Performing Literature Reviews with nails package

Juho Salminen and Antti Knutas

2017-12-17

Introduction

NAILS performs statistical and Social Network Analysis (SNA) on citation data. SNA is a new way for researchers to map large datasets and get insights from new angles by analyzing connections between articles. As the amount of publications grows on any given field, automatic tools for this sort of analysis are becoming increasingly important prior to starting research on new fields. NAILS also provides useful data when performing Systematic Mapping Studies (SMS) in scientific literature. According to Kitchenham et al. performing a SMS can be especially suitable if few literature reviews have been done on the topic and there is a need to get a general overview of the field of interest.

The nails package provides functionality for parsing Web of Science data for quantitative Systematic Mapping Study analysis, and a series of custom statistical and network analysis functions to give the user an overview of literature datasets. The features can be divided into two primary sections: Firstly, statistical analysis, which for example gives an overview of publication frequencies, most published authors and journals. Secondly, the more novel network analysis, which gives further insight into relationship between the interlinked citations and cooperation between authors. For example, the most basic features can use citation network analysis identify the most cited authors and publication forums. Lastly, it provides a few convenience functions to use the topicmodels and stm packages to create Latent Dirichlet allocation -based topic models.

For further details see the following article: Knutas, A., Hajikhani, A., Salminen, J., Ikonen, J., Porras, J., 2015. Cloud-Based Bibliometric Analysis Service for Systematic Mapping Studies. CompSysTech 2015.

Example workflow and report

In this section we present how to load Web of Science data using nails package functions and then to create an example report using ggplot2-based visualizations.

Loading data

Below is an example of how data exported from Web of Science can be loaded and parsed using the nails package functions.

```
# Setup

# Load packages
devtools::load_all()
require(ggplot2)

# Set ggplot theme
theme_set(theme_minimal(12))

# Load data
literature <- read_wos_data("../tests/testthat/test_data")
```

```
# Clean data
literature <- clean_wos_data(literature)
```

Generating visualizations with knittr

Below we present how to generate example report using nails calls and then using ggplot2 and knittr to generate visual reports.

This report provides an analysis on the records downloaded from Web of Science. The analysis identifies the important authors, journals, and keywords in the dataset based on the number of occurences and citation counts. A citation network of the provided records is created and used to identify the important papers according to their in-degree, total citation count and PageRank scores. The analysis finds also often-cited references that were not included in the original dataset downloaded from the Web of Science.

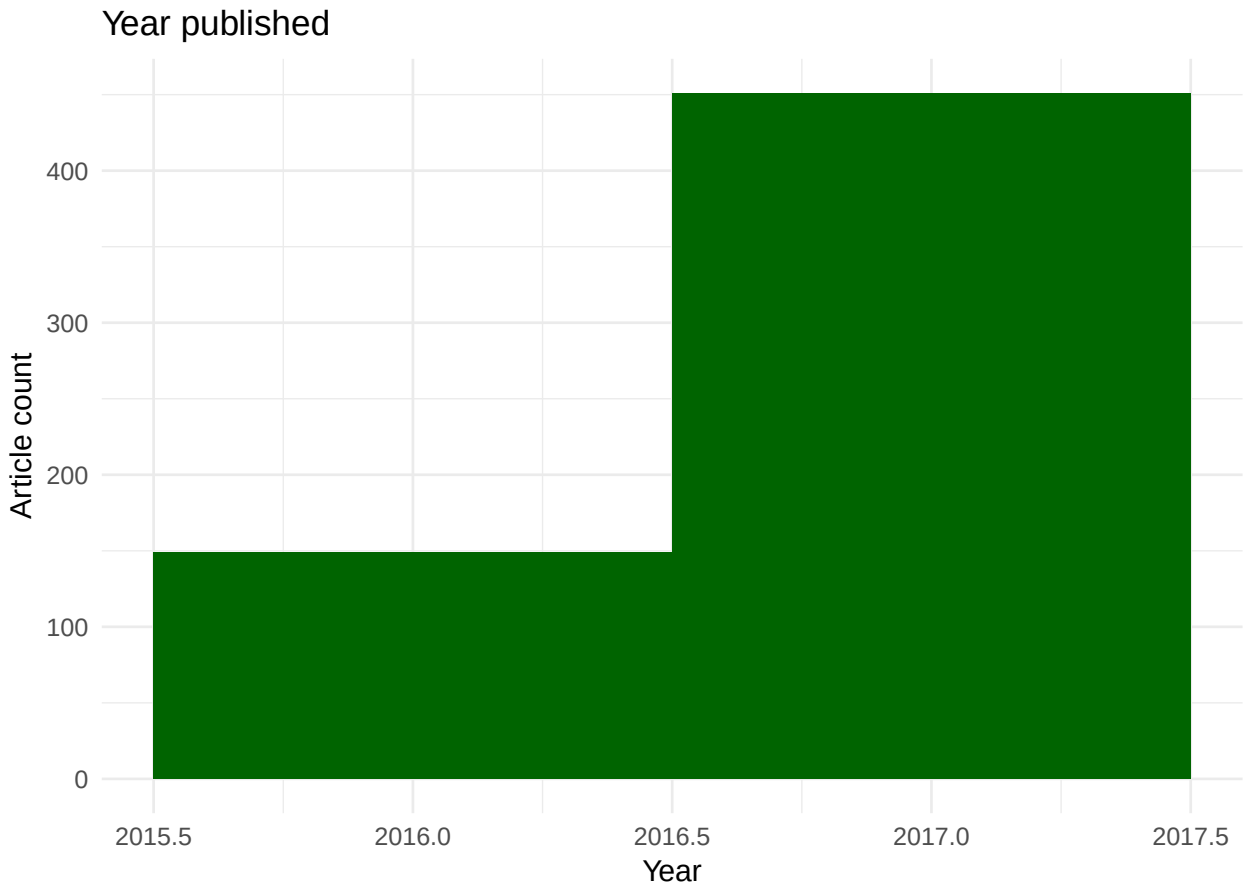
Reports can also be generated by using the online analysis service, and the source code is available at GitHub. Instructions and links to tutorial videos can be found at the project page. Please consider citing our research paper on bibliometrics at if you publish the analysis results.

```
# Setup

# Load packages
devtools::load_all()
require(ggplot2)

# Set ggplot theme
theme_set(theme_minimal(12))
```

The analysed dataset, loaded in section “loading data”, consist of 600 records with 69 variables. More information about the variables can be found at Web of Science.

