 2,
  mailto = NULL
)
```

Arguments

author_position	Integer. The numerical position of the author in the authors list (default: 1)
doi	Character. DOI of the article used to identify the authors
verbose	Logical. Print informative messages during execution (default: FALSE)
return_all_authors	Logical. If TRUE, returns information for all co-authors (default: FALSE)
sleep_time	Numeric. Seconds to wait between API calls to respect rate limits (default: 1)
max_retries	Integer. Maximum number of retry attempts for failed API calls (default: 3)
retry_delay	Numeric. Base delay in seconds before retrying after an error (default: 2)
mailto	Character. Email address for OpenAlex polite pool (default: NULL, uses openalexR.mailto option if set)

Details

The function first retrieves the work information using the provided DOI, then extracts author IDs from the authorships data, and finally fetches detailed author profiles from OpenAlex. It enriches the author data with paper-specific information such as authorship position, corresponding author status, and affiliations as listed in the paper.

The function implements automatic retry logic with exponential backoff to handle rate limiting (HTTP 429 errors) and temporary network issues. It respects OpenAlex API rate limits by adding configurable delays between requests.

IMPORTANT: For better rate limits, set your OpenAlex API key using: `Sys.setenv(openalexR_apikey = "YOUR_API_KEY")` Get a free API key at: <https://openalex.org/>

Value

If `return_all_authors = FALSE`, returns a tibble with comprehensive information about the specified author including:

- Basic information (name, ORCID, OpenAlex ID)
- Bibliometric indicators (works count, citations, h-index, i10-index)
- Affiliation details from both the paper and author profile
- Research topics and areas
- Paper-specific metadata (corresponding author status, position type)

If `return_all_authors = TRUE`, returns a list of tibbles, one for each co-author.

Examples

```
## Not run:
# Get information for the first author
first_author <- authorBio(doi = "10.1016/j.joi.2017.08.007")

# Get information for the second author with verbose output
second_author <- authorBio(
  author_position = 2,
  doi = "10.1016/j.joi.2017.08.007",
  verbose = TRUE
)

# Get information for all co-authors with custom rate limiting
all_authors <- authorBio(
  doi = "10.1016/j.joi.2017.08.007",
  return_all_authors = TRUE,
  sleep_time = 0.5,
  max_retries = 5
)

## End(Not run)
```

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:

<code>dfAU</code>	is a data frame
<code>dfpapersAU</code>	is a data frame
<code>graph</code>	a ggplot object

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)
```

`biblioAnalysis`

Bibliometric Analysis

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
nAUpaper	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 An

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,
  sep = ";",
  short = FALSE,
  shortlabel = TRUE,
  remove.terms = NULL,
  synonyms = 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", "all_k", "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")
- 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:

- Batagelj, V., & Cerinsek, M. (2013). On bibliographic networks. *Scientometrics*, 96(3), 845-864.
- Aria, M., & Cuccurullo, C. (2017). *bibliometrix: An R-tool for comprehensive science mapping analysis*. *Journal of Informetrics*, 11(4), 959-975.

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", labelsSize=0.5)

# EXAMPLE 2: Co-citation network

data(scientometrics, package = "bibliometrixData")

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

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

biblioshiny

Shiny UI for bibliometrix package

Description

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

Usage

```
biblioshiny(
  host = "127.0.0.1",
  port = 3838,
  launch.browser = TRUE,
```

```

    maxUploadSize = 500,
    max.rows = Inf
  )

```

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 the default port 3838.
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)
max.rows	is an integer or Inf. The maximum number of records (publications) allowed when importing or loading a file. Files exceeding this limit will be rejected with an informative message. Default value is Inf (no limit).

Examples

```
# biblioshiny()
```

bibtag	<i>Tag list and bibtex fields.</i>
--------	------------------------------------

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:

TAG Tag Fields

SCOPUS Scopus bibtex fields

ISI WOS/ISI bibtex fields

GENERIC Generic bibtex fields

DIMENSIONS_OLD DIMENSIONS cvs/xlsx old fields

DIMENSIONS DIMENSIONS cvs/xlsx fields

bradford

*Bradford's law***Description**

It estimates Bradford's law source distribution and tests the goodness of fit.

Usage

```
bradford(M)
```

Arguments

M is a bibliographic dataframe.

Details

Bradford's Law of Scattering, first formulated by Samuel C. Bradford in 1934, describes the phenomenon of concentration and dispersion in scientific publishing: a small number of core journals account for a disproportionately large share of the literature on a given topic, while the remaining literature is scattered across an increasingly large number of peripheral journals.

If journals are ranked in decreasing order of productivity and partitioned into three zones, each containing roughly one-third of the total articles, the number of journals in each zone follows the ratio $1:n:n^2$, where n is the Bradford multiplier.

The Bradford distribution models the cumulative number of articles $C(r)$ contributed by the top r sources as: $C(r) = a + b * \log(r)$

Reference:

Bradford, S. C. (1934). Sources of information on specific subjects. *Engineering*, 137, 85-86.

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 Bradford bibliograph plot in ggplot2 format
graph_shiny	the Bradford bibliograph plot for biblioshiny (without logo)
zoneSummary	a dataframe summarizing the three Bradford zones
stat	a list of statistical results (coefficients, R2, KS test, Bradford multiplier)

See Also

[biblioAnalysis](#) function for bibliometric analysis

[summary](#) method for class 'bibliometrix'

Examples

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

BR <- bradford(management)

## End(Not run)
```

citations	<i>Citation frequency distribution</i>
-----------	--

Description

It calculates frequency distribution of citations.

Usage

```
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:

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

Bibliographic bipartite network matrices

Description

cocMatrix computes 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,
  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: <table> <tr> <td>AU</td> <td>Authors</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>DE</td> <td>Author Keywords</td> </tr> <tr> <td>ID</td> <td>Keywords associated by WoS or SCOPUS database</td> </tr> <tr> <td>KW_Merged</td> <td>All Keywords (merged by DE and ID)</td> </tr> <tr> <td>CR</td> <td>Cited References</td> </tr> </table>	AU	Authors	SO	Publication Name (or Source)	JI	ISO Source Abbreviation	DE	Author Keywords	ID	Keywords associated by WoS or SCOPUS database	KW_Merged	All Keywords (merged by DE and ID)	CR	Cited References
AU	Authors														
SO	Publication Name (or Source)														
JI	ISO Source Abbreviation														
DE	Author Keywords														
ID	Keywords associated by WoS or SCOPUS database														
KW_Merged	All Keywords (merged by DE and ID)														
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 <code>matrix</code>
type = "sparse"	produces an object of class <code>dgMatrix</code> of the package <code>Matrix</code> . "sparse" argument generates a compact matrix
n	is an integer. It indicates the number of items to select. If <code>N = NULL</code> , all items are selected.
sep	is the field separator character. This character separates strings in each column of the data frame. The default is <code>sep = ";"</code> .
binary	is a logical. If <code>TRUE</code> each cell contains a 0/1. if <code>FALSE</code> each cell contains the frequency.
short	is a logical. If <code>TRUE</code> all items with <code>frequency < 2</code> 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 <code>remove.terms = NULL</code> .
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 <code>synonyms = NULL</code> .

