

# Package ‘bibliometrix’

January 11, 2019

**Type** Package

**Title** An R-Tool for Comprehensive Science Mapping Analysis

**Version** 2.1.0

**Date** 2019-01-06

**Description** Tool for quantitative research in scientometrics and bibliometrics. It provides various routines for importing bibliographic data from SCOPUS (<<http://scopus.com>>), Clarivate Analytics Web of Science (<<http://www.webofknowledge.com/>>), Cochrane Library (<<http://www.cochranelibrary.com/>>) and PubMed (<<https://www.ncbi.nlm.nih.gov/pubmed/>>) databases, performing bibliometric analysis and building networks for co-citation, coupling, scientific collaboration and co-word analysis.

**License** GPL-3

**URL** <http://www.bibliometrix.org>

**LazyData** FALSE

**Encoding** UTF-8

**Depends** R (>= 3.3.0)

**Imports** stats, dplyr, DT, factoextra, FactoMineR, ggraph, ggplot2, ggpepel, igraph, Matrix, networkD3, RColorBrewer, RISmed, rscopus, shiny, shinyessloaders, shinythemes, SnowballC, stringdist, stringr

**Suggests** knitr, rmarkdown,

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Author** Massimo Aria [cre, aut],  
Corrado Cuccurullo [aut]

**Maintainer** Massimo Aria <[aria@unina.it](mailto:aria@unina.it)>

**VignetteBuilder** knitr,

**Repository** CRAN

**Date/Publication** 2019-01-11 22:10:03 UTC

**R topics documented:**

bibliometrix-package	3
authorProdOverTime	6
bib2df	7
biblio	8
biblioAnalysis	8
biblioNetwork	10
biblioshiny	12
biblio_df	12
bibtag	13
bradford	13
citations	14
cochrane2df	15
cocMatrix	16
conceptualStructure	18
convert2df	19
countries	21
dominance	22
duplicatedMatching	23
garfield	24
Hindex	25
histNetwork	26
histPlot	27
idByAuthor	28
isi2df	29
isibib2df	30
isiCollection	31
keywordAssoc	32
KeywordGrowth	33
localCitations	34
lotka	35
mergeDbSources	35
metaTagExtraction	37
networkPlot	38
networkStat	40
normalizeSimilarity	42
plot.bibliometrix	43
plotThematicEvolution	44
pubmed2df	45
readFiles	46
retrievalByAuthorID	46
rpys	48
scientometrics	49
scientometrics_text	50
scopus2df	51
scopusCollection	52
sourceGrowth	53

stopwords . . . . .	54
summary.bibliometrix . . . . .	54
summary.bibliometrix_netstat . . . . .	55
tableTag . . . . .	56
termExtraction . . . . .	57
thematicEvolution . . . . .	59
thematicMap . . . . .	60
timeslice . . . . .	61
trim . . . . .	62
trim.leading . . . . .	62

<b>Index</b>	<b>64</b>
--------------	-----------

---

bibliometrix-package    *An R-Tool for Comprehensive Science Mapping Analysis*

---

## Description

Tool for quantitative research in scientometrics and bibliometrics. It provides various routines for importing bibliographic data from SCOPUS (<<http://scopus.com>>), Clarivate Analytics Web of Science (<<http://www.webofknowledge.com/>>), Cochrane Library (<<http://www.cochranelibrary.com/>>) and PubMed (<<https://www.ncbi.nlm.nih.gov/pubmed/>>) 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 read by R using the function `*readFiles*`: (An example from bibliometrix vignettes)

```
D <- readFiles("http://www.bibliometrix.org/datasets/savedrecc.bib")
```

D is a large character vector. `*readFiles*` argument contains the name of files downloaded from SCOPUS, Clarivate Analytics WOS, or Cochrane CDSR website.

The function `*readFiles*` combines all the text files onto a single large character vector. Furthermore, the format is converted into UTF-8.

```
es. D <- readFiles("file1.txt", "file2.txt", ...)
```

The object D can be converted in a data frame using the function `*convert2df*`:

```
M <- convert2df(D, dbsource = "isi", 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="CA")
CS <- conceptualStructure(M,field="ID", method="CA", minDegree=4, k.max=8, stemming=FALSE, labels=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, n = 20, sep = ". ")
# Plot a historical co-citation network
net <- histPlot(histResults, size = FALSE,label=TRUE, arrowsize = 0.5)
```

**Author(s)**

Massimo Aria [cre, aut], Corrado Cuccurullo [aut]  
 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 ([http://www.bibliometrix.org/documents/2015Carme\\_cuccurullo](http://www.bibliometrix.org/documents/2015Carme_cuccurullo)).

Sarto, F., Cuccurullo, C., & Aria, M. (2014). Exploring healthcare governance literature: systematic review and paths for future research. *\*Mecosan\** ([http://www.francoangeli.it/Riviste/Scheda\\_Rivista.aspx?IDarticolo=52780](http://www.francoangeli.it/Riviste/Scheda_Rivista.aspx?IDarticolo=52780)).

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>).

---

authorProdOverTime      *Top-Authors' Productivity over the Time*

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

The function authorProdOverTime returns a list containing two objects:

dfAU	is a data frame
graph	a ggplot object

**See Also**

[biblioAnalysis](#) function for bibliometric analysis

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

**Examples**

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

---

bib2df

*Convert a bibtex file into a data frame*

---

**Description**

It converts a bibtex file and create a data frame from it, with cases corresponding to articles and variables to Field Tag in the original file.