Publication years

```
ggplot(literature, aes(YearPublished)) +
    geom_histogram(binwidth = 1, fill = "darkgreen") +
    ggtitle("Year published") + xlab("Year") +
    ylab("Article count")
```




```
# Calculate relative publication counts
# yearTable <-
# as.data.frame(table(literature$YearPublished))
# names(yearDF) <- c('Year', 'Freq') #
# Fix column names

# Merge to dataframe of total publication
# numbers (years) yearDF <- merge(yearDF,
# years, by.x = 'Year', by.y = 'Year',
# all.x = TRUE) yearDF$Year <-
# as.numeric(as.character(yearDF$Year)) #
# factor to numeric Calculate published
# articles per total articles by year
# yearDF$Fraction <- yearDF$Freq /
# yearDF$Records
```

Relative publication volume

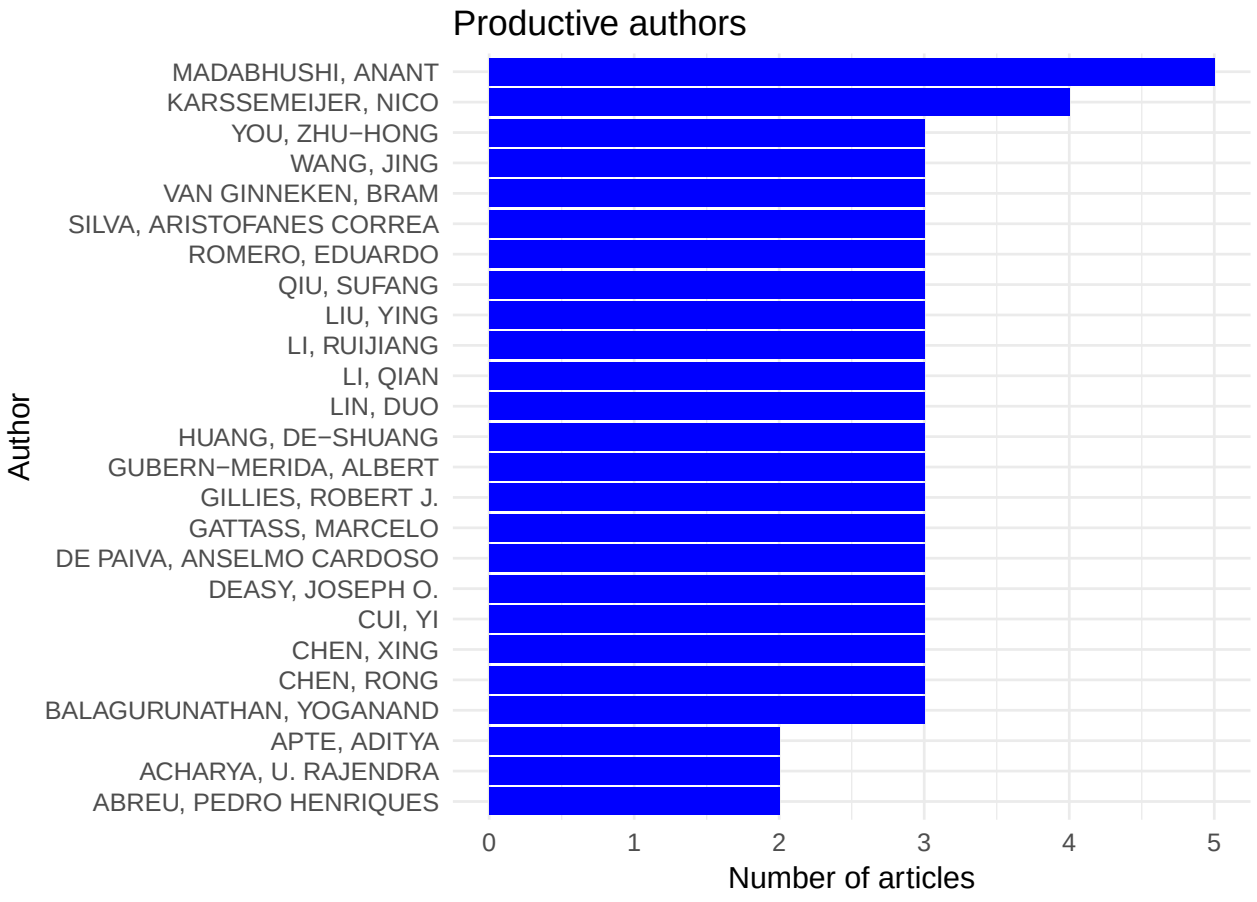
```
# ADD PLOT HERE!
print("Placeholder")
```

```
## [1] "Placeholder"
```