Details

This occurrence matrix represents a bipartite network which can be transformed into a collection of bibliographic networks such as coupling, co-citation, etc..

The function follows the approach proposed by Batagelj & Cerinsek (2013) and Aria & cuccurullo (2017).

References:

Batagelj, V., & Cerinsek, M. (2013). On bibliographic networks. *Scientometrics*, 96(3), 845-864.
Aria, M., & Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4), 959-975.

Value

a bipartite network 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 occurrence matrix

data(scientometrics, package = "bibliometrixData")
WA <- cocMatrix(scientometrics, Field = "AU", type = "sparse", sep = ";")

# EXAMPLE 2: Articles x Cited References occurrence matrix

# data(scientometrics, package = "bibliometrixData")

# WCR <- cocMatrix(scientometrics, Field = "CR", type = "sparse", sep = ";")

# EXAMPLE 3: Articles x Cited First Authors occurrence matrix

# data(scientometrics, package = "bibliometrixData")
# scientometrics <- metaTagExtraction(scientometrics, Field = "CR_AU", sep = ";")
# WCR <- cocMatrix(scientometrics, Field = "CR_AU", type = "sparse", sep = ";")
```

collabByRegionPlot *Country Collaboration Networks by Region*

Description

A function to create and plot country collaboration networks by Region

Usage

```
collabByRegionPlot(
  NetMatrix,
  normalize = NULL,
  n = NULL,
  degree = NULL,
  type = "auto",
  label = TRUE,
  labelsize = 1,
  label.cex = FALSE,
  label.color = FALSE,
  label.n = Inf,
  halo = FALSE,
  cluster = "walktrap",
  community.repulsion = 0,
  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 country collaboration 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.
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

<code>label</code>	is logical. If TRUE vertex labels are plotted.
<code>labelsize</code>	is an integer. It indicates the label size in the plot. Default is <code>labelsize=1</code>
<code>label.cex</code>	is logical. If TRUE the label size of each vertex is proportional to its degree.
<code>label.color</code>	is logical. If TRUE, for each vertex, the label color is the same as its cluster.
<code>label.n</code>	is an integer. It indicates the number of vertex labels to draw.
<code>halo</code>	is logical. If TRUE communities are plotted using different colors. Default is <code>halo=FALSE</code>
<code>cluster</code>	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").
<code>community.repulsion</code>	is a real. It indicates the repulsion force among network communities. It is a real number between 0 and 1. Default is <code>community.repulsion = 0.1</code> .
<code>vos.path</code>	is a character indicating the full path where VOSviewer.jar is located.
<code>size</code>	is integer. It defines the size of each vertex. Default is <code>size=3</code> .
<code>size.cex</code>	is logical. If TRUE the size of each vertex is proportional to its degree.
<code>curved</code>	is a logical or a number. If TRUE edges are plotted with an optimal curvature. Default is <code>curved=FALSE</code> . Curved values are any numbers from 0 to 1.
<code>noloops</code>	is logical. If TRUE loops in the network are deleted.
<code>remove.multiple</code>	is logical. If TRUE multiple links are plotted using just one edge.
<code>remove.isolates</code>	is logical. If TRUE isolates vertices are not plotted.
<code>weighted</code>	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 <code>weighted</code> argument. If it is TRUE then a weighted graph is created and the name of the edge attribute will be <code>weight</code> .
<code>edgesize</code>	is an integer. It indicates the network edge size.
<code>edges.min</code>	is an integer. It indicates the min frequency of edges between two vertices. If <code>edge.min=0</code> , all edges are plotted.
<code>alpha</code>	is a number. Legal alpha values are any numbers from 0 (transparent) to 1 (opaque). The default alpha value usually is 0.5.
<code>verbose</code>	is a logical. If TRUE, network will be plotted. Default is <code>verbose = TRUE</code> .

Value

It is a list containing the following elements:

<code>graph</code>	a network object of the class <code>igraph</code>
<code>cluster_obj</code>	a <code>communities</code> object of the package <code>igraph</code>

cluster_res a data frame with main results of clustering procedure.

Examples

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

management <- metaTagExtraction(management, Field = "AU_CO")

NetMatrix <- biblioNetwork(management, analysis = "collaboration", network = "countries")

net <- collabByRegionPlot(NetMatrix,
  edgesize = 4, label.cex = TRUE, labelsize = 2.5,
  weighted = TRUE, size = 0.5, size.cex = TRUE, community.repulsion = 0,
  verbose = FALSE
)

cbind(names(net))

plot(net[[4]]$graph)

## End(Not run)
```

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 key-word, title or abstract fields.

Usage

```
conceptualStructure(
  M,
  field = "ID",
  ngrams = 1,
  method = "MCA",
  quali.suppl = NULL,
  quanti.suppl = 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 <code>convert2df</code> . 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
KW_Merged	All 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.sup	is a vector indicating the indexes of the categorical supplementary variables. It is used only for CA and MCA.
quanti.sup	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 analyze and plot. The default value is 2.
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.
labelsize	is an integer. It indicates the label size in the plot. Default is labelsize=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:

