Package ‘pubmed.mineR’

July 23, 2019

Type    Package
Title   Text Mining of PubMed Abstracts
Version 1.0.16
Date    2019-07-23
Author  Jyoti Rani, S.Ramachandran, Ab Rauf Shah
Maintainer S. Ramachandran <ramuigib@gmail.com>
Depends R (>= 2.10), methods
Imports RCurl, XML, boot, R2HTML
Collate 'Abstracts-class.R' 'HGNC-class.R' 'Yearwise.R' 'Genewise.R'
   'combineabs.R' 'gene_atomization.R' 'Find_conclusion.R'
   'getabs.R' 'getabsT.R' 'gethgnc.R' 'ready.R' 'readabs.R'
   'removeabs.R' 'searchabsL.R' 'searchabsT.R' 'sendabs.R'
   'subabs.R' 'cleanabs.R' 'word_atomizations.R' 'SentenceToken.R'
   'contextSearch.R' 'uniprotfun.R' 'local_uniprotfun.R'
   'tdm_for_lsa.R' 'printabs.R' 'pubtator_function.R'
   'cos_sim_calc.R' 'cos_sim_calc_boot.R' 'wordscluster.R'
   'whichcluster.R' 'wordsclusterview.R' 'find_intro_conc_html.R'
   'cluster_words.R' 'get_original_term.R' 'get_original_term2.R'
   'input_for_find_intro_conc_html.R' 'xmlreadabs.R'
   'xmlword_atomizations.R' 'xmlgene_atomizations.R'
   'pubtator_result_list_to_table.R' 'genes_BW1.R' 'BW1.R'
   'currentabs_fn.R' 'previousabs_fn.R' 'altnamesfun.R'
   'subsetabs.R' 'prevsymbol_fn.R' 'alias_fn.R' 'get_NMids.R'
   'get_PMIDS.R' 'get_PMCTable.R' 'get_Sequences.R'
   'Give_Sentences_PM.C.R' 'head_abbrev.R' 'names_fn.R'
   'official_fn.R' 'pmids_to_abstracts.R' 'get_gene_sentences.R'
   'Give_Sentences.R' 'get_MedlinePlus.R' 'co_occurrence_fn.R'
   'space_quasher.R' 'readabsnew.R' 'word_associations.R'
   'get.DOIs.R'
License GPL-3
LazyLoad yes
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2019-07-23 07:40:02 UTC

R topics documented:

Abstracts-class ................................. 2
alias_fn ........................................... 3
altnamesfun ................................. 4
BWI ........................................... 5
cleanabs ........................................ 6
cleanabs-methods ............................. 6
cluster_words .................................. 7
combineabs ...................................... 7
combineabs-methods ............................ 8
common_words_new ............................ 9
contextSearch ................................... 9
contextSearch-methods ...................... 10
cos_sim_calc ................................... 10
cos_sim_calc_boot ............................ 11
co_occurrence_fn ............................ 12
currentabs_fn ................................ 13
Find_conclusion ............................. 14
find_intro_conc_html ....................... 14
genes_BWI ..................................... 15
GeneToEntrez .................................. 16
Genewise ....................................... 17
Genewise-methods ............................. 18
gene_atomization ............................. 18
getabs .......................................... 19
getabs-methods ............................... 20
getabsT ......................................... 20
getabsT-methods ............................. 21
get_DOIs ....................................... 21
get_gene_sentences .......................... 22
get_MedlinePlus ............................. 22
get_NMids .................................... 23
get_original_term ............................. 24
get_original_term2 ......................... 25
get_PMCIDs ................................... 25
get_PMCtable ................................ 26
get_Sequences ................................ 26
Give_Sentences .............................. 27
Give_Sentences_PMC ....................... 28
head_abbrev .................................. 29
HGNC-class .................................... 29
topics documented:

HGNC2UniprotID .................................................. 30
HGNCdata ............................................................. 31
input_for_find_intro_conc_html .................................. 32
local_uniprotfun .................................................... 33
names_fn ............................................................. 33
official_fn ........................................................... 34
pmids_to_abstracts ................................................. 35
previousabs_fn ....................................................... 36
prevsymbol_fn ....................................................... 37
printabs ............................................................. 38
pubtator_function ................................................... 38
pubtator_result_list_to_table .................................... 39
readabs .............................................................. 40
readabsnew .......................................................... 41
ready ................................................................. 41
removeabs ............................................................ 42
removeabs-methods .................................................. 43
searchabsL ............................................................ 43
searchabsL-methods .................................................. 44
searchabsT ............................................................ 44
searchabsT-methods .................................................. 46
sendabs .............................................................. 46
sendabs-methods ..................................................... 47
SentenceToken ........................................................ 47
space_quasher ........................................................ 48
subabs ............................................................... 48
subabs-methods ...................................................... 49
subsetabs ............................................................ 50
subsetabs-methods ................................................... 50
tdm_for_lsa ........................................................... 51
uniprotfun ............................................................ 51
whichcluster .......................................................... 52
wordscluster .......................................................... 53
wordsclusterview .................................................... 54
word_associations .................................................... 55
word_atomizations .................................................... 56
xmlgene_atomizations ............................................... 57
xmlreadabs ........................................................... 57
xmlword_atomizations ............................................... 58
Yearwise .............................................................. 59
Yearwise-methods .................................................... 60
Abstracts-class  

Class "Abstracts" Abstract Class

Description
S4 Class with three slots Journal, Abstract, PMID to store abstracts from PubMed

Objects from the Class
Objects can be created by calls of the form new("Abstracts", ...).

Slots
- **Journal**: Object of class "character" to store Journals of the abstracts from PubMed
- **Abstract**: Object of class "character" to store Abstracts from the PubMed
- **PMID**: Object of class "numeric" to store PMIDs of abstracts from PubMed

Methods
No methods defined with class "Abstracts" in the signature.

Author(s)
S. Ramachandran, Ab Rauf Shah

See Also
searchabs, getabs, contextSearch, GeneSearch, Yearwise, combineabs, subabs, subsetabs, readabs

Examples
showClass("Abstracts")

alias_fn

To extract sentences containing Alias of the Human Genes from Pubmed abstracts.

Description
**alias_fn** This function returns the sentences containing alias of gene and the user given terms from the Abstracts using HGNC gee data table. In this sense this function is a 2 Dimensional search.

Usage
alias_fn(genes, data, abs, filename, terms)
Arguments

genes: a table containing genes (official symbol, first column) with its frequency of occurrence (second column) could be an output of gene_atomization function and subsequently subsetting the table using for example the code `genes_table = subset(t2diababs_genes, select = c("Gene_symbol", "Freq"))`. Alternatively, a custom gene table can be supplied with two columns, the first one being the column for Gene symbols and the second one being the Frequency of occurrence. If Frequency of occurrence is not available then a dummy value of 1 can be set.

data: data is HGNC data table with all 49 features (columns) available from the web site https://www.genenames.org/

abs: abs an S4 object of class Abstracts.

filename: filename specifies the name of output file. Please note that the term alias will be suffixed to the given filename.

terms: terms query term(s) to be search in the abstracts, could be a vector of terms.

Value

An output file containing sentences with aliases of genes. For convenience both the official symbol and the corresponding alias are written in the output. The PMID of the corresponding Abstract containing the extracted sentence also appears just before the sentence. Note that multiple sentences from different abstracts are clubbed together under one gene alias that appears in those sentences.

Author(s)

S. Ramachandran

See Also

`prevsymbol_fn`

Examples

```r
## Not run: alias_fn(genes, data, myabs, "nephro_", c("diabetic nephropathy", "kidney disease"))

## genes output of gene_atomization()
```

---

**To Get Alternative names of Genes**

Description

This function is used to retrieve the Alternative names of genes from UniProt using HGNC gene symbol.
**Usage**

`altnamesfun(m)`

**Arguments**

- `m` is a character vector of HGNC official gene symbols.