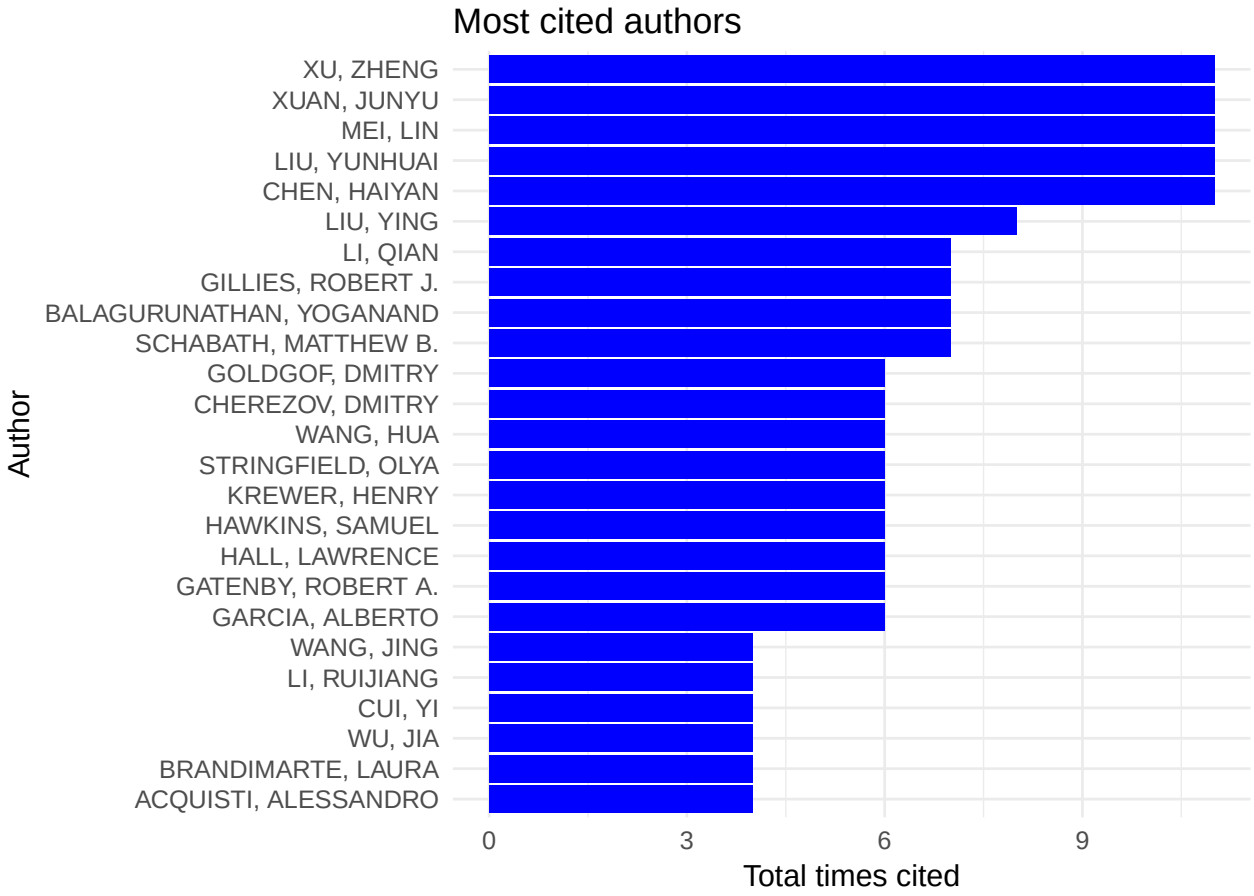
Important authors

Sorted by the number of articles published and by the total number of citations.

```
# Get author network nodes, which contain
# the required information
author_network <- get_author_network(literature)
author_nodes <- author_network$author_nodes
# Change Id to AuthorFullName
names(author_nodes)[names(author_nodes) ==
    "Id"] <- "AuthorFullName"

# Sort by number of articles by author
author_nodes <- author_nodes[with(author_nodes,
    order(-Freq)), ]
# Re-order factor levels
author_nodes <- transform(author_nodes, AuthorFullName = reorder(AuthorFullName,
    Freq))

ggplot(head(author_nodes, 25), aes(AuthorFullName,
    Freq)) + geom_bar(stat = "identity",
    fill = "blue") + coord_flip() + ggtitle("Productive authors") +
    xlab("Author") + ylab("Number of articles")
```



```
# Reorder AuthorFullName factor according
# to TotalTimesCited (decreasing order)
author_nodes <- transform(author_nodes, AuthorFullName = reorder(AuthorFullName,
    TotalTimesCited))

# Sort by number of articles by author
author_nodes <- author_nodes[with(author_nodes,
    order(-TotalTimesCited)), ]

ggplot(head(author_nodes, 25), aes(AuthorFullName,
    TotalTimesCited)) + geom_bar(stat = "identity",
    fill = "blue") + coord_flip() + ggtitle("Most cited authors") +
    xlab("Author") + ylab("Total times cited")
```