**Usage**

```
bib2df(D, dbsource = "isi")
```

**Arguments**

D is a character array containing bibliographic data in bibtex format.

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

**Value**

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

**See Also**

[isi2df](#) for converting ISI Export file (in plain text format)

Other converting functions: [cochrane2df](#), [convert2df](#), [isi2df](#), [isibib2df](#), [pubmed2df](#), [scopus2df](#)

### Examples

```
# A ISI Export file can be read using \link{readLines} function:  
  
# largechar <- readFiles('filename1.bib', 'filename2.bib2,...')  
  
# filename.bib is a bibtex file in plain text format.  
  
# largechar <- readFiles('http://www.bibliometrix.org/datasets/ranking.bib')  
  
# ranking <- bib2df(largechar)
```

---

biblio	<i>Dataset of "Bibliometrics" scientific documents.</i>
--------	---

---

### Description

The set of manuscripts which the title containing the word "bibliometrics" and published in a journal indexed by ISI WoK database.  
Period: 2006 - 2015  
Database: [ISI Web of Knowledge](#)

### Format

A large character with 9014 rows.  
Data has been imported by an ISI Export file in bibtex format using the function [readLines](#).

### Source

<http://www.webofknowledge.com>

---

biblioAnalysis	<i>Bibliometric Analysis</i>
----------------	------------------------------

---

### Description

It performs a bibliometric analysis of a dataset imported from SCOPUS and Thomson Reuters' ISI Web of Knowledge databases.

### Usage

```
biblioAnalysis(M, sep = ";")
```



**Arguments**

<code>M</code>	is a bibliographic data frame obtained by the converting function <code>convert2df</code> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge file.
<code>sep</code>	is the field separator character. This character separates strings in each column of the data frame. The default is <code>sep = ";"</code> .

**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	first author of each manuscript
nAUpperPaper	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	Intracountry (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 Thomson Reuters

**See Also**

`convert2df` to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.

`summary` to obtain a summary of the results.

`plot` to draw some useful plots of the results.

**Examples**

```
data(scientometrics)

results <- biblioAnalysis(scientometrics)
```

```
summary(results, k = 10, pause = FALSE)
```

---

 biblioNetwork

*Creating Bibliographic networks*


---

## Description

biblioNetwork creates different bibliographic networks from a bibliographic data frame.

## Usage

```
biblioNetwork(M, analysis = "coupling", network = "authors",
  sep = ";", shortlabel = TRUE)
```

## Arguments

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge file.
analysis	is a character object. It indicates the type of analysis have to be performed. analysis argument can be "collaboration", "coupling", "co-occurrences" or "co-citation". Default is analysis = "coupling".
network	is a character object. It indicates the network typology. The network argument can be "authors", "references", "sources", "countries", "keywords", "author_keywords", "titles", or "abstracts". Default is network = "authors".
sep	is the field separator character. This character separates strings in each column of the data frame. The default is sep = ";".
shortlabel	is logical. IF TRUE, reference labels are stored in a short format. Default is shortlabel=TRUE.

## Details

The function [biblioNetwork](#) can create a collection of bibliographic networks following the approach proposed by Batagely and Cerinsek (2013).

Typical networks output of biblioNetwork are:

```
##### Collaboration Networks #####
```

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

## 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)

# 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)

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

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

---

biblioshiny	<i>Shiny UI for bibliometrix package</i>
-------------	--

---

**Description**

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

**Usage**

```
biblioshiny()
```

**Examples**

```
#biblioshiny()
```

---

biblio_df	<i>Dataset of "Bibliometrics" manuscripts.</i>
-----------	--

---

**Description**

The set of manuscripts which the title containing the word "bibliometrics" and published in a journal indexed by ISI WoK database.

Period: 2006 - 2015

Database: **ISI Web of Knowledge**

**Format**

#' A data frame with 99 rows (manuscripts) and 16 variables (ISI tag field):

**AU** Authors

**TI** Document Title

**SO** Publication Name (or Source)

**JI** ISO Source Abbreviation

**DT** Document Type

**DE** Author Keywords

**ID** Keywords associated by ISI or SCOPUS 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

### Source

<http://www.webofknowledge.com>

---

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

---

### Description

Data frame containing a list of tags and corresponding WoS and SCOPUS bibtex field.

### Format

A data frame with 39 rows and 4 variables:

**TAG** country names  
**SCOPUS** continent names  
**ISI** country centroid longitude  
**GENERIC** country centroid latitude

---

bradford	<i>Bradford's law</i>
----------	-----------------------

---

### Description

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

### Usage

bradford(M)

### Arguments

M is a bibliographic dataframe.