**Value**

It returns a list of alternative names of given Gene symbols.

**Author(s)**

S. Ramachandran

**References**


**See Also**

`uniprotfun`, `~`~`

**Examples**

```r
## Not run: test = altnamesfun(c("ADIPOQ","BDNF"))
## here "ADIPOQ" is the HGNC gene symbol for which alternative name(s) is required.
```

---

**BWI**

*To obtain the Buzz Word Index of terms from the Abstracts.*

**Description**

This function is used to obtain the Buzz word index value for the terms.

**Usage**

`BWI(current, previous, n, N)`

**Arguments**

- `current` current an S4 object containing the Abstracts for the current year we require the BWI an output from currentabs_fn()
- `previous` previous an S4 object containing the Abstracts for years previous to current year of study an output from previousabs_fn()
- `n` `n` is a character term for which Buzz Word Index is to be calculated.
- `N` `N` is a character value specifying the theme from the large corpus.
Value

It returns a list containing BWI value for the given word.

Author(s)

S. Ramachandran

References


See Also

genes_BWI

Examples

## Not run: result = BWI(mycurrentabs, mypreviousabs, "insulin", "inflammation")
## BWI for the term "insulin" and the theme is inflammation.
## Note that in the previous, years are starting one before the current year 2015;
## current is an S4 object containing the output from currentabs_fn()
## previous is an S4 object containing the output from previousabs_fn().
## 'n' and 'N' are query terms whose BWI is sought and the theme respectively

---

cleanabs

To clean the result of searchabsL

description

It will remove the 'NONE' abstracts from the result of searchabsL..

Usage

cleanabs(object)

Arguments

object an S4 object of class Abstracts.

Value

an S4 object of class Abstracts.

Author(s)

Jyoti Rani
See Also
searchabsL

Examples
## Not run: test1 = searchabsL(abs, include=c("term1", "term2"));
test2 = cleanabs(test1)
## End(Not run)
## here 'abs' is an S4 object of class Abstracts
## 'term1', 'term2' are the searchterms
## test1 is an S4 object containing abstracts for given terms
## and test2 is an S4 object of class Abstracts containing clean abstracts of searchabsL

cleanabs-methods 

Methods for Function cleanabs

Description
To clean 'NONE' part of searchabsL output.

Methods
signature(object = "Abstracts") From an S4 object of class 'Abstracts' the cleanabs function is able to clean the output of searchabsL by removing the 'NONE' part of resultant abstracts.

cluster_words

To Find the highest frequency of words within clusters

Description
Function for finding the word (term) of highest frequency within clusters.

Usage
cluster_words(wordscluster, n)

Arguments
  wordscluster an R object containing the output of wordscluster()
  n a numeric vector containing cluster numbers

Value
a list containing cluster and its highest frequency word
**combineabs**

**Author(s)**
S. Ramachandran

**See Also**
wordscluster

**Examples**
```r
## Not run: test = cluster_words(wordscluster, 5)
## wordscluster is an R object of wordscluster
## 5 is number of cluster
## End(Not run)
```

### Description

`combineabs` will automatically combine two abstracts of two objects.

### Usage

```r
combineabs(object1, object2)
```

### Arguments

- `object1`: An S4 object of class Abstracts
- `object2`: An S4 object of class Abstracts

### Details

Two objects of class 'Abstracts' are combined to return non-redundant combined abstracts. It can be used sequentially to combine many objects of class 'Abstracts'. It will also write the number of combined abstracts into a text file named "data_out.txt"

### Value

An R object containing the combined abstracts, and a text file named "data_out.txt" containing the number of abstracts combined together

**Author(s)**

S.Ramachandran, Jyoti Rani

### Examples

```r
## Not run: res1 = combineabs(x,y)
## here 'x', 'y' are the S4 objects of class 'Abstracts'.
```
Method to Combine Abstracts

Description

combineabs method to combine the abstracts. object1 and object2 are from Abstracts class.