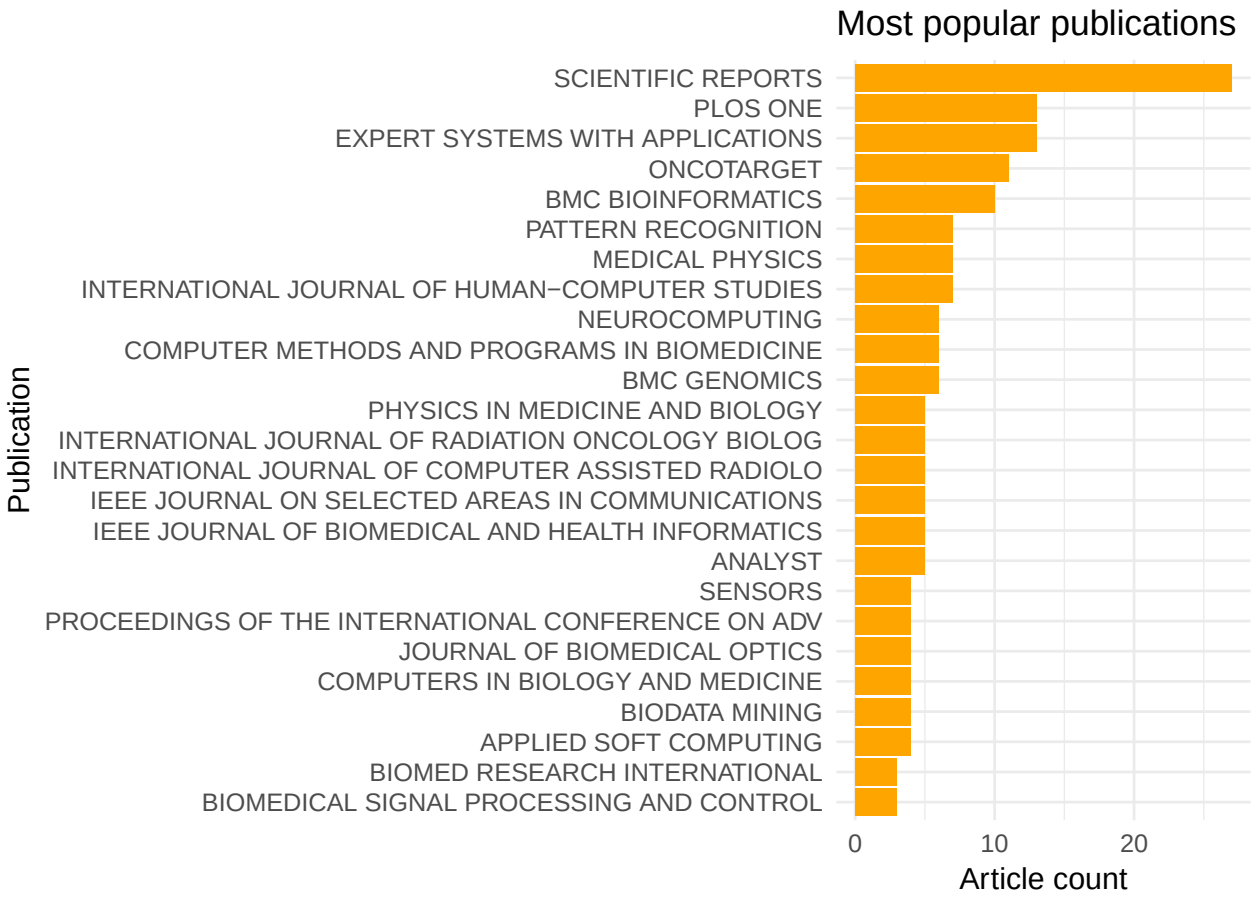


Important publications

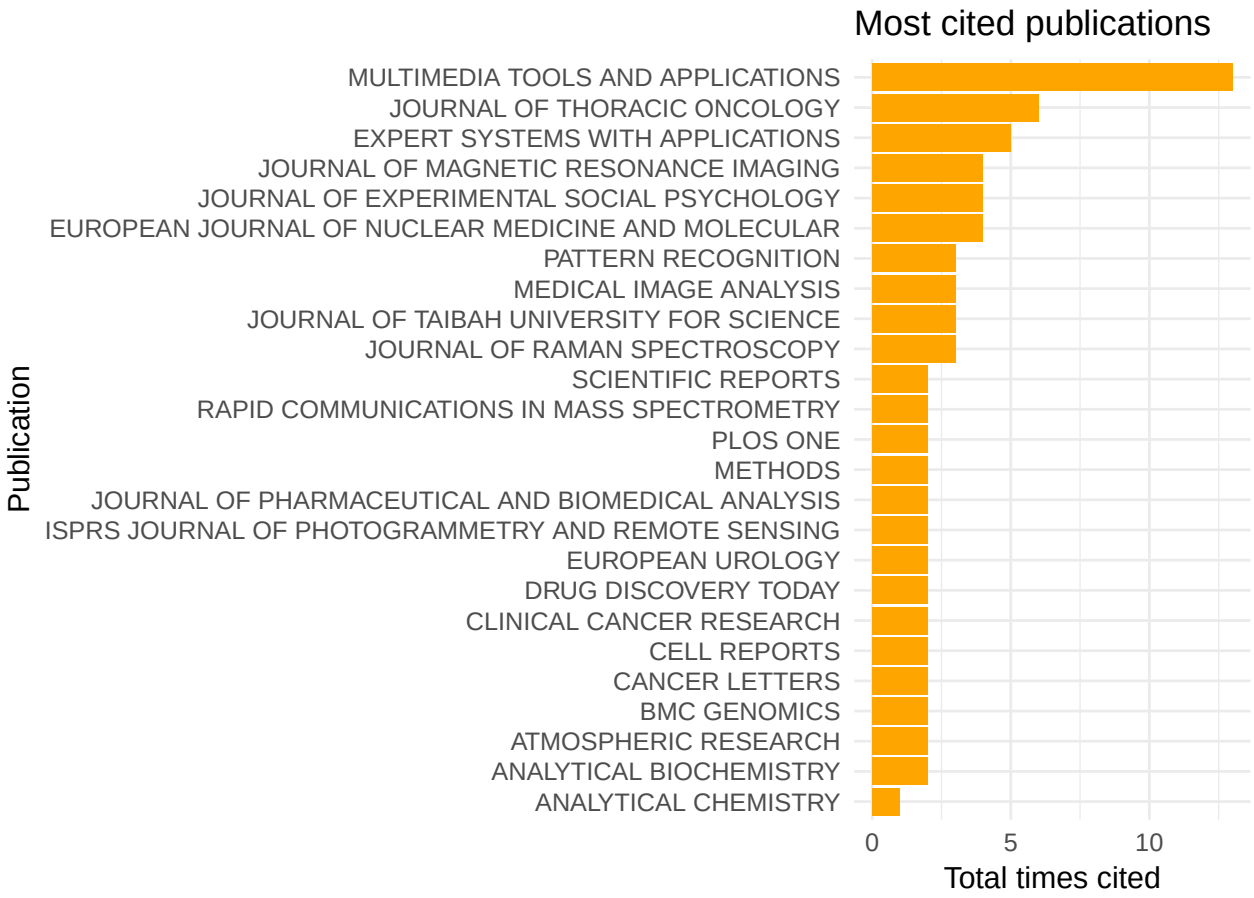
Sorted by number of published articles in the dataset and by the total number of citations.

```
# Calculate publication occurences
publications <- as.data.frame(table(literature$PublicationName))

# Fix names
names(publications) <- c("Publication", "Count")
```

```r
# Trim publication name to maximum of 50
# characters for displaying in plot
publications$Publication <- strtrim(publications$Publication,
    50)

# Sort descending
publications <- publications[with(publications,
    order(-Count)), ]

# Reorder factor levels
publications <- transform(publications, Publication = reorder(Publication,
    Count))


# WHY???  literature <- merge(literature,
# citation_sums, by = 'PublicationName' )

ggplot(head(publications, 25), aes(Publication,
    Count)) + geom_bar(stat = "identity",
    fill = "orange") + coord_flip() + theme(legend.position = "none") +
    ggtitle("Most popular publications") +
    xlab("Publication") + ylab("Article count")
```



```
# Calculating total citations for each
# publication.
citation_sums <- aggregate(literature$TimesCited,
    by = list(PublicationName = literature$PublicationName),
    FUN = sum, na.rm = T)

# Fix column names
names(citation_sums) <- c("PublicationName",
    "PublicationTotalCitations")

# Trim publication name to maximum of 50
# characters for displaying in plot
citation_sums$PublicationName <- strtrim(citation_sums$PublicationName,
    50)

# Sort descending and reorder factor
# levels accordingly
citation_sums <- citation_sums[with(citation_sums,
    order(-PublicationTotalCitations)), ]
citation_sums <- transform(citation_sums,
    PublicationName = reorder(PublicationName,
        PublicationTotalCitations))
ggplot(head(citation_sums, 25), aes(PublicationName,
    PublicationTotalCitations)) + geom_bar(stat = "identity",
    fill = "orange") + coord_flip() + theme(legend.position = "none") +
    ggtitle("Most cited publications") +
    xlab("Publication") + ylab("Total times cited")
```