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

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",
  remove.duplicates = TRUE
)
```

Arguments

- file** a character array containing a sequence of filenames coming from WoS, Scopus, Dimensions, Lens.org, OpenAlex and Pubmed. Alternatively, file can be an object resulting from an API query fetched from Dimensions, and PubMed databases:
- a) 'wos' Clarivate Analytics WoS (in plaintext '.txt', Endnote Desktop '.ciw', or bibtex formats '.bib');
 - b) 'scopus' SCOPUS (exclusively in bibtex format '.bib');
 - c) 'dimensions' Digital Science Dimensions (in csv '.csv' or excel '.xlsx' formats);
 - d) 'lens' Lens.org (in csv '.csv');
 - e) 'pubmed' an object of the class pubmedR (package pubmedR) containing a collection obtained from a query performed on PubMed;
 - f) 'dimensions' an object of the class dimensionsR (package dimensionsR) containing a collection obtained from a query performed on Dimensions;
 - g) 'openalex' OpenAlex .csv file;
 - h) 'openalex_api' the filename and path to a list object returned by openalexR package, containing a collection of works returned from an API query.
- dbsource** is a character indicating the bibliographic database. dbsource can be dbsource = c('cochrane', 'dimensions', 'generic', 'isi', 'openalex', 'pubmed', 'scopus', 'wos', 'lens'). Default is dbsource = "isi".
- format** is a character indicating the SCOPUS, Clarivate Analytics WoS, and other databases export file format. format can be c('api', 'bibtex', 'csv', 'endnote', 'excel', 'plaintext', 'pubmed'). Default is format = "plaintext".
- remove.duplicates** is logical. If TRUE, the function will remove duplicated items checking by DOI and database ID.

Value

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

I.e We have three files download 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:

AU	Authors
TI	Document Title
SO	Publication Name (or Source)
JI	ISO Source Abbreviation

DT	Document Type
DE	Authors' Keywords
ID	Keywords associated by SCOPUS or WoS database
AB	Abstract
C1	Author Address
RP	Reprint Address
CR	Cited References
TC	Times Cited
PY	Year
SC	Subject Category
UT	Unique Article Identifier
DB	Database

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 202 rows and 5 variables:

countries country names

continent continent names

iso2 country ISO 3166-1 alpha-2 code

Longitude country centroid longitude

Latitude country centroid latitude

couplingMap

*Coupling Analysis***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 impact.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 <code>community.repulsion = 0.1</code> .
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 <code>n.labels = 1</code> .
repel	is logical. If it is TRUE ggplot uses <code>geom_label_repel</code> instead of <code>geom_label</code> .
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 <code>networkPlot</code> 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(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)
```

customTheme	<i>Custom Theme variables for Biblioshiny.</i>
-------------	--

Description

List containing a set of custom theme variables for Biblioshiny.

Format

A list with 3 elements:

name object name
attribs attributes
children CSS style

dominance	<i>Authors' dominance ranking</i>
-----------	-----------------------------------

Description

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

Usage

```
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	Author's name
Dominance Factor	Dominance Factor ($DF = FAA / MAA$)
Tot Articles	N. of Authored Articles (TAA)
Single Authored	N. of Single-Authored Articles (SAA)
Multi Authored	N. of Multi-Authored Articles ($MAA=TAA-SAA$)
First Authored	N. of First Authored Articles (FAA)
Rank by Articles	Author Ranking by N. of Articles
Rank by DF	Author Ranking by Dominance Factor

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 <code>exact = TRUE</code> the function searches duplicates using exact matching. If <code>exact=FALSE</code> , 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.

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

```
data(scientometrics, package = "bibliometrixData")  
  
M <- rbind(scientometrics[1:20, ], scientometrics[10:30, ])  
  
newM <- duplicatedMatching(M, Field = "TI", exact = FALSE, tol = 0.95)  
  
dim(newM)
```

fieldByYear

Field Tag distribution by Year

Description

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

Usage

```
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.
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. I indicates the maximum number of items per year to include in the plot.
labelsize	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 Filed 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

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

findAuthorWorks

Find Author's Co-authored Works

Description

Searches for an author's name in a bibliometric dataframe and returns the DOIs and author positions of their co-authored works.

Usage

```
findAuthorWorks(author_name, data, partial_match = TRUE, exact_match = FALSE)
```

Arguments

author_name	Character. The author's name to search for (case-insensitive)
data	Data.frame. The bibliometric dataframe with AU and DI columns
partial_match	Logical. If TRUE, allows partial name matching (default: TRUE)
exact_match	Logical. If TRUE, requires exact name matching (default: FALSE)

Details

The function searches through the AU column which contains author names separated by semi-colons. It identifies the position of the target author and returns comprehensive information about each matching work.

Value

A data.frame with columns:

- doi: DOI of the work
- author_position: Numerical position of the author in the author list
- total_authors: Total number of authors in the work
- all_authors: Complete list of authors for reference
- matched_name: The exact name variant that was matched

Author(s)

Your Name

Examples

```
## Not run:
# Find works by "ARIA M"
works <- findAuthorWorks("ARIA M", M)

# Find works with exact matching
works_exact <- findAuthorWorks("PESTANA MH", M, exact_match = TRUE)

# Find works with partial matching disabled
works_full <- findAuthorWorks("MASSIMO ARIA", M, partial_match = FALSE)

## End(Not run)
```

get_authors_summary *Get Authors Summary from OpenAlex*

Description

Retrieves a quick summary of all authors from a paper without making additional API calls for individual author profiles. Useful for getting an overview of the authorship structure.

Usage

```
get_authors_summary(
  doi = "10.1016/j.joi.2017.08.007",
  verbose = FALSE,
  sleep_time = 0.2,
  max_retries = 3,
  mailto = NULL
)
```

Arguments

doi	Character. DOI of the article
verbose	Logical. Print informative messages during execution (default: FALSE)
sleep_time	Numeric. Seconds to wait before API call (default: 0.2)
max_retries	Integer. Maximum number of retry attempts (default: 3)
mailto	Character. Email address for the OpenAlex API polite pool (default: NULL)

Value

A data frame with summary information for all authors including:

- position: Author position in the paper
- display_name: Author name as it appears in the paper

- `author_position_type`: Type of position (first, last, middle)
- `is_corresponding`: Whether the author is a corresponding author
- `orcid`: ORCID identifier if available
- `openalex_id`: OpenAlex author identifier
- `primary_affiliation`: Main institutional affiliation

Examples

```
## Not run:
# Get a quick summary of all authors
summary <- get_authors_summary(doi = "10.1016/j.joi.2017.08.007")
print(summary)

## End(Not run)
```

Hindex

h-index calculation

Description

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

Usage

```
Hindex(M, field = "author", elements = NULL, sep = ";", years = Inf)
```

Arguments

<code>M</code>	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.
<code>field</code>	is character. It can be equal to <code>c("author", "source")</code> . <code>field</code> indicates if H-index have to be calculated for a list of authors or for a list of sources. Default value is <code>field = "author"</code> .
<code>elements</code>	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 <code>C("SURNAME1 N", "SURNAME2 N", ...)</code> , in other words, for each author: surname and initials separated by one blank space. If <code>elements=NULL</code> , the function calculates impact indices for all elements contained in the data frame. i.e for the authors SEMPRONIO TIZIO CAIO and ARIA MASSIMO <code>elements</code> argument is <code>elements = c("SEMPRONIO TC", "ARIA M")</code> .
<code>sep</code>	is the field separator character. This character separates authors in each string of AU column of the bibliographic data frame. The default is <code>sep = ";"</code> .
<code>years</code>	is a integer. It indicates the number of years to consider for Hindex calculation. Default is <code>Inf</code> .