**Details**

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

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

**Value**

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

<code>table</code>	a dataframe with the source distribution partitioned in the three zones
<code>graph</code>	the source distribution plot in ggplot2 format

**See Also**

[biblioAnalysis](#) function for bibliometric analysis

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

**Examples**

```
# D <- readFiles('http://www.bibliometrix.org/datasets/bibliometrics_articles.txt')
# M <- convert2df(file = D, dbsource = "isi", format = "bibtex")
# BR <- bradford(M)
```

---

citations

*Citation frequency distribution*

---

**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 Thomson Reuters' ISI Web of Knowledge 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 ISI 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)

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]
```

**Description**

It converts a Cochrane Database Export file and create a data frame from it, with cases corresponding to articles and variables to Field Tag in the original file.

**Usage**

```
cochrane2df(D)
```

**Arguments**

D is a character array containing data read from a ISI Export file (in plain text format).

**Value**

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

**See Also**

[scopus2df](#) for converting SCOPUS Export file (in bibtex format)

Other converting functions: [bib2df](#), [convert2df](#), [isi2df](#), [isibib2df](#), [pubmed2df](#), [scopus2df](#)

**Examples**

```
# A group of Cochrane Database Export files can be read using \code{\link{readFiles}} function:
# largechar <- readFiles('filename1.txt','filename2.txt','filename3.txt')
# filename.txt is a Cochrane Database Export file in plain text format.
# scientometrics_text <- readFiles('http://www.bibliometrix.org/datasets/cochrane.txt')
# scient_df <- cochrane2df(cochrane_text)
```

---

cocMatrix

*Co-occurrence matrix*

---

**Description**

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

**Usage**

```
cocMatrix(M, Field = "AU", type = "sparse", sep = ";",
  binary = 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.

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 this tags:



AU	Authors
SO	Publication Name (or Source)
JI	ISO Source Abbreviation
DE	Author Keywords
ID	Keywords associated by ISI or SCOPUS database
CR	Cited References

for a complete list of filed tags see: [Field Tags used in bibliometrix](#)

type	indicates the output format of co-occurrences:
type = "matrix"	produces an object of class <code>matrix</code>
type = "sparse"	produces an object of class <code>dgMatrix</code> of the package <code>Matrix</code> . "sparse" argument generates a compact
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 TRUE each cell contains a 0/1. if FALSE each cell contains the frequency.

### Details

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

### Value

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

### See Also

[convert2df](#) to import and convert an ISI or SCOPUS Export file in a data frame.  
[biblioAnalysis](#) to perform a bibliometric analysis.  
[biblioNetwork](#) to compute a bibliographic network.

### Examples

```
# EXAMPLE 1: Articles x Authors co-occurrence matrix

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

# EXAMPLE 2: Articles x Cited References co-occurrence matrix

# data(scientometrics)

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

```
# EXAMPLE 3: Articles x Cited First Authors co-occurrence matrix

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

---

conceptualStructure     *Creating and plotting conceptual structure map of a scientific field*

---

### Description

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

### Usage

```
conceptualStructure(M, field = "ID", method = "MCA",
  quali.sup = NULL, quanti.sup = NULL, minDegree = 2, k.max = 5,
  stemming = FALSE, labelsize = 10, documents = 10, graph = 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 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 this tags:
ID	Keywords Plus associated by ISI or SCOPUS database
DE	Author's keywords
ID_TM	Keywords Plus stemmed through the Porter's stemming algorithm
DE_TM	Author's Keywords stemmed through the Porter's stemming algorithm
TI	Terms extracted from titles
AB	Terms extracted from abstracts
method	is a character object. It indicates the factorial method used to create the factorial map. Use <code>method="CA"</code> for Correspondence Analysis, <code>method="MCA"</code> for Multiple Correspondence Analysis or <code>method="MDS"</code> for Metric Multidimensional Scaling. The default is <code>method="MCA"</code>
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.
k.max	is an integer. It indicates the maximum number of clusters to keep. The default value is 5. The maximum value is 8.
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 to plot in the factorial map. The default value is 10. 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

### Value

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

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

### See Also

[termExtraction](#) to extract terms from a textual field (abstract, title, author's keywords, etc.) of a bibliographic data frame.

[biblioNetwork](#) to compute a bibliographic network.

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

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

### Examples

```
# EXAMPLE Conceptual Structure using Keywords Plus

data(scientometrics)

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

---

convert2df

*Convert a Clarivate Analytics WoS, SCOPUS and COCHRANE Database Export files or RISmed PubMed/MedLine object into a data frame*

---

**Description**

It converts a SCOPUS, Clarivate Analytics WoS and COCHRANE Database export files or RISmed PubMed/MedLine object into a data frame, with cases corresponding to articles and variables to Field Tags as used in WoS.

**Usage**

```
convert2df(file, dbsource = "isi", format = "plaintext")
```

**Arguments**

file	can be: a) a character array containing data read from a Clarivate Analytics WoS Export file (in plain text or bibtex format) or SCOPUS Export file (exclusively in bibtex format); b) an object of the class pubmed (package RISmed) containing a collection obtained from a query performed with RISmed package.
dbsource	is a character indicating the bibliographic database. dbsource can be "isi", "scopus" or pubmed. Default is dbsource = "isi".
format	is a character indicating the format of the SCOPUS and Clarivate Analytics WoS export file. format can be "bibtex" or "plaintext". Default is format = "plaintext".

**Details**

Actually the function allows to convert both SCOPUS/WoS files in bibtex format and just WoS files in plain text format.

**Value**

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

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

AU	Authors
TI	Document Title
S0	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](#)

### See Also

[scopus2df](#) for converting SCOPUS Export file (in bibtex format)

[isibib2df](#) for converting ISI Export file (in bibtex format)

[isi2df](#) for converting ISI Export file (in plain text format)

[pubmed2df](#) for converting an object of the class pubmed (RISmed package)

Other converting functions: [bib2df](#), [cochrane2df](#), [isi2df](#), [isibib2df](#), [pubmed2df](#), [scopus2df](#)

### Examples

```
# An ISI or SCOPUS Export file can be read using \code{\link{readLines}} function:

# D <- readFiles('filename1.txt','filename2.txt','filename3.txt')

# filename1.txt, filename2.txt and filename3.txt are WoS or SCOPUS Export file
# in plain text or bibtex format.

# biblio <- readFiles('http://www.bibliometrix.org/datasets/bibliometrics_articles.txt')

data(biblio)

biblio_df_df <- convert2df(file = biblio, dbsource = "isi", format = "bibtex")
```

---

countries

*Index of Countries.*

---

### Description

Data frame containing a normalized index of countries.