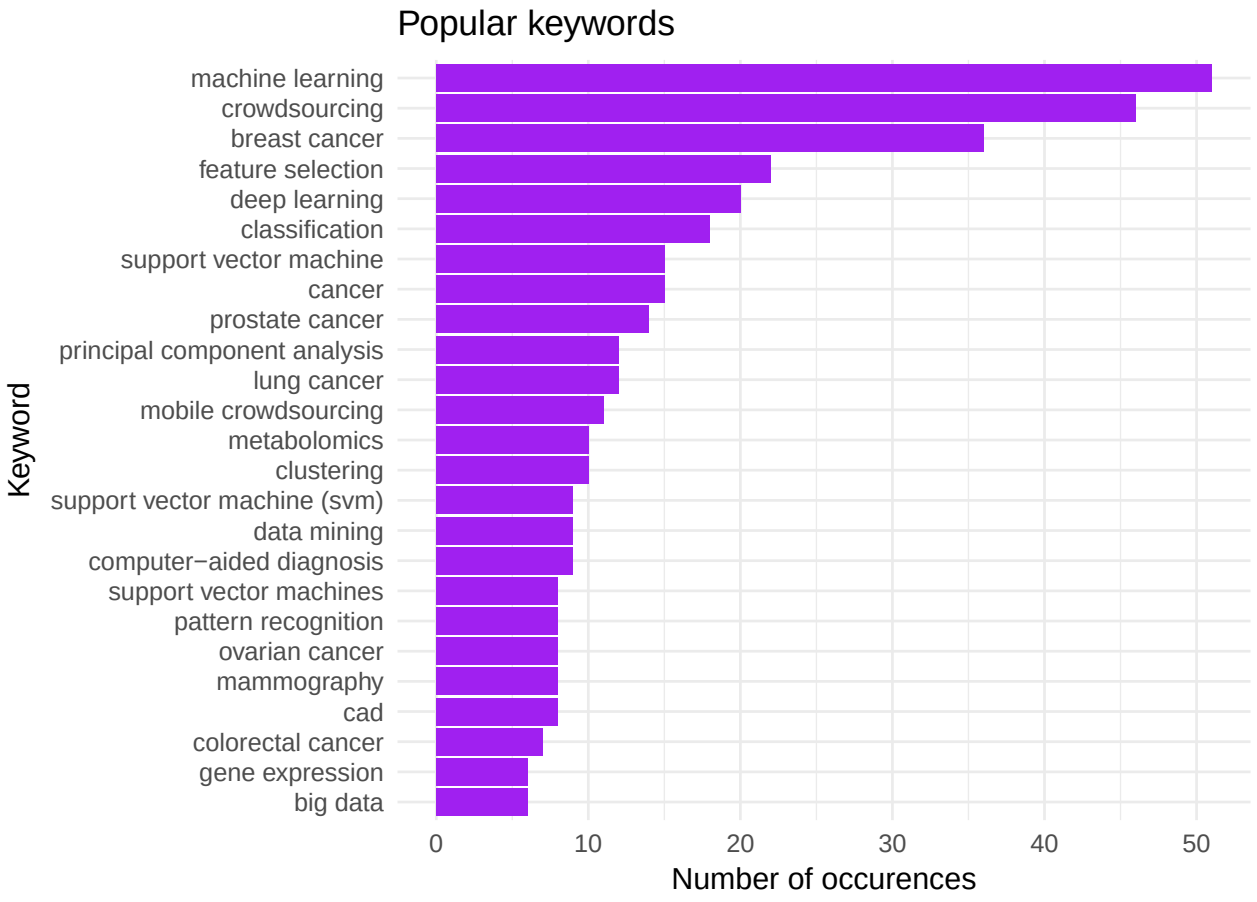


Important keywords

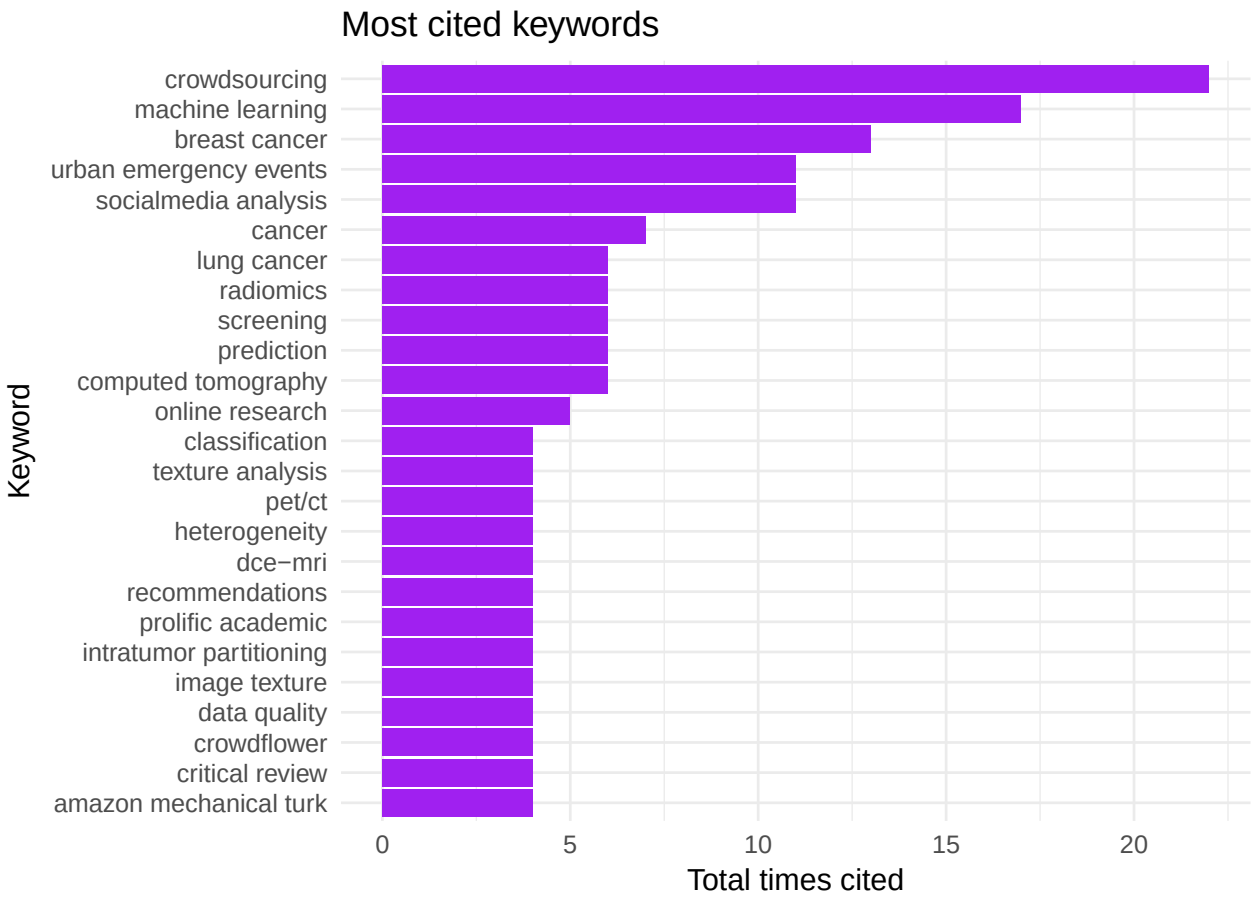
Sorted by the number of articles where the keyword is mentioned and by the total number of citations for the keyword.

```
# Calculating total citations for each
# keyword

literature_by_keywords <- arrange_by(literature,
    "AuthorKeywords")



# Sometimes AuthorKeywords column is
# empty.  Following if-else hack prevents
# crashing in those situations, either by
# using KeywordsPlus column or skipping
# keyword analysis.
if (nrow(literature_by_keywords) == 0) {
    cat("No keywords.")
} else {
    keyword_citation_sum <- aggregate(literature_by_keywords$TimesCited,
        by = list(AuthorKeywords = literature_by_keywords$AuthorKeywords),
        FUN = sum, na.rm = T)
```

```
    names(keyword_citation_sum) <- c("AuthorKeywords",
        "TotalTimesCited")

    keywords <- unlist(strsplit(literature$AuthorKeywords,
        ";"))
    keywords <- trim(keywords)
    keywords <- as.data.frame(table(keywords))
    names(keywords) <- c("AuthorKeywords",
        "Freq")

    keywords <- merge(keywords, keyword_citation_sum,
        by = "AuthorKeywords")
    keywords <- keywords[with(keywords, order(-Freq)),
        ]
    keywords <- transform(keywords, AuthorKeywords = reorder(AuthorKeywords,
        Freq))

    ggplot(head(keywords, 25), aes(AuthorKeywords,
        Freq)) + geom_bar(stat = "identity",
        fill = "purple") + coord_flip() +
        ggtitle("Popular keywords") + xlab("Keyword") +
        ylab("Number of occurences")
}
```



```
if (nrow(literature_by_keywords) > 0) {
    keywords <- keywords[with(keywords, order(-TotalTimesCited)),
        ]
    keywords <- transform(keywords, AuthorKeywords = reorder(AuthorKeywords,
        TotalTimesCited))
    ggplot(head(keywords, 25), aes(AuthorKeywords,
        TotalTimesCited)) + geom_bar(stat = "identity",
        fill = "purple") + coord_flip() +
        ggtitle("Most cited keywords") +
        xlab("Keyword") + ylab("Total times cited")
}
```