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

```
### 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

```
histNetwork(M, min.citations, 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, OpenAlex, Lens.org and Clarivate Analytics Web of Science file.
min.citations	DEPRECATED. New algorithm does not use this parameters. It will be remove in the next version of bibliometrix.
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 a supported 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

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

histResults <- histNetwork(management, sep = ";")

## End(Not run)
```

histPlot	<i>Plotting historical co-citation network</i>
----------	--

Description

histPlot plots a historical co-citation network.

Usage

```
histPlot(
  histResults,
  n = 20,
  size = 5,
  labelsSize = 5,
  remove.isolates = TRUE,
  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

is a network matrix obtained by the function [histNetwork](#).

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.

labelsSize is an integer. It indicates the label size in the plot. Default is labelsSize=5.

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

title.as.label is a logical. DEPRECATED

label is a character. It indicates which label type to use as node id in the historiograph. It can be label=c("short", "title", "keywords", "keywordsplus"). Default is label = "short".

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

Details

The function [histPlot](#) can plot a historical co-citation network previously created by [histNetwork](#).

Value

It is list containing: a network object of the class `igraph` and a plot object of the class `ggraph`.

See Also

[histNetwork](#) to compute a historical co-citation network.

[cocMatrix](#) to compute a co-occurrence matrix.

[biblioAnalysis](#) to perform a bibliometric analysis.

Examples

```
# EXAMPLE Citation network
## Not run:
data(management, package = "bibliometrixData")

histResults <- histNetwork(management, sep = ";")

net <- histPlot(histResults, n = 20, labelsize = 5)

## End(Not run)
```

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:

<code>lastname</code>	author's last name
<code>firstname</code>	author's first name
<code>affiliation</code>	Part of the affiliation name (university name, city, etc.)

i.e. `df[1,1:3]<-c("aria","massimo","naples")` When affiliation is not specified, the field `df$affiliation` have to be NA. i.e. `df[2,1:3]<-c("cuccurullo","corrado", NA)`

`api_key` is a character. It contains the Elsevier API key. Information about how to obtain an API Key [Elsevier API website](#)

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

```
## 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)
```

keywordAssoc

ID and DE keyword associations

Description

It associates authors' keywords to keywords plus.

Usage

```
keywordAssoc(M, sep = ";", n = 10, excludeKW = NA)
```

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.
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 a integer. It indicates the number of authors' keywords to associate to each keyword plus. The default is n = 10.
excludeKW	is 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
)
```

Arguments

M	is a data frame obtained by the converting function <code>convert2df</code> . 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, DE, KW_Merged) or a field tag created by <code>termExtraction</code> 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 <code>sep = ";"</code> .
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 <code>remove.terms = NULL</code> .
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 <code>synonyms = NULL</code> .

Value

an object of class `data.frame`

Examples

```
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)
```

lifeCycle

Life Cycle Analysis with Logistic Growth Model

Description

Estimates logistic growth model for annual (non-cumulative) publications following Meyer et al. (1999) methodology

Usage

```
lifeCycle(data, forecast_years = 5, plot = TRUE, verbose = FALSE)
```

Arguments

data	Data frame with columns: year (PY) and number of publications (n)
forecast_years	Number of years to forecast beyond saturation
plot	Logical, if TRUE produces plots
verbose	Logical, if TRUE prints detailed output

Value

List containing parameters, forecasts and metrics

loadBookDataset	<i>Load the Biblioshiny Book Dataset</i>
-----------------	--

Description

Downloads and caches the management collection dataset used in the book "Science Mapping Analysis - A primer with Biblioshiny" by Massimo Aria and Corrado Cuccurullo (McGraw-Hill, 2026). The dataset is downloaded from GitHub on first use and cached locally in `~/ .bibliometrix/` for subsequent calls.

Usage

```
loadBookDataset(force_download = FALSE)
```

Arguments

force_download	Logical. If TRUE, re-downloads the dataset even if a cached copy exists locally. Default is FALSE.
----------------	--

Details

The dataset contains a comprehensive collection of original research articles, written in English, that employ bibliometric or scientometric methods within management and business fields, as indexed by Web of Science, over the period 1985–2025.

Value

A bibliometrix data frame containing the book dataset.

References

Aria, M., & Cuccurullo, C. (2026). *Science Mapping Analysis - A primer with Biblioshiny*. McGraw-Hill. ISBN: 978-88-386-2297-7.

Examples

```
## Not run:  
M <- loadBookDataset()  
  
## End(Not run)
```

localCitations	<i>Author local citations</i>
----------------	-------------------------------

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, ]
```

 logo

Bibliometrix logo.

Description

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

Format

A matrix with 927 rows and 800 columns.

 lotka

Lotka's law coefficient estimation

Description

It estimates Lotka's law coefficients for scientific productivity and tests the goodness of fit.

Usage

```
lotka(M)
```

Arguments

M is an object of the class 'bibliometrixDB'.

Details

Lotka's Law, first formulated by Alfred J. Lotka in 1926, describes the frequency distribution of scientific productivity among authors. The law states that the number of authors producing n publications is approximately C/n^β , where C is a constant and β is the productivity exponent.

In the original formulation, Lotka proposed that $\beta = 2$, meaning that the number of authors who publish n papers is approximately $1/n^2$ of those who publish one paper. The function estimates both the empirical β via regression and tests the fit of the theoretical distribution ($\beta = 2$) using a

Kolmogorov-Smirnov test.

Reference:

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of the Washington Academy of Sciences*, 16(12), 317-323.