Data are used by [biblioAnalysis](#) function to extract Country Field of Cited References and Authors.

### Format

A data frame with 197 rows and 4 variables:

**countries** country names

**continent** continent names

**Longitude** country centroid longitude

**Latitude** country centroid latitude

---

dominance                      *Authors' dominance ranking*

---

### Description

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

### Usage

```
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)
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", 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 this tags: TI (title), AB (abstract), UT (manuscript ID).
tol	is a numeric value giving the minimum relative similarity to match two manuscripts. Default value is tol = 0.95.

### 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 Thomson Reuters' ISI Web of Knowledge 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 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)

M=rbind(scientometrics[1:20,],scientometrics[10:30,])

newM <- duplicatedMatching(M, Field = "TI", tol = 0.95)

dim(newM)
```

---

garfield

*Eugene Garfield's manuscripts.*

---

**Description**

All manuscripts published by Eugene Garfield.  
Period: 1954 - 2014  
Database: [SCOPUS source](#)

**Format**

A data frame with 147 rows and 15 variables:

**AU** Authors

**TI** Document Title

**SO** Publication Name (or Source)

**JI** ISO Source Abbreviation

**DT** Document Type

**DE** Author Keywords

**ID** Keywords associated by ISI or SCOPUS database

**AB** Abstract

**C1** Author Address

**RP** Reprint Address

**CR** Cited References

**TC** Times Cited

**PY** Year

**UT** Unique Article Identifier

**DB** Database

**Source**

<http://www.scopus.com>



---

Hindex	<i>h-index calculation</i>
--------	----------------------------

---

### Description

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

### Usage

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

### Arguments

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge file.
field	is character. It can be equal to c("author", "source"). field indicates if H-index have to be calculated for a list of authors or for a list of sources. Default value is field = "author".
elements	is a character vector. It contains the the authors' names list or the source list for which you want to calculate the H-index. When the field is "author", the argument has the form C("SURNAME1 N", "SURNAME2 N",...), in other words, for each author: surname and initials separated by one blank space. i.e for the authors SEMPRONIO TIZIO CAIO and ARIA MASSIMO elements argument is elements = c("SEMPRONIO TC", "ARIA M").
sep	is the field separator character. This character separates authors in each string of AU column of the bibliographic data frame. The default is sep = ";".
years	is a integer. It indicates the number of years to consider for Hindex calculation. Default is 10.

### Value

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

### See Also

[convert2df](#) to import and convert an 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

```
### EXAMPLE 1: ###

data(scientometrics)

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)

indices=Hindex(garfield, field = "author", elements = "GARFIELD E", , sep = ";")

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

# Papers and total citations
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 = 0, sep = ";")
```

## Arguments

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Clarivate Analitics' Web of Knowledge file.
min.citations	is an integer. It sets the minimum number of citations for the documents included in the analysis. The default is min.citations = 0.
sep	is the field separator character. This character separates strings in CR column of the data frame. The default is sep = ";".

## Value

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

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

### See Also

[convert2df](#) to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.

[summary](#) to obtain a summary of the results.

[plot](#) to draw some useful plots of the results.

[biblioNetwork](#) to compute a bibliographic network.

### Examples

```
data(scientometrics)
```

```
histResults <- histNetwork(scientometrics, min.citations = 10, sep = ";")
```

---

histPlot

*Plotting historical co-citation network*

---

### Description

histPlot plots a historical co-citation network.

### Usage

```
histPlot(histResults, n = 20, size.cex = TRUE, size = 5,
  labelsize = 0.8, arrowsize = 0.1, edgesize = 2, color = 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 numebr of vertices to plot.

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

size is an integer. It define the point size of the vertices. Default value is 5.

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

arrowsize	is numerical. It indicates the edge arrow size.
edgesize	is numerical. It indicates the edge size.
color	is logical. If TRUE, edges are colored according to citing references.

### Details

The function `histPlot` can plot a historical co-citation network previously created by `histNetwork`.

### Value

It is a network object of the class `igraph`.

### 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

data(scientometrics)

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

net <- histPlot(histResults, n=20, size.cex=TRUE, size = 5, arrowsize=0.3)
```

---

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

<code>df</code>	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)
```

---

isi2df

*Convert an ISI WoK Export file into a data frame*

---

### Description

It converts an ISI Wok Export file and create a data frame from it, with cases corresponding to articles and variables to Field Tag in the original file.

### Usage

```
isi2df(D)
```

### Arguments

`D` is a character array containing data read from a ISI Export file (in plain text format).

**Value**

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

**See Also**

[scopus2df](#) for converting SCOPUS Export file (in bibtex format)

Other converting functions: [bib2df](#), [cochrane2df](#), [convert2df](#), [isibib2df](#), [pubmed2df](#), [scopus2df](#)

**Examples**

```
# A group of ISI Export files can be read using \link{readFiles} function:
# largechar <- readFiles('filename1.txt','filename2.txt','filename3.txt')