Important papers

The most important papers and other sources are identified below using three importance measures: 1) in-degree in the citation network, 2) citation count provided by Web of Science (only for papers included in the dataset), and 3) PageRank score in the citation network. The top 25 highest scoring papers are identified using these measures separately. The results are then combined and duplicates are removed. Results are sorted by in-degree, and ties are first broken by citation count and then by the PageRank.

When a Digital Object Identifier (DOI) is available, the full paper can be found using Resolve DOI website.

```
# Extract citation nodes
citation_network <- get_citation_network(literature)
```

```
citation_nodes <- citation_network$citation_nodes


# Extract the articles included in the
# data set and articles not included in
# the dataset
citations_lit <- citation_nodes[citation_nodes$Origin ==
    "literature", ]
citations_ref <- citation_nodes[citation_nodes$Origin ==
    "reference", ]

# Create article strings (document title,
# reference information and abstract
# separated by '|')
citations_lit$Article <- paste(toupper(citations_lit$DocumentTitle),
    " | ", citations_lit$FullReference, " | ",
    citations_lit$Abstract)
```

Included in the dataset

These papers were included in the 600 records downloaded from the Web of Science.

```
# Sort citations_lit by TimesCited,
# decreasing
citations_lit <- citations_lit[with(citations_lit,
    order(-TimesCited)), ]
# Extract top 25
top_lit <- head(citations_lit, 25)
# Sort by InDegree, decreasing
citations_lit <- citations_lit[with(citations_lit,
    order(-InDegree)), ]
# Add to list of top 25 most cited papers
top_lit <- rbind(top_lit, head(citations_lit,
    25))
# Sort by PageRank, decreasing
citations_lit <- citations_lit[with(citations_lit,
    order(-PageRank)), ]
# Add to list of most cited and highest
# InDegree papers
top_lit <- rbind(top_lit, head(citations_lit,
    25))
# Remove duplicates
top_lit <- top_lit[!duplicated(top_lit[,
    "FullReference"]), ]
# Sort top_lit by InDegree, break ties by
# TimesCited, then PageRank.
top_lit <- top_lit[with(top_lit, order(-InDegree,
    -TimesCited, -PageRank)), ]
# Print list
knitr::kable(top_lit[, c("Article", "InDegree",
    "TimesCited", "PageRank")])
```