Value

The function `lotka` returns a list containing the following objects:

AuthorProd	Authors' Productivity frequency table
g	Lotka's law plot in ggplot2 format (with logo)
g_shiny	Lotka's law plot for biblioshiny (without logo)
stat	list of statistical results (Beta, C, R2, KS tests)
Beta	Beta coefficient (estimated)
C	Constant coefficient
R2	Goodness of Fit (R-squared)
fitted	Fitted Values
p.value	p-value of KS test (theoretical Beta=2)

See Also

[biblioAnalysis](#) function for bibliometric analysis

[summary](#) method for class 'bibliometrix'

Examples

```
data(management, package = "bibliometrixData")
L <- lotka(management)
L
```

ltwa

Index of ltwa.

Description

Data frame containing a normalized index of words used in journal names and their ISO4 abbreviations.

Format

A data frame with 56463 rows and 3 variables:

WORD word from journal names

ABBREVIATION ISO4 abbreviation

LANGUAGES Language of the journal name

mergeDbSources	<i>Merge bibliographic data frames from supported bibliographic DBs</i>
----------------	---

Description

Merge bibliographic data frames from different databases (WoS, SCOPUS, Lens, Openalex, etc-) into a single one.

Usage

```
mergeDbSources(..., remove.duplicated = TRUE, verbose = TRUE)
```

Arguments

`...` are the bibliographic data frames to merge.
`remove.duplicated` is logical. If TRUE duplicated documents will be deleted from the bibliographic collection.
`verbose` is logical. If TRUE, information on duplicate documents is printed on the screen.

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

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

`mergeKeywords`*Merge DE and ID Fields into a Unified Keywords Column*

Description

This function creates a new column 'KW_Merged' by combining the contents of the 'DE' (author keywords) and 'ID' (keywords plus) fields in a bibliographic dataframe. Duplicate keywords within each record are removed, and leading/trailing spaces are trimmed. The merged keywords are separated by a semicolon (;).

Usage

```
mergeKeywords(M, force = FALSE)
```

Arguments

M	A dataframe containing at least the 'DE' and/or 'ID' columns, typically generated by 'convert2df()' from the 'bibliometrix' package.
force	Logical. If 'TRUE', an existing 'KW_Merged' column will be overwritten. Default is 'FALSE'.

Details

If the 'KW_Merged' column already exists, it will not be overwritten unless 'force = TRUE' is specified.

Value

A dataframe with an added (or updated) 'KW_Merged' column containing deduplicated and cleaned keyword strings.

Examples

```
## Not run:  
data(management, package = "bibliometrix")  
M <- mergeKeywords(management)  
head(M$KW_Merged)  
  
## End(Not run)
```

metaTagExtraction *Meta-Field Tag Extraction*

Description

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

Usage

```
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]
```

missingData

Completeness of bibliographic metadata

Description

It calculates the percentage of missing data in the metadata of a bibliographic data frame.

Usage

```
missingData(M)
```

Arguments

M is a bibliographic data frame obtained by [convert2df](#) function.

Details

Each metadata is assigned a status c("Excellent," "Good," "Acceptable", "Poor", "Critical," "Completely missing") depending on the percentage of missing data. In particular, the column **status** classifies the percentage of missing value in 5 categories: "Excellent" (0 "Poor" (from 20.01

The results of the function allow us to understand which analyses can be performed with bibliometrix and which cannot based on the completeness (or status) of different metadata.

Value

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

allTags is a data frame including results for all original metadata tags from the collection
 mandatoryTags is a data frame that included only the tags needed for analysis with bibliometrix and biblioshiny.

Examples

```
data(scientometrics, package = "bibliometrixData")
res <- missingData(scientometrics)
print(res$mandatoryTags)
```

net2Pajek	<i>Save a network graph object as Pajek files</i>
-----------	---

Description

The function `net2Pajek` save a bibliographic network previously created by `networkPlot` as pajek files.

Usage

```
net2Pajek(net, filename = "my_pajek_network", path = NULL)
```

Arguments

`net` is a network graph object returned by the function `networkPlot`.
`filename` is a character. It indicates the filename for Pajek export files.
`path` is a character. It indicates the path where the files will be saved. When `path="NULL"`, the files will be saved in the current folder. Default is `NULL`.

Value

The function returns no object but will save three Pajek files in the folder given in the "path" argument with the name "filename.clu," "filename.vec," and "filename.net."

See Also

[net2VOSviewer](#) to export and plot the network with VOSviewer software.

Examples

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

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

```
net <- networkPlot(NetMatrix, n = 30, type = "auto", Title = "Co-occurrence Network", labelsizesize = 1)
net2Pajek(net, filename = "pajekfiles", path = NULL)

## End(Not run)
```

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

```
# 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)
```

networkPlot

Plotting Bibliographic networks

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 = "walktrap",
  community.repulsion = 0.5,
  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,
  seed = 123,
  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: <table> <tr> <td>type="auto"</td> <td>Automatic layout selection</td> </tr> <tr> <td>type="circle"</td> <td>Circle layout</td> </tr> <tr> <td>type="sphere"</td> <td>Sphere layout</td> </tr> <tr> <td>type="mds"</td> <td>Multidimensional Scaling layout</td> </tr> <tr> <td>type="fruchterman"</td> <td>Fruchterman-Reingold layout</td> </tr> <tr> <td>type="kamada"</td> <td>Kamada-Kawai layout</td> </tr> </table>	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
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.												
labelsize	is an integer. It indicates the label size in the plot. Default is labelsize=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 number between 0 and 1. Controls the separation between communities with adaptive scaling. The algorithm automatically adapts to network size and structure. Recommended values: <table> <tr> <td>0.0</td> <td>No repulsion - communities may overlap</td> </tr> <tr> <td>0.2-0.4</td> <td>Light separation - suitable for dense networks</td> </tr> <tr> <td>0.5-0.7</td> <td>Moderate separation - general use (RECOMMENDED)</td> </tr> <tr> <td>0.8-1.0</td> <td>Strong separation - for many small communities</td> </tr> </table> <p>Default is community.repulsion = 0.5.</p>	0.0	No repulsion - communities may overlap	0.2-0.4	Light separation - suitable for dense networks	0.5-0.7	Moderate separation - general use (RECOMMENDED)	0.8-1.0	Strong separation - for many small communities				
0.0	No repulsion - communities may overlap												
0.2-0.4	Light separation - suitable for dense networks												
0.5-0.7	Moderate separation - general use (RECOMMENDED)												
0.8-1.0	Strong separation - for many small communities												
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.
seed	is an integer. It indicates the random seed for clustering reproducibility. Default is seed = 123.
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.