Methods

signature(object1 = "Abstracts") An S4 object of class "Abstracts"
signature(object2 = "Abstracts") An S4 object of class "Abstracts"

common_words_new R Data containing words which frequently in text

Description

This dataset is used to remove common words from the abstracts. This step is used for size reduction for further data mining.

Usage

data(common_words_new)

Format

The format is: chr "common_words_new"

Details

The dataset containing common words used to remove them from the text for size reduction.

References

https://en.wikipedia.org/wiki/Most_common_words_in_English

Examples

data(common_words_new)
contextSearch

For Context Search

Description

contextSearch is a method to extract the sentences containing a given query term

Usage

contextSearch(object, y)

Arguments

object  An S4 object of Class Abstracts containing text abstracts
y  a character vector of term(s)

Details

It takes object of class Abstracts and query term(s) as arguments and returns a text and latex file of the sentences containing query term. The latex file can be further converted into PDF by using the system command in R i.e. system("pdflatex filename.tex"). pdflatex is a shell command in Linux to convert the latex file into PDF. In the pdf file the terms are written in bold face type to enable ease of reading

Value

contextSearch() will write two files one is a text file named "companion.txt", and other is a Latex file. If the single term is given in query then file name comes with the term name. If multiple terms are used then the file name will be "combined.tex"

Author(s)

Dr.S.Ramachandran, Jyoti Rani

Examples

## Not run: contextSearch(x, "diabetes")
## here 'x' is S4 object of class 'Abstracts', and query term is 'diabetes'.
contextSearch-methods

Method for Context Search

Description

contextSearch will search the sentence for the given term(s).

Methods

signature(object = "Abstracts") The object from where it will search should be an S4 object of class Abstracts

cos_sim_calc

To calculate the cosine similarity between terms.

Description

cos_sim_calc calculates the cosine measure of similarity between pairs of terms from a corpus.

Usage

cos_sim_calc(nummatrix)

Arguments

nummatrix A numerical matrix for e.g. a Term Document matrix (output from tdm_for_lsa)

Details

The term document matrix is taken as input and cosine measures of similarity between all pairs of terms are calculated.

Value

A tab delimited text file containing the similarity values between all pairs of terms.

Note

This file can be input to cytoscape directly.

Author(s)

S. Ramachandran
**cos_sim_calc_boot**

References

https://en.wikipedia.org/wiki/Cosine_similarity

See Also

tdm_for_lsa

Examples

```r
## Not run: x = cos_sim_calc(nummatrix)
## here nummatrix is the 'Term Document Matrix' generated from tdm_for_lsa()
```

---

**cos_sim_calc_boot**  *Cosine Similarity Calculation by Boot Strapping*

Description

cos_sim_calc_boot allows boot strap analysis. This function should be used as argument for 'statistic' in the boot function of 'boot' package.

Usage

`cos_sim_calc_boot(data, indices)`

Arguments

data  
Term Document Matrix generated from tdm_for_lsa function of this package. In this matrix, rows are terms and columns are abstracts.

indices  
index of matrix.

Details

while calling this function we need to transpose the input tdm and can also set the number of replicates. boot package is required to call this function.

Value

It will return a matrix containing the cosine similarity of pairs of terms in the abstracts. This object is in same format as returned by the 'boot' function of 'boot' package.

Author(s)

Dr.S.Ramachandran

See Also

tdm_for_lsa
co_occurrence_fn

**Examples**

```r
## Not run: test_boot = boot(data = t(nummatrix), statistic = cos_sim_calc_boot, R = 2)
## here 'nummatrix' is a Term Document Matrix, boot inbuilt function of boot package,
## R is number of replicates here it is 2. User can extend this number.
```

---

**co_occurrence_fn**  
Extracts sentences with co-occurrence of two sets of terms

**Description**

`co_occurrence_fn` will automatically extract sentences with co-occurrence of two sets of terms.