# scientometrics_text <- readFiles('http://www.bibliometrix.org/datasets/scientometrics.txt')
# data scientometrics_text
# scient_df <- isi2df(scientometrics_text)
```

---

isibib2df

---

*Convert a Clarivate Analytcs WoS Export file into a data frame*


---

**Description**

It converts a Clarivate Analytcs WoS Export file and create a data frame from it, with cases corresponding to articles and variables to Field Tag in the original file.

**Usage**

```
isibib2df(D)
```

**Arguments**

D is a character array containing data read from an WoS Export file (in bibtex format).

**Value**

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

**See Also**

[isi2df](#) for converting ISI Export file (in plain text format)

Other converting functions: [bib2df](#), [cochrane2df](#), [convert2df](#), [isi2df](#), [pubmed2df](#), [scopus2df](#)

**Examples**

```
# A ISI Export file can be read using \code{\link{readLines}} function:  
  
# largechar <- readFiles('filename1.bib','filename2.bib2,...')  
  
# filename.bib is a Clarivate Analytics WoS Export file in plain text format.  
  
# largechar <- readFiles('http://www.bibliometrix.org/datasets/ranking.bib')  
  
# ranking <- isibib2df(largechar)
```

---

isiCollection

*"Bibliometrics" manuscripts from ISI WOS.*

---

**Description**

Manuscripts including the term "bibliometrics" in the title.  
Period: 1985 - 2017  
Database: **ISI Web of Knowledge**  
Format: bibtex

**Format**

A data frame with 329 rows and 16 variables:

**AU** Authors

**TI** Document Title

**SO** Publication Name (or Source)

**JI** ISO Source Abbreviation

**DT** Document Type

**DE** Author Keywords

**ID** Keywords associated by ISI or SCOPUS 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

**Source**

<http://www.webofknowledge.com>

---

keywordAssoc	<i>ID and DE keyword associations</i>
--------------	---------------------------------------

---

**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 <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge 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 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)

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	<i>Yearly occurrences of top keywords/terms</i>
---------------	---

---

## Description

It calculates yearly occurrences of top keywords/terms.

## Usage

```
KeywordGrowth(M, Tag = "ID", sep = ";", top = 10, cdf = TRUE)
```

## Arguments

M	is a data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to articles and variables to Field Tag in the original ISI or SCOPUS file.
Tag	is a character object. It indicates one of the keyword field tags of the standard ISI WoS Field Tag codify (ID or DE) or a field tag created by <a href="#">termExtraction</a> function (TI_TM, AB_TM, etc.).
sep	is the field separator character. This character separates strings in each keyword column of the data frame. The default is sep = ";".
top	is a numeric. It indicates the number of top keywords to analyze. The default value is 10.
cdf	is a logical. If TRUE, the function calculates the cumulative occurrences distribution.

## Value

an object of class data.frame

## Examples

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

# Plotting results
#
# library(reshape2)
# library(ggplot2)
# DF=melt(topKW, id='Year')
# ggplot(DF,aes(Year,value, group=variable, color=variable))+geom_line()
```

---

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

---

### Description

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

### Usage

```
localCitations(M, sep = ";")
```

### Arguments

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge file.
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 = ";".

### 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)

CR <- localCitations(scientometrics, sep = ";")

CR$Authors[1:10,]
CR$Papers[1:10,]
```

---

lotka	<i>Lotka's law coefficient estimation</i>
-------	---

---

**Description**

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

**Usage**

```
lotka(results)
```

**Arguments**

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

**Value**

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

the list contains the following objects:

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

**See Also**

[biblioAnalysis](#) function for bibliometric analysis

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

**Examples**

```
data(scientometrics)
results <- biblioAnalysis(scientometrics)
L=lotka(results)
L
```

**Description**

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

**Usage**

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

**Arguments**

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

**Details**

bibliographic data frames are obtained by the converting function [convert2df](#). The function merges data frames identifying common tag fields and duplicated records.

**Value**

the value returned from mergeDbSources is a bibliographic data frame.

**See Also**

[convert2df](#) to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.

[biblioAnalysis](#) function for bibliometric analysis.

[summary](#) to obtain a summary of the results.

[plot](#) to draw some useful plots of the results.

**Examples**

```
data(isiCollection)
```

```
data(scopusCollection)
```

```
M <- mergeDbSources(isiCollection, scopusCollection, remove.duplicated=TRUE)
```

```
dim(M)
```

---

metaTagExtraction	<i>Meta-Field Tag Extraction</i>
-------------------	----------------------------------

---

### Description

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

### Usage

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

### Arguments

M	is a data frame obtained by the converting function <a href="#">convert2df</a> . 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. New tag extracted from aggregated data is specified by this string. Field can be equal to one of this tags:
"CR_AU"	First Author of each cited reference
"CR_SO"	Source of each cited reference
"AU_CO"	Country of affiliation for each co-author
"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

[scopus2df](#) for converting ISO or SCOPUS Export file into a data frame.

[biblioAnalysis](#) function for bibliometric analysis

### Examples

```
# Example 1: First Authors for each cited reference
data(scientometrics)
```

```

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_S0", sep = ";")
unlist(strsplit(scientometrics$CR_S0[1], ";"))

#Example 3: Affiliation country for co-author

data(scientometrics)
scientometrics <- metaTagExtraction(scientometrics, Field = "AU_CO", sep = ";")
scientometrics$AU_CO[1:10]

```

---

networkPlot

*Plotting Bibliographic networks*


---

## Description

networkPlot plots a bibliographic network.

## Usage