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

```
# EXAMPLE Keywordd 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)
```

networkStat	<i>Calculating network summary statistics</i>
-------------	---

Description

networkStat calculates main network statistics.

Usage

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

Arguments

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

Details

The function [networkStat](#) can calculate the main network statistics from a bibliographic network previously created by [biblioNetwork](#).

Value

It is a list containing the following elements:

graph	a network object of the class <code>igraph</code>
network	a <code>communities</code> 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 is 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 `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

```
## Not run:
data(management, package = "bibliometrixData")
NCS <- normalizeCitationScore(management, 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".

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

Leydesdorff, L. (2008). On the normalization and visualization of author Cocitation data: Salton's cosine versus the Jaccard index. *Journal of the American Society for Information Science and Technology*, 59(1), 77– 85.

Van Eck, N.J., Waltman, L., Van den Berg, J., & Kaymak, U. (2006). Visualizing the computational intelligence field. *IEEE Computational Intelligence Magazine*, 1(4), 6– 10.

Van Eck, N.J., & Waltman, L. (2007). Bibliometric mapping of the computational intelligence field. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(5), 625– 645

. Van Eck, N. J., & Waltman, L. (2009). How to normalize cooccurrence data? An analysis of some well-known similarity measures. *Journal of the American society for information science and technology*, 60(8), 1635-1651.

Zitt, M., Bassecouard, E., & Okubo, Y. (2000). Shadows of the past in international cooperation: Collaboration profiles of the top five producers of science. *Scientometrics*, 47(3), 627– 657.

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 = ";")
```

```
)
S <- normalizeSimilarity(NetMatrix, type = "association")
```

normalize_citations *Normalize and match bibliographic citations*

Description

This function performs advanced normalization and fuzzy matching of bibliographic citations to identify and group citations that refer to the same work but are formatted differently. It uses a multi-phase approach combining string normalization, blocking strategies, hierarchical clustering, and post-processing to achieve both speed and accuracy on large citation datasets.

Usage

```
normalize_citations(CR_vector, threshold = 0.9, method = "jw", min_chars = 20)
```

Arguments

CR_vector	Character vector containing bibliographic citations to be normalized and matched.
threshold	Numeric value between 0 and 1 indicating the similarity threshold for matching citations. Higher values (e.g., 0.90-0.95) produce more conservative matching, while lower values (e.g., 0.75-0.80) produce more aggressive matching. Default is 0.85, which provides a good balance between precision and recall.
method	String distance method to use for fuzzy matching. Options include: <ul style="list-style-type: none"> • "jw" (default): Jaro-Winkler distance, optimized for bibliographic strings • "lv": Levenshtein distance • Other methods supported by stringdistmatrix
min_chars	Minimum characters for valid citations (default: 20)

Details

The function implements a five-phase matching algorithm:

Phase 1: Normalization and Feature Extraction

- Converts text to uppercase
- Removes issue numbers and page numbers (which often contain typos)
- Removes punctuation and normalizes whitespace
- Expands common journal abbreviations (e.g., "J. CLEAN. PROD." -> "JOURNAL OF CLEANER PRODUCTION")
- Extracts key features: first author, year, journal, volume, pages

Phase 1.5: Journal Normalization The function uses the LTWA (List of Title Word Abbreviations) database from ISO 4 standards to normalize journal names. This ensures that abbreviated forms (e.g., "J. Clean. Prod.") and full forms (e.g., "Journal of Cleaner Production") are recognized as the same journal and matched together.

The LTWA database is included in the bibliometrix package. If not found, the function attempts to download it from ISSN.org. Journal normalization can be disabled by ensuring the LTWA database is not available.

Phase 2: Blocking Citations are grouped into blocks by first author and year. This dramatically reduces computational complexity from $O(n^2)$ to approximately $O(k*m^2)$, where k is the number of blocks and m is the average block size.

Phase 3: Within-Block Matching Within each block, citations are compared using string distance metrics and hierarchical clustering. For blocks larger than 500 citations, exact matching on normalized strings is used instead to maintain performance.

Phase 4: Canonical Representative Selection For each cluster, the most complete citation (prioritizing those with volume and page information) is selected as the canonical representative.

Phase 5: Post-Processing Citations sharing the same first author, year, journal, and volume are merged into a single cluster, even if they weren't matched in Phase 3. This catches cases where minor title variations prevented matching.

Value

A data frame with the following columns:

- CR_original: Original citation string
- CR_canonical: Canonical (representative) citation for the cluster
- cluster_id: Unique identifier for each citation cluster
- n_cluster: Number of citations in the cluster
- first_author: First author surname
- year: Publication year
- journal_iso4: Journal name normalized to ISO4 abbreviated form
- journal_original: Original journal name as extracted from citation
- volume: Volume number
- doi: Digital Object Identifier (when available)
- blocking_key: Internal key used for blocking (author_year_journal)

References

Aria, M. & Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4), 959-975.

See Also

[applyReferenceMatching](#) for direct application to bibliometrix data frames

Examples

```
## Not run:
# Load bibliometrix data
data(scientometrics, package = "bibliometrixData")

# Extract and normalize citations
CR_vector <- unlist(strsplit(scientometrics$CR, ";"))
CR_vector <- trimws(CR_vector)

# Perform normalization with default threshold
matched <- normalize_citations(CR_vector)

# View matching statistics
table(matched$n_cluster)

# Find all variants of a specific citation
subset(matched, cluster_id == matched$cluster_id[1])

# Use more conservative matching
matched_conservative <- normalize_citations(CR_vector, threshold = 0.90)

## End(Not run)
```

plot.bibliodendrogram *Plotting dendrogram resulting from Conceptual Structure Analysis*

Description

plot method for class 'bibliodendrogram'

Usage

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

Arguments

x is the object for which plots are desired.
... is a generic param for plot functions.

Value

The function plot draws a dendrogram.

plot.bibliometrix *Plotting bibliometric analysis results*

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 plot a Thematic Evolution Analysis performed using the [thematicEvolution](#) function.