**Usage**

```r
co_occurrence_fn(terms1, abs, filename, terms2)
```

**Arguments**

- `terms1`: a character vector of terms.
- `abs`: an S4 object of class Abstracts
- `filename`: a single character, filename
- `terms2`: a character vector of terms.

**Details**

Sentences with co-occurrence of two terms will be extracted along with the corresponding PMIDs. The data will be written in a text file with the user given filename and the word co_occurrence will be suffixed to it.

**Value**

A text file.

**Author(s)**

S. Ramachandran

**Examples**

```r
## Not run: co_occurrence_fn("resistance",myabs,"resistance_genetic",c("genes","genetic")
##
```
Description

This function is used to extract the abstracts for year we want to study. Its output is used as input in other functions like BWI() and genes_BWI().

Usage

currentabs_fn(yr_to_include, theme, parentabs)

Arguments

yr_to_include
  yr_to_include is the year for which we want to extract the Abstracts.
theme
  theme is a character value specifying the themes for the Abstracts.
parentabs
  parentabs an S4 object containing the Abstracts.

Value

It returns an S4 object containing the abstracts of the given year.

Author(s)

S.Ramachandran

See Also

previousabs_fn

Examples

## Not run: test = currentabs_fn("2015", "atherosclerosis", diabetesabs)
## here "2015" is the year for which, we wish to extract the abstracts on theme*"Atherosclerosis*"
## from the large corpus of diabetes i.e. diabetesabs.
Find_conclusion

To find the conclusion from the abstract(s).

Description

This function is designed for the user convinience, so that user can get the conclusion from the abstract(s) without reading the whole abstract(s).

Usage

Find_conclusion(y)

Arguments

y An S4 object of class 'Abstract'.

Value

A list containing conclusions of given abstract(s)

Author(s)

S.Ramachandran, Jyoti Rani

Examples

## Not run: res1 = Find_conclusion(y)
## here 'y' is an S4 object of class Abstract.

find_intro_conc_html

To find the introduction and conclusion from the abstracts.

Description

it helps to fetch the introduction and conclusion part from the abstracts.

Usage

find_intro_conc_html(y, themes, all)

Arguments

y and S4 object of class Abstracts
themes a character vector containing terms to be search in the abstracts
all is logical, if true, will include title and author otherwise only abstracts will be considered.
genes_BWI

Details

find_intro_conc_html provides an HTML file containing space separated introduction and conclusion part from the abstracts of given query term as well as gives a link directly to PubMed for the resulting PMID.

Value

an HTML file.

Author(s)

S. Ramachandran, Jyoti Rani

See Also

input_for_find_intro_conc_html

Examples

## Not run: test = find_intro_conc_html(abs, "diet", all=FALSE)
## here 'abs' is an S4 object of class Abstracts
## and 'diet' is a term to be search from the abstracts
## this function works for small size of corpus, say about 30-40 abstracts

---

genes_BWI

*Function to obtain the Buzz Word Index of Genes from the abstracts.*

Description

This function provides the Buzz word index for each gene. The theme is the context in which the gene is studied for e.g. atherosclerosis. Using this function user can identify abstracts with emphasis on a given gene.

Usage

genes_BWI(currentabs, previousabs, theme, genes)

Arguments

currentabs currentabs an S4 object containing the Abstracts for the year we want to study. Output from currentabs_fn().

previousabs previousabs an S4 object containing the Abstracts for years previous than our year of study. Output from previousabs_fn().

theme theme a character value to categorize our search. For e.g. 'Atherosclerosis' from 'diabetes' Abstracts.

genes genes list of genes.
Value

It returns a dataframe containing Genes with their corresponding BWI values.