```

networkPlot(NetMatrix, normalize = NULL, n = NULL, degree = NULL,
  Title = "Plot", type = "kamada", label = TRUE, labelsize = 1,
  label.cex = FALSE, label.color = FALSE, label.n = NULL,
  halo = FALSE, cluster = "walktrap", 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)

```

## Arguments

NetMatrix	is a network matrix obtained by the function <a href="#">biblioNetwork</a> .
normalize	is a character. It can be "association", "jaccard", "inclusion", "salton" or "equivalence" to obtain Association Strength, Jaccard, Inclusion, Salton or Equivalence similarity index respectively. The default is type = NULL.
n	is an integer. It indicates the number of vertices to plot.
degree	is an integer. It indicates the min frequency of a vertex. If degree is not NULL, n is ignored.
Title	is a character indicating the plot title.

type	is a character object. It indicates the network map layout:
type="auto"	Automatic layout selection
type="circle"	Circle layout
type="sphere"	Sphere layout
type="mds"	Multidimensional Scaling layout
type="fruchterman"	Fruchterman-Reingold layout
type="kamada"	Kamada-Kawai layout
type="vosviewer"	Network is plotted using VOSviewer software
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", "infomap", "edge_betweenness", "walktrap").
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.

**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 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>
<code>cluster_res</code>	a data frame with main results of clustering procedure.

**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

data(scientometrics)

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

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

---

networkStat

*Calculating network summary statistics*


---

**Description**

`networkStat` calculates main network statistics.

**Usage**

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



## Arguments

object	is a network matrix obtained by the function <a href="#">biblioNetwork</a> 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")</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 <a href="#">communities</a> 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)

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

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

---

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 <a href="#">biblioNetwork</a> or <a href="#">cocMatrix</a> .
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.

### Value

a similarity matrix.

### See Also

[biblioNetwork](#) function to compute a bibliographic network.

[cocMatrix](#) to compute a bibliographic bipartite network.

## Examples

```
data(scientometrics)
NetMatrix <- biblioNetwork(scientometrics, analysis = "co-occurrences",
                           network = "keywords", sep = ";")
S=normalizeSimilarity(NetMatrix, type = "association")
```

---

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 set of plots of the object of class bibliometrix and a dataframe of citation analysis.

## See Also

The bibliometric analysis function [biblioAnalysis](#).  
[summary](#) to compute a list of summary statistics of the object of class bibliometrix.

## Examples

```
data(scientometrics)

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 = "inclusion")
```

### Arguments

Nodes is a list of nodes obtained by [thematicEvolution](#) function.  
Edges is a list of edges obtained by [thematicEvolution](#) function.  
measure is a character. It can be measure=("inclusion", "stability").

### Value

a sankeyPlot

### See Also

[thematicMap](#) function to create a thematic map based on co-word network analysis and clustering.

[thematicMap](#) function to perform a thematic evolution analysis.

[networkPlot](#) to plot a bibliographic network.

### Examples

```
data(scientometrics)
years=c(2000)

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

#plotThematicEvolution(nexus$Nodes, nexus$Edges)
```

---

`pubmed2df`*Convert a PubMed/MedLine collection into a data frame*

---

### Description

It converts a PubMed/MedLine collection (obtained through a query performed with RISmed package) and create a data frame from it, with cases corresponding to articles and variables to Field Tags as proposed by Clarivate Analytics WoS.

### Usage

```
pubmed2df(D)
```

### Arguments

`D` is an object of class `MedLine` (package "RISmed") containing data resulting of a query performed on MedLine using the package `RISmed`.

### Value

a data frame with cases corresponding to articles and variables to Field Tags as proposed by Clarivate Analytics WoS.

### See Also

[scopus2df](#) for converting SCOPUS Export file (in bibtex format)

[isi2df](#) for converting Clarivate Analytics WoS Export file (in plaintex format)

[isibib2df](#) for converting Clarivate Analytics WoS Export file (in bibtex format)

Other converting functions: [bib2df](#), [cochrane2df](#), [convert2df](#), [isi2df](#), [isibib2df](#), [scopus2df](#)

### Examples

```
# library(RISmed)
# search_topic <- 'epidermolysis bullosa'
# search_query <- EUtilsSummary(search_topic, retmax=200, mindate=2014, maxdate=2014)
# summary(search_query)
# D <- EUtilsGet(search_query)

# M <- pubmed2df(D)
```

---

readFiles	<i>Load a sequence of ISI or SCOPUS Export files into a large character object</i>
-----------	--

---

### Description

It loads a sequence of SCOPUS and Thomson Reuters' ISI Web of Knowledge export files and create a large character vector from it.

### Usage

```
readFiles(...)
```

### Arguments

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

### Value

a character vector of length the number of lines read.

### See Also

[convert2df](#) for converting SCOPUS of ISI Export file into a dataframe

### Examples

```
# ISI or SCOPUS Export files can be read using \code{\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('http://www.bibliometrix.org/datasets/bibliometrics_articles.txt')
```

---

retrievalByAuthorID	<i>Get Author Content on SCOPUS by ID</i>
---------------------	---

---

### Description

Uses SCOPUS API search to get information about documents on a set of authors using SCOPUS ID.

**Usage**

```
retrievalByAuthorID(id, api_key, remove.duplicated = TRUE,
  country = TRUE)
```

**Arguments**

`id` is a vector of characters containing the author's SCOPUS IDs. SCOPUS IDs can be obtained using the function [idByAuthor](#).

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

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

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

**Value**

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

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

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

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

**See Also**

[idByAuthor](#) for downloading auhtor 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.

(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.)

## Usage