Usage

```
plotThematicEvolution(Nodes, Edges, measure = "lineage_strength", 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 is deprecated and has no effect. The weight of the flows is always calculated using lineage_strength.
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.
[thematicEvolution](#) function to perform a thematic evolution analysis.
[networkPlot](#) to plot a bibliographic network.

Examples

```
## Not run:  
data(management, package = "bibliometrixData")  
years=c(2004,2008,2015)  
  
nexus <- thematicEvolution(management,field="DE",years=years,n=100,minFreq=2)  
  
plotThematicEvolution(nexus$Nodes,nexus$Edges)  
  
## End(Not run)
```

```
print_author_works_summary
```

Print Summary of Author Works Search

Description

Prints a summary of the search results from `find_author_works()`

Usage

```
print_author_works_summary(works_df)
```

Arguments

`works_df` Data.frame. Result from `find_author_works()`

```
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 or ISI Export file into a dataframe

Examples

```
# WoS or SCOPUS Export files can be read using \link{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

<code>id</code>	is a vector of characters containing the author's SCOPUS IDs. SCOPUS IDs can be obtained using the function idByAuthor .
<code>api_key</code>	is a character. It contains the Elsevier API key. Information about how to obtain an API Key Elsevier API website
<code>remove.duplicated</code>	is logical. If TRUE duplicated documents will be deleted from the bibliographic collection.
<code>country</code>	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
```

rpys

*Reference Publication Year Spectroscopy***Description**

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

Usage

```
rpys(M, sep = ";", timespan = NULL, median.window = "centered", 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).
median.window	is a character string that can be "centered" or "backward". It indicates the type of median to be used. "centered" is the default value and it uses the centered 5-year median (t-2 to t+2) as proposed by Marx et al. (2014). "backward" uses the backward 5-year median (t-4 to t) as proposed by Aria and Cuccurullo (2017).
graph	is a logical. If TRUE the function plot the spectroscopy otherwise the plot is created but not drawn down.

Details**References:**

Marx, W., Bornmann, L., Barth, A., & Leydesdorff, L. (2014). Detecting the historical roots of research fields by reference publication year spectroscopy (RPYS). *Journal of the Association for Information Science and Technology*, 65(4), 751-764.

Thor A., Bornmann L., Mark W. & Mutz R.(2018). Identifying single influential publications in a research field: new analysis opportunities of the CRExplorer. *Scientometrics*, 116:591–608 <https://doi.org/10.1007/s11192-018-2733-7>

Value

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

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

```
## Not run:
data(management, package = "bibliometrixData")
res <- rplys(management, sep = ";", graph = TRUE)

## End(Not run)
```

sourceGrowth

Number of documents published annually per Top Sources

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 convert2df . 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.

Value

an object of class data.frame

Examples

```
data(scientometrics, package = "bibliometrixData")
topS0 <- sourceGrowth(scientometrics, top = 1, cdf = TRUE)
topS0

# Plotting results
## Not run:
install.packages("reshape2")
library(reshape2)
library(ggplot2)
DF <- melt(topS0, id = "Year")
ggplot(DF, aes(Year, value, group = variable, color = variable)) +
  geom_line()

## End(Not run)
```

splitCommunities	<i>Splitting Network communities</i>
------------------	--------------------------------------

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`

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 Keywordd 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.

summary.bibliometrix *Summarizing bibliometric analysis results*

Description

summary method for class 'bibliometrix'

Usage

```
## 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

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

summary.bibliometrix_netstat

Summarizing network analysis results

Description

summary method for class 'bibliometrix_netstat'

Usage

```
## 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")

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

tableTag	<i>Tabulate elements from a Tag Field column</i>
----------	--

Description

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

Usage

```
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 <code>sep = ";"</code> .
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 <code>remove.terms = NULL</code> .
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 <code>synonyms = NULL</code> .

Details

tableTag is an internal routine of main function [biblioAnalysis](#).

Value

an object of class table

Examples

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

termExtraction

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

```
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 <code>convert2df</code> . 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 : <ul style="list-style-type: none"> "TI" Manuscript title "AB" Manuscript abstract "ID" Manuscript keywords plus "DE" Manuscript author's keywords <p>The default is <code>Field = "TI"</code>.</p>
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 <code>ngrams=1</code> .
stemming	is logical. If TRUE the Porter Stemming algorithm is applied to all extracted terms. The default is <code>stemming = FALSE</code> .
language	is a character. It is the language of textual contents ("english", "german", "italian", "french", "spanish"). The default is <code>language="english"</code> .
remove.numbers	is logical. If TRUE all numbers are deleted from the documents before term extraction. The default is <code>remove.numbers = TRUE</code> .
remove.terms	is a character vector. It contains a list of additional terms to delete from the corpus after term extraction. The default is <code>remove.terms = NULL</code> .
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 <code>keep.terms = NULL</code> .
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 <code>synonyms = NULL</code> .
verbose	is logical. If TRUE the function prints the most frequent terms extracted from documents. The default is <code>verbose=TRUE</code> .

Value

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

See Also

[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
scientometrics <- 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

```
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 = "louvain",
  seed = 1234,
  assign.evolution.colors = list(assign = TRUE, alpha = 0.5)
)
```

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","KW_Merged","TI","AB"). Default value is field="ID".
years	is a numeric vector of one 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").
seed	is numerical. It indicates the seed for random number generator to obtain always the same results. Default value is seed = 1234.
assign.evolution.colors	is a list. If assignEvolutionColors = list(assign = TRUE), colors are assigned to lineages based on the highest weighted inclusion value. If a list is provided, it must contain the arguments assignEvolutionColors = list(assign = c(TRUE, FALSE), alpha=0.5). Default is assign.evolution.colors = list(assign=TRUE, alpha = 0.5). If assign = FALSE, measure argument is ignored.

Details

[thematicEvolution](#) starts from two or more thematic maps created by [thematicMap](#) function.

Reference:

Cobo, M. J., Lopez-Herrera, A. G., Herrera-Viedma, E., & Herrera, F. (2011). An approach for detecting, quantifying, and visualizing the evolution of a research field: A practical application to the fuzzy sets theory field. *Journal of Informetrics*, 5(1), 146-166.

Value

a list containing:

nets	The thematic nexus graph for each comparison
incMatrix	Some useful statistics about the thematic nexus

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

```
## Not run:
data(management, package = "bibliometrixData")
years=c(2004,2008,2015)
```

```
nexus <- thematicEvolution(management, field="DE", 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

```
thematicMap(
  M,
  field = "ID",
  n = 250,
  minfreq = 5,
  ngrams = 1,
  stemming = FALSE,
  size = 0.5,
  n.labels = 1,
  community.repulsion = 0.5,
  repel = TRUE,
  remove.terms = NULL,
  synonyms = NULL,
  cluster = "louvain",
  subgraphs = FALSE,
  alpha = 0.5,
  seed = 1234
)
```

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", "KW_Merged", "TI", "AB"). biblioNetwork or cocMatrix .
n	is an integer. It indicates the number of terms to include in the analysis.
minfreq	is a 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 = 0.5.
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").
subgraphs	is a logical. If TRUE cluster subgraphs are returned.
alpha	is a real. it indicates the balancing between word frequency and pagerank centrality to identify cluster labels. Default is alpha = 0.5.
seed	is an integer. It indicates the seed for random number generation. Default is seed = 1234.