	Article
31810	DIFFERENTIATION OF DIGESTIVE SYSTEM CANCERS BY USING SERUM PROTEIN-BASED SURFACE-
4109	PREDICTING MALIGNANT NODULES FROM SCREENING CT SCANS | HAWKINS S, 2016, J THORAC ON
46310	INTRATUMOR PARTITIONING AND TEXTURE ANALYSIS OF DYNAMIC CONTRAST-ENHANCED (DCE
35110	MULTI-CROP CONVOLUTIONAL NEURAL NETWORKS FOR LUNG NODULE MALIGNANCY SUSPICIOU
34710	LARGE SCALE DEEP LEARNING FOR COMPUTER AIDED DETECTION OF MAMMOGRAPHIC LESIONS
16647	CORRELATION OF LIPIDOMIC COMPOSITION OF CELL LINES AND TISSUES OF BREAST CANCER PA
8125	ESTIMATING PERSONALIZED DIAGNOSTIC RULES DEPENDING ON INDIVIDUALIZED CHARACTERIS
34310	DIFFERENTIATING TUMOR HETEROGENEITY IN FORMALIN-FIXED PARAFFIN-EMBEDDED (FFPE) F
9812	DISSECTING TARGET TOXIC TISSUE AND TISSUE SPECIFIC RESPONSES OF IRINOTECAN IN RATS U
55310	CROWDSOURCING BASED SOCIAL MEDIA DATA ANALYSIS OF URBAN EMERGENCY EVENTS | XU Z,
34610	CHARACTERIZATION OF PET/CT IMAGES USING TEXTURE ANALYSIS: THE PAST, THE PRESENTA. .
5659	BEYOND THE TURK: ALTERNATIVE PLATFORMS FOR CROWDSOURCING BEHAVIORAL RESEARCH |
45010	MODELLING THE CYTOTOXIC ACTIVITY OF PYRAZOLO-TRIAZOLE HYBRIDS USING DESCRIPTORS
18816	DESIGN OF EFFICIENT COMPUTATIONAL WORKFLOWS FOR IN SILICO DRUG REPURPOSING | VANE
2465	PAN-CANCER IMMUNOGENOMIC ANALYSES REVEAL GENOTYPE-IMMUNOPHENOTYPE RELATIONSI
3531	FUZZY CLUSTER BASED NEURAL NETWORK CLASSIFIER FOR CLASSIFYING BREAST TUMORS IN U
36910	CHEMICAL COMPOSITION AND SOURCE APPORTIONMENT OF PM2.5 DURING CHINESE SPRING FES
41110	FEATURE SELECTION METHODS FOR BIG DATA BIOINFORMATICS: A SURVEY FROM THE SEARCH I
4333	NANOSCOPIC TUMOR TISSUE DISTRIBUTION OF PLATINUM AFTER INTRAPERITONEAL ADMINISTI
4729	BIG DATA AND MACHINE LEARNING IN RADIATION ONCOLOGY: STATE OF THE ART AND FUTURE
48410	A COMPUTATIONAL APPROACH FOR DETECTING PIGMENTED SKIN LESIONS IN MACROSCOPIC IM
49010	IDENTIFICATION AND COMPARATIVE ORIDONIN METABOLISM IN DIFFERENT SPECIES LIVER MICI
54410	GAMIFYING COLLECTIVE HUMAN BEHAVIOR WITH GAMEFUL DIGITAL RHETORIC | SAKAMOTO M
55110	RULE-GUIDED HUMAN CLASSIFICATION OF VOLUNTEERED GEOGRAPHIC INFORMATION | ALI AL,
36010	ICAGES: INTEGRATED CANCER GENOME SCORE FOR COMPREHENSIVELY PRIORITIZING DRIVER C
2912	SUBGROUPS OF CASTRATION-RESISTANT PROSTATE CANCER BONE METASTASES DEFINED THROU
5851	UNTARGETED LC-HRMS-BASED METABOLOMICS FOR SEARCHING NEW BIOMARKERS OF PANCREA
7739	WESTERN DIETARY PATTERN INCREASES, AND PRUDENT DIETARY PATTERN DECREASES, RISK O
8033	A SURVEY ON SEMI-SUPERVISED FEATURE SELECTION METHODS | SHEIKHPOUR R, 2017, PATTERN
9551	CURE-SMOTE ALGORITHM AND HYBRID ALGORITHM FOR FEATURE SELECTION AND PARAMETER
49110	LIPIDOMIC PROFILING OF LUNG PLEURAL EFFUSION IDENTIFIES UNIQUE METABOTYPE FOR EGF

Not included in the dataset

These papers and other references were not among the 600 records downloaded from the Web of Science.

```
# Sort citations_ref by InDegree,
# decreasing
citations_ref <- citations_ref[with(citations_ref,
    order(-InDegree)), ]
# Extract top 25
top_ref <- head(citations_ref, 25)
# Sort by PageRank, decreasing
citations_ref <- citations_ref[with(citations_ref,
    order(-PageRank)), ]
# Add to list of highes in degree papers
# (references)
top_ref <- rbind(top_ref, head(citations_ref,
    25))
# Remove duplicates
top_ref <- top_ref[!duplicated(top_ref[,
    "FullReference"]), ]
```

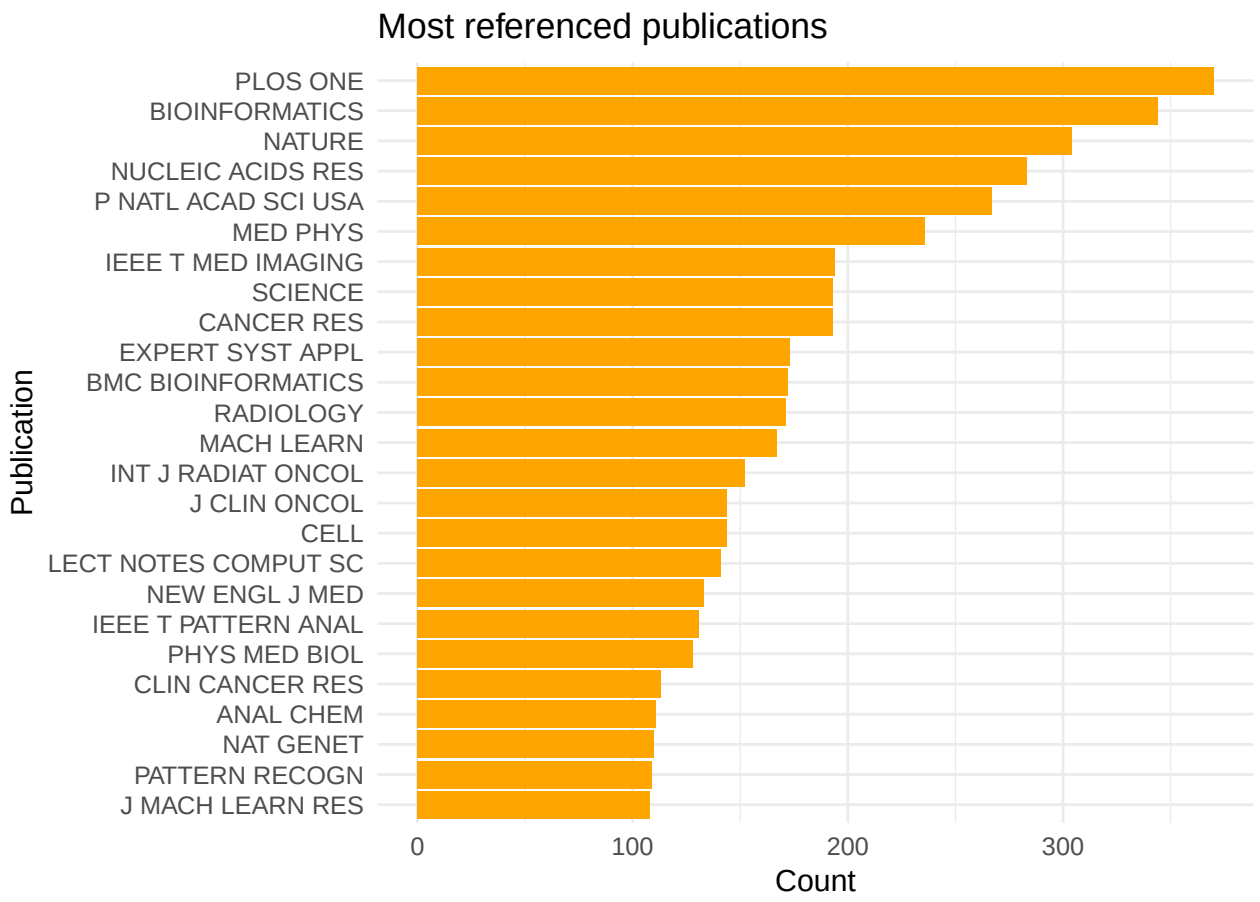
```
# Sort by InDegree, break ties by
# PageRank
top_ref <- top_ref[with(top_ref, order(-InDegree,
    -PageRank)), ]
# Print results
knitr::kable(top_ref[, c("FullReference",
    "InDegree", "PageRank")])
```