```
rpys(M, sep = ";", timespan = NULL, graph = T)
```

## Arguments

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



sep	is the cited-references separator character. This character separates cited-references in the CR column of the data frame. The default is sep = ";".
timespan	is a numeric vector c(min year,max year). The default value is NULL (the entire timespan is considered).
graph	is a logical. If TRUE the function plot the spectrography otherwise the plot is created but not drawn down.

**Value**

a list containing the spectroscopy (class ggplot2) and two dataframes with the number of citation per year and the list of the cited-references for each 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**

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

---

scientometrics      *"Co-citation analysis" and "Coupling analysis" manuscripts.*

---

**Description**

Manuscripts about the topics "co-citation analysis" and "coupling analysis" published on Scientometrics Journal.

Period: 1985 - 2015

Database: [ISI Web of Knowledge](#)

**Format**

A data frame with 147 rows and 17 variables:

**AU** Authors

**TI** Document Title

**SO** Publication Name (or Source)

**JJ** ISO Source Abbreviation

**DT** Document Type

**DE** Author Keywords  
**ID** Keywords associated by ISI or SCOPUS 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  
**SR** Short Reference

**Source**

<http://www.webofknowledge.com>

---

scientometrics\_text    *"Co-citation analysis" and "Coupling analysis" manuscripts.*

---

**Description**

Manuscripts about the topics "co-citation analysis" and "coupling analysis" published on Scientometrics Journal.  
Period: 1985 - 2015  
Database: [ISI Web of Knowledge](#)

**Format**

A large character with 12731 rows.  
Data has been imported by an ISI Export file in plain text format using the function [readLines](#).

**Source**

<http://www.webofknowledge.com>

---

`scopus2df`*Convert a SCOPUS Export file into a data frame*

---

### Description

It converts a SCOPUS Export file and create a data frame from it, with cases corresponding to articles and variables to Field Tag in the original file.

### Usage

```
scopus2df(D)
```

### Arguments

`D` is a character array containing data read from a SCOPUS Export file (in bibtext format).

### Value

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

### See Also

[isi2df](#) for converting ISI Export file (in plain text format)

Other converting functions: [bib2df](#), [cochrane2df](#), [convert2df](#), [isi2df](#), [isibib2df](#), [pubmed2df](#)

### Examples

```
# A SCOPUS Export file can be read using \link{readFiles} function:
# largechar <- readFiles('filename1.bib','filename2.bib2,...')
# filename.bib is a SCOPUS Export file in plain text format.
#largechar <- readFiles('http://www.bibliometrix.org/datasets/scopus.bib')

#scopus_df <- scopus2df(largechar)
```

---

scopusCollection      *"Bibliometrics" manuscripts from SCOPUS.*

---

**Description**

Manuscripts including the term "bibliometrics" in the title.  
Period: 1975 - 2017  
Database: **SCOPUS**  
Format: bibtex

**Format**

A data frame with 487 rows and 15 variables:

**AU** Authors

**TI** Document Title

**SO** Publication Name (or Source)

**JI** ISO Source Abbreviation

**DT** Document Type

**DE** Author Keywords

**ID** Keywords associated by ISI or SCOPUS database

**AB** Abstract

**C1** Author Address

**RP** Reprint Address

**CR** Cited References

**TC** Times Cited

**PY** Year

**UT** Unique Article Identifier

**DB** Database

**Source**

<http://www.scopus.com>

---

sourceGrowth	<i>Number of documents published annually per Top Sources</i>
--------------	---

---

### 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 <a href="#">convert2df</a> . 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)
topS0=sourceGrowth(scientometrics, top=1, cdf=TRUE)
topS0

# Plotting results
#
# library(reshape2)
# library(ggplot2)
# DF=melt(topS0, id='Year')
# ggplot(DF,aes(Year,value, group=variable, color=variable))+geom_line()
```

---

stopwords	<i>List of English stopwords.</i>
-----------	-----------------------------------

---

**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	<i>Summarizing bibliometric analysis results</i>
----------------------	--

---

**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)

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)

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

---

tableTag

*Tabulate elements from a Tag Field column*


---

**Description**

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

**Usage**

```
tableTag(M, Tag = "CR", sep = ";")
```

**Arguments**

M	is a data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to articles and variables to Field Tag in the original ISI or SCOPUS file.
Tag	is a character object. It indicates one of the field tags of the standard ISI WoS Field Tag codify.
sep	is the field separator character. This character separates strings in each column of the data frame. The default is sep = ";".

**Details**

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

**Value**

an object of class table

**Examples**

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



---

termExtraction	<i>Term extraction tool from textual fields of a manuscript</i>
----------------	---

---

### Description

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

### Usage

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

### Arguments

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

**Field** is a character object. It indicates the field tag of textual data :

"TI"	Manuscript title
"AB"	Manuscript abstract
"ID"	Manuscript keywords plus
"DE"	Manuscript author's keywords

The default is `Field = "TI"`.