Author(s)

S. Ramachandran

See Also

BWI

Examples

```r
## Not run: test = genes_BWI(currentabs, previousabs, theme, genes)
## currentabs is an S4 object containing the Abstracts for the year we want to study.
## previousabs is an S4 object containing the Abstracts for the years previous
## than our query year for e.g. before 2015
## theme is a character value specifying the search.
## genes is a character vector of gene symbols.
```

---

**GeneToEntrez**  
*Data containing Entrez Ids*

Description

This dataset is used in `DAVID_info` function of the package, and it contains the Entrez Ids for the respective genes and these Entrez Ids will be used to get information about human genes.

Usage

```r
data(GeneToEntrez)
```

Format

The format is: `chr "GeneToEntrez"

Examples

```r
data(GeneToEntrez)
```
Description

Genewise reports the number of abstracts for given gene(s) name(s)

Usage

Genewise(object, gene)

Arguments

object An S4 object of class Abstracts
gene a character input of gene name (HGNC approved symbol)

Details

This function will report the number of abstracts containing the query gene term(s) [HGNC approved symbols], and the result is saved in a text file "dataout.txt". Genewise() will report numbers of abstracts only. The abstracts themselves for corresponding gene names can be obtained using searchabsL() and searchabsT.

Value

Genewise will return an R object containing the abstracts for given gene, and a text file named "dataout.txt" containing the number of abstracts

Author(s)

S. Ramachandran, Jyoti Rani

Examples

```r
## Not run: Genewise(x, "TLR4")
## here 'x' contains the S4 object of Abstracts.
```
Genewise-methods  method to find the abstracts for the given gene.

Description

Genewise The method Genewise will automatically report the numbers of abstracts for a given gene. It will write the result in the text file named "dataout.txt"

Methods

signature(object = "Abstracts") This method will search in an S4 object, containing abstracts. It will write a text file named "dataout.txt", containing the number of abstracts for the query gene terms

gene_atomization  To Extract Genes from the Abstracts

Description

gene_atomization will automatically fetch the genes (HGNC approved Symbol) from the text and report their frequencies. presently only HGNC approved symbols are used.

Usage

gene_atomization(m)

Arguments

m          An S4 object of class Abstracts

Details

The function writes a text file with file name "data_table.txt". The function gene_atomization() is used to obtain the name of genes along with their frequencies of occurence.

Value

A tab delimited table containing gene name and their frequencies of occurrence.

Author(s)

S.Ramachandran, Jyoti Rani
Examples

```r
## Not run: getabs(x, "term")
## x is an S4 object of class Abstracts containing the abstracts
```
getabs-methods  getabs *To Get abstracts for a term*

**Description**

getabs will search for the abstracts of a given term. It is case sensitive.

**Methods**

```signature(object = "Abstracts")```

This method takes three arguments, first `object` containing data to be search, `x`, the term to be search, `y` is logical if set "YES" will consider the case of text.

getabsT  *To get Abstracts for a given term.*

**Description**

getabsT will automatically fetch the abstracts containing the query term.

**Usage**

getabsT(object, x, y)

**Arguments**

- **object**  An S4 object of class Abstracts
- **x**  A character string for the term
- **y**  is logical, if set TRUE, search will be case sensitive.

**Details**

getabsT() is similar to getabs(), but it performs more specific search.

**Value**

An object of class 'Abstracts', containing the resulted abstracts for term.

**Author(s)**

S.Ramachandran

**Examples**

```## Not run: getabsT(diabdata, "term")```
Description

getabsT will automatically return the abstracts of a term from the data.

Methods

signature(object = "Abstracts")  getabsT will search for the abstracts of a term in the data, and will automatically write the number of abstracts into a text file named "dataout.txt".

get_DOIs  

function for extracting Digital Object Identifier (DOIs) of papers

Description

get_DOIs is used to extract DOIs of papers.

Usage

get_DOIs(abs)

Arguments

abs  An S4 object of class Abstracts

Details

get_DOIs allow users to get DOIs for individual papers.