Details

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

Reference:

Cobo, M. J., Lopez-Herrera, A. G., Herrera-Viedma, E., & Herrera, F. (2011). An approach for detecting, quantifying, and visualizing the evolution of a research field: A practical application to the fuzzy sets theory field. *Journal of Informetrics*, 5(1), 146-166.

Value

a list containing:

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(management, package = "bibliometrixData")
res <- thematicMap(management, field = "ID", n = 250, minfreq = 5, size = 0.5, repel = TRUE)
plot(res$map)
plot(res$net$graph)

## 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("DE", "AU", "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 <code>fields = c("AU", "DE", "SO")</code> .
n	is a integer vector. It indicates how many items to plot, for each of the three fields. Default is <code>n = c(20, 20, 20)</code>

Value

a sankeyPlot

Examples

```
# data(scientometrics, package = "bibliometrixData")

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

timeslice	<i>Bibliographic data frame time slice</i>
-----------	--

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

```
data(scientometrics, package = "bibliometrixData")  
list_df <- timeslice(scientometrics, breaks = c(1995, 2005))  
names(list_df)
```

trim	<i>Deleting leading and ending white spaces</i>
------	---

Description

Deleting leading and ending white spaces from a character object.

Usage

```
trim(x)
```

Arguments

x is a character object.

Details

tableTag is an internal routine of bibliometrics package.

Value

an object of class character

Examples

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

trim.leading	<i>Deleting leading white spaces</i>
--------------	--------------------------------------

Description

Deleting leading white spaces from a character object.

Usage

```
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

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

trimES	<i>Deleting extra white spaces</i>
--------	------------------------------------

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

```
char <- 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-occurrences**
 - [bibliometrix-package](#), 3
 - * **co-word analysis**
 - [bibliometrix-package](#), 3
 - * **collaboration**
 - [bibliometrix-package](#), 3
 - * **network**
 - [bibliometrix-package](#), 3
- [applyCitationMatching](#), 7
- [applyReferenceMatching](#), 7, 7, 66
- [assignEvolutionColors](#), 10
- [authorBio](#), 12
- [authorProdOverTime](#), 13
-
- [biblioAnalysis](#), 14, 14, 18, 21, 22, 24, 29, 31, 35–37, 41, 44, 46, 49, 51, 52, 54, 60, 62, 68, 74–77, 79, 81, 87
- [bibliometrix \(bibliometrix-package\)](#), 3
- [bibliometrix-package](#), 3
- [biblioNetwork](#), 16, 17, 24, 25, 29, 33, 42, 57, 59–64, 74, 75, 84, 86
- [biblioshiny](#), 18
- [bibtag](#), 19
- [bradford](#), 20
-
- [citations](#), 9, 21, 49
-
- [cocMatrix](#), 18, 22, 29, 33, 44, 60, 62–64, 75, 83, 84, 86
- [collabByRegionPlot](#), 24
- [conceptualStructure](#), 27
- [convert2df](#), 7, 8, 14–16, 18, 21, 23, 24, 28, 29, 36, 37, 40–42, 45–47, 49, 52, 54, 55, 62, 70, 73, 74, 79–82, 86, 87
- [countries](#), 31
- [couplingMap](#), 32, 32
- [customTheme](#), 34
-
- [dominance](#), 34
- [duplicatedMatching](#), 35
-
- [fieldByYear](#), 36
- [findAuthorWorks](#), 38
-
- [get_authors_summary](#), 39
-
- [Hindex](#), 40
- [histNetwork](#), 41, 43, 44
- [histPlot](#), 43, 43
-
- [idByAuthor](#), 44, 71, 72
-
- [keywordAssoc](#), 45
- [KeywordGrowth](#), 46
-
- [lifeCycle](#), 47
- [loadBookDataset](#), 48
- [localCitations](#), 9, 49
- [logo](#), 50
- [lotka](#), 50
- [ltwa](#), 51
-
- [mergeDbSources](#), 52
- [mergeKeywords](#), 53
- [metaTagExtraction](#), 54
- [missingData](#), 55
-
- [net2Pajek](#), 56, 56

net2VOSviewer, [56](#), [57](#), [60](#), [75](#)
networkPlot, [33](#), [56](#), [57](#), [58](#), [60](#), [69](#), [75](#), [83](#), [86](#)
networkStat, [61](#), [61](#)
normalize_citations, [7–9](#), [65](#)
normalizeCitationScore, [62](#), [62](#)
normalizeSimilarity, [63](#)

plot, [15](#), [22](#), [36](#), [41](#), [42](#), [46](#), [49](#), [52](#), [77](#), [87](#)
plot.bibliodendrogram, [67](#)
plot.bibliometrix, [68](#)
plotThematicEvolution, [11](#), [69](#)
print_author_works_summary, [70](#)

readFiles, [70](#)
retrievalByAuthorID, [45](#), [71](#)
rphys, [73](#)

sourceGrowth, [74](#)
splitCommunities, [75](#), [75](#)
stopwords, [76](#)
stringdistmatrix, [7](#), [8](#), [65](#)
summary, [14](#), [15](#), [21](#), [22](#), [35–37](#), [41](#), [42](#), [46](#), [49](#),
[51](#), [52](#), [68](#), [87](#)
summary.bibliometrix, [76](#)
summary.bibliometrix_netstat, [77](#)

tableTag, [78](#)
termExtraction, [29](#), [47](#), [79](#)
thematicEvolution, [10](#), [11](#), [69](#), [82](#), [83](#)
thematicMap, [69](#), [83](#), [84](#)
threeFieldsPlot, [86](#)
timeslice, [87](#)
trim, [88](#)
trim.leading, [88](#)
trimES, [89](#)