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

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

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

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

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

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

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

**Value**

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

**See Also**

[convert2df](#) to import and convert an ISI or SCOPUS Export file in a bibliographic data frame.  
[biblioAnalysis](#) function for bibliometric analysis

**Examples**

```
# Example 1: Term extraction from titles

data(scientometrics)

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

# term extraction
scientometrics <- termExtraction(scientometrics, Field = "TI",
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)

# vector of terms to remove
remove.terms=c("analysis","bibliographic")

# term extraction
scientometrics <- termExtraction(scientometrics, Field = "AB", stemming=TRUE,language="english",
remove.numbers=TRUE, remove.terms=remove.terms, 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",
synonyms=synonyms, verbose=TRUE)
```

---

thematicEvolution	<i>Perform a Thematic Evolution Analysis</i>
-------------------	--

---

**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)
```

**Arguments**

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> .
field	is a character object. It indicates the content field to use. Field can be one of c("ID","DE","TI","AB"). Default value is field="ID".
years	is a numeric vector of two or more unique cut points.
n	is numerical. It indicates the number of words to use in the network analysis
minFreq	is numerical. It indicates the min frequency of words included in to a cluster.

**Details**

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

**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**

```
data(scientometrics)
years=c(2000)
```

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

---

thematicMap	<i>Create a thematic map</i>
-------------	------------------------------

---

### 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(Net, NetMatrix, S = NULL, minfreq = 5)
```

### Arguments

Net	is a igraph object created by <a href="#">networkPlot</a> function.
NetMatrix	is a co-occurrence matrix obtained by the network functions <a href="#">biblioNetwork</a> or <a href="#">cocMatrix</a> .
S	is a similarity matrix obtained by the <a href="#">normalizeSimilarity</a> function. If S is NULL, map is created using co-occurrence counts.
minfreq	is a integer. It indicates the minimum frequency of a cluster.

### Details

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

### 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

### See Also

[biblioNetwork](#) function to compute a bibliographic network.

[cocMatrix](#) to compute a bibliographic bipartite network.

[networkPlot](#) to plot a bibliographic network.

### Examples

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

```
S <- normalizeSimilarity(NetMatrix, type = "association")
net <- networkPlot(S, n = 100, Title = "co-occurrence network", type="fruchterman",
  labelsize = 0.7, halo = FALSE, cluster = "walktrap", remove.isolates=FALSE,
  remove.multiple=FALSE, noloops=TRUE, weighted=TRUE)
res <- thematicMap(net, NetMatrix, S)
plot(res$map)
```

---

timeslice

*Bibliographic data frame time slice*


---

## Description

Divide a bibliographic data frame into time slice

## Usage

```
timeslice(M, breaks = NA, k = 5)
```

## Arguments

M	is a bibliographic data frame obtained by the converting function <a href="#">convert2df</a> . It is a data matrix with cases corresponding to manuscripts and variables to Field Tag in the original SCOPUS and Thomson Reuters' ISI Web of Knowledge file.
breaks	is a numeric vector of two or more unique cut points.
k	is a 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)

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)
```

# Index

\*Topic **Science Mapping, Bibliometrics, Clarivate Analytics Web of Science, Scopus, citations, co-citation, citation-network, coupling, co-authors, co-occurrence, collaboration, co-word analysis**

bibliometrix-package, 3

authorProdOverTime, 6

bib2df, 7, 16, 21, 30, 45, 51

biblio, 8

biblio\_df, 12

biblioAnalysis, 7, 8, 11, 14, 15, 17, 19, 21–23, 25, 28, 32, 34–37, 40, 41, 43, 49, 54–56, 58, 61

bibliometrix (bibliometrix-package), 3

bibliometrix-package, 3

biblioNetwork, 10, 10, 17, 19, 27, 38, 40–42, 49, 60

biblioshiny, 12

bibtag, 13

bradford, 13

citations, 14, 34

cochrane2df, 7, 15, 21, 30, 45, 51

cocMatrix, 11, 16, 19, 28, 40–42, 59, 60

communities, 40, 41

conceptualStructure, 18

convert2df, 6, 7, 9–11, 14, 16–18, 19, 23, 25–27, 30, 32–34, 36, 37, 45, 46, 48, 49, 51, 53, 56–59, 61

countries, 21

dominance, 22

duplicatedMatching, 23

garfield, 24

Hindex, 25

histNetwork, 26, 27, 28

histPlot, 27, 28

idByAuthor, 28, 47

isi2df, 7, 16, 21, 29, 30, 45, 51

isibib2df, 7, 16, 21, 30, 30, 45, 51

isiCollection, 31

keywordAssoc, 32

KeywordGrowth, 33

localCitations, 34

lotka, 35

Matrix, 11, 17

mergeDbSources, 35

metaTagExtraction, 37

networkPlot, 38, 40, 44, 59, 60

networkStat, 40, 41

normalizeSimilarity, 42, 60

plot, 9, 15, 23, 25, 27, 32, 34, 36, 55, 61

plot.bibliometrix, 43

plotThematicEvolution, 44

pubmed2df, 7, 16, 21, 30, 45, 51

readFiles, 46

readLines, 8, 50

retrievalByAuthorID, 29, 46

rphys, 48

scientometrics, 49

scientometrics\_text, 50

scopus2df, 7, 16, 21, 30, 37, 45, 51

scopusCollection, 52

sourceGrowth, 53

stopwords, 54

summary, 7, 9, 14, 15, 22, 23, 25, 27, 32, 34–36, 43, 61



summary.bibliometrix, [54](#)  
summary.bibliometrix\_netstat, [55](#)

tableTag, [56](#)  
termExtraction, [19](#), [33](#), [57](#)  
thematicEvolution, [44](#), [59](#), [59](#)  
thematicMap, [44](#), [59](#), [60](#)  
timeslice, [61](#)  
trim, [62](#)  
trim.leading, [62](#)