	FullReference
1461	BREIMAN L, 2001, MACH LEARN, V45, P5, DOI 10.1023/A:1010933404324
214	CHIH-CHUNG CHANG, 2011, ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY, V2
291	CORTES C, 1995, MACH LEARN, V20, P273, DOI 10.1023/A:1022627411411
2756	2
457	HARALICK RM, 1973, IEEE T SYST MAN CYB, VSMC3, P610, DOI 10.1109/TSMC.1973.4309314
3353	GOLUB TR, 1999, SCIENCE, V286, P531, DOI 10.1126/SCIENCE.286.5439.531
1386	GUYON I, 2002, MACH LEARN, V46, P389, DOI 10.1023/A:1012487302797
1950	LECUN Y, 2015, NATURE, V521, P436, DOI 10.1038/NATURE14539
61	PENG HC, 2005, IEEE T PATTERN ANAL, V27, P1226, DOI 10.1109/TPAMI.2005.159
2435	HALL M., 2009, SIGKDD EXPLORATIONS, V11, P10, DOI 10.1145/1656274.1656278
4406	BREIMAN L, 1996, MACH LEARN, V24, P123, DOI 10.1023/A:1018054314350
3367	SAEYS Y, 2007, BIOINFORMATICS, V23, P2507, DOI 10.1093/BIOINFORMATICS/BTM344
2798	TIBSHIRANI R, 1996, J ROY STAT SOC B MET, V58, P267
129	HANAHAN D, 2011, CELL, V144, P646, DOI 10.1016/J.CELL.2011.02.013
8006	KOUROU K, 2015, COMPUT STRUCT BIOTEC, V13, P8, DOI 10.1016/J.CSBJ.2014.11.005
364	CHAWLA NV, 2002, J ARTIF INTELL RES, V16, P321
900	PEDREGOSA F, 2011, J MACH LEARN RES, V12, P2825
3598	GUYON I., 2003, JOURNAL OF MACHINE LEARNING RESEARCH, V3, P1157, DOI 10.1162/153244303322753
2597	SIEGEL RL, 2015, CA-CANCER J CLIN, V65, P5, DOI 10.3322/CAAC.21254
1949	LECUN Y, 1998, P IEEE, V86, P2278, DOI 10.1109/5.726791
2207	SRIVASTAVA N, 2014, J MACH LEARN RES, V15, P1929
882	HINTON GE, 2006, SCIENCE, V313, P504, DOI 10.1126/SCIENCE.1127647
2668	OJALA T, 2002, IEEE T PATTERN ANAL, V24, P971, DOI 10.1109/TPAMI.2002.1017623
3841	JEMAL A, 2011, CA-CANCER J CLIN, V61, P2011, DOI 10.3322/CAAC.20107
2891	TORRE LA, 2015, CA-CANCER J CLIN, V65, P87, DOI 10.3322/CAAC.21262
1498	VAPNIK V.N., 1998, STAT LEARNING THEORY
1659	HUANG ZW, 2003, INT J CANCER, V107, P1047, DOI 10.1002/IJC.11500
1997	GURCAN M. N., 2009, BIOMEDICAL ENG IEEE, V2, P147, DOI 10.1109/RBME.2009.2034865
1212	ARMATO SG, 2011, MED PHYS, V38, P915, DOI 10.1118/1.3528204
9701	FENG SY, 2011, SCI CHINA LIFE SCI, V54, P828, DOI 10.1007/S11427-011-4212-8

Most referenced publications

```
references <- unlist(strsplit(literature$CitedReferences,
    ";"))

get_publication <- function(x) {
    publication <- "Not found"
    try(publication <- unlist(strsplit(x,
        ","))[[3]], silent = TRUE)
    return(publication)
```

```
}

refPublications <- sapply(references, get_publication)
refPublications <- sapply(refPublications,
    trim)
refPublications <- refPublications[refPublications !=
    "Not found"]
refPublications <- as.data.frame(table(refPublications))
names(refPublications) <- c("Publication",
    "Count")
refPublications <- refPublications[with(refPublications,
    order(-Count)), ]

refPublications <- transform(refPublications,
    Publication = reorder(Publication, Count))

ggplot(head(refPublications, 25), aes(Publication,
    Count)) + geom_bar(stat = "identity",
    fill = "orange") + coord_flip() + theme(legend.position = "none") +
    ggtitle("Most referenced publications") +
    xlab("Publication") + ylab("Count")
```



Topic Model

Topic modeling is a type of statistical text mining method for discovering common "topics" that occur in a collection of documents. A topic modeling algorithm essentially looks through the abstracts included in the datasets for clusters of co-occurring of words and groups them together by a process of similarity.

The following columns describe each topic detected using LDA topic modeling by listing the ten most characteristic words in each topic.

You can specify K, the number of topics, when calling *build_topicmodel_from_literature(literature, K)*. If left empty, *stm::searchK* function is used to estimate the number of topics. For performance reasons the search range is between 4 and 12. The number of topics is estimated using the structural topic model library semantic coherence diagnostic values. Raw values are available in output file as kqualityvalues.csv and can be interpreted with stm documentation if necessary (see section 3.4).

The analysis below creates the topic model using the convenience functions and then prints out ten most descriptive words for each discovered topic. See topicmodels documentation on the TopicModel-class on other information and instructions and documentation on *build_topicmodel_from_literature* how to use the rest of the data the convenience function procides.

```
topicmodel <- build_topicmodel_from_literature(literature)

topickeywords <- topicmodels::terms(topicmodel$fit,
    10)
tw <- data.frame(topickeywords)
colnames(tw) <- gsub("X", "Topic ", colnames(tw))
knitr::kable(tw, col.names = colnames(tw))
```

Topic.1	Topic.2	Topic.3	Topic.4	Topic.5
analysi	crowdsourc	cancer	imag	model
patient	data	gene	cancer	method
studi	inform	predict	breast	data
compon	research	cell	detect	featur
sampl	studi	identifi	method	learn
princip	collect	express	system	algorithm
cancer	task	analysi	base	classif
risk	design	tumor	featur	machin
valid	system	studi	segment	perform
signific	provid	treatment	result	select