Value

It returns a list object containing DOIs. This is useful for further extraction of papers

Author(s)

S.Ramachandran

Examples

## Not run: test = get_DOIs(vitiligoabs)
##
get_gene_sentences To extract the sentences for genes from the corpus.

Description
get_gene_sentences is used to extract the exact sentence in which query gene is discussed.

Usage
get_gene_sentences(genes, abs, filename)

Arguments
genes a character vector containing the gene symbols.
abs an S4 object of class Abstracts
filename specifies the output file name.

Value
an output file containing the sentences for given gene.

Author(s)
S.Ramachandran

Examples
## Not run: get_gene_sentences("RBP4", abstracts, "RBP4_sentence.txt")

get_MedlinePlus To Get MedLinePlus Summary

Description
This function is to get the summary from MedLinePlus.

Usage
get_MedlinePlus(x)

Arguments
x is a character input of terms: for examples 'malaria', 'pneumonia', 'chronic diseases'
Value
It returns a HTML file with name result_Medline_plus.html to be opened with any browser

Author(s)
S.Ramachandran

References

Examples
```r
## Not run: get_MedlinePlus("malaria")
```

---

**get_NMids**

To extract NM ids from NCBI.

Description
get_NMids is to fetch the NM ids from the NCBI for corresponding gene/s to further fetch the sequence of that gene/s.

Usage
```
get_NMids(x)
```

Arguments
- `x` an R object containing Locus IDs for genes from NCBI2R package.

Value
It returns a list object containing corresponding NM id from NCBI.

Author(s)
S.Ramachandran

References

See Also
get_Sequences
**get_original_term**

## Examples

```r
## Not run: getNMids("5950")
## 5950 is Locus id of RBP4 gene.
```

---

**get_original_term**  To get the original terms from the corpus. deprecated

---

### Description

get_original_term is used to get the exact term as it is present in corpus. This function is not recommended anymore.

### Usage

```r
get_original_term(m, n)
```

### Arguments

- **m**: an S4 object of class Abstracts containing the corpus.
- **n**: a list object output from the function cluster_words

### Value

a list object containing the terms.

### Author(s)

S.Ramachandran, Jyoti Rani

### See Also

wordscluster

### Examples

```r
## Not run: test = get_original_term(abs, words)
## here abs is an S4 object of class Abstracts
## words is the output object of cluster_words()
```
**get_original_term2**  
*To get the original terms from the corpus.*

**Description**

`get_original_term2` is used to get the exact term as it is present in corpus. It takes one term at a time. For multiple terms we can use `lapply`.

**Usage**

```r
get_original_term2(x, y)
```

**Arguments**

- `x`  
  - `x` is a character value specifying the query term.
- `y`  
  - `y` is an S4 object containing abstracts.

**Value**

It returns a list object containing accurate term.

**Author(s)**

Jyoti Rani, S.Ramachandran.

**See Also**

`get_original_term`

**Examples**

```r
## Not run: test = get_original_term("hbalc", diababs)
## here it will return accurate formation of hbalc i.e. HbAlc from diababs.
```

---

**get_PMCIDS**  
*To extract the PMC Ids of the abstracts.*

**Description**

`get_PMCIDS` is used to fetch the PMC Ids of the abstracts from the corpus.

**Usage**

```r
get_PMCIDS(abs)
```
get_PMCTable

Arguments
abs absan S4 object of class Abstracts.

Value
It returns a list containing PMC Ids.

Author(s)
S.Ramachandran

Examples
## Not run: get_PMCTable(abstracts)

get_PMCTable is used to extract the full text article by giving query PMC Id. Deprecated.

Usage
get_PMCTable(url)

Arguments
url url is url of query PMC Id.

Value
It will return a full text article.

Author(s)
S.Ramachandran

References

See Also
get_PMCTable

Examples
\textbf{get\_Sequences} \hspace{1cm} \textit{To extract the Gene sequence from the NCBI.}

**Description**

\texttt{get\_Sequences} is used to fetch the sequences of genes using NM ids.

**Usage**

\begin{verbatim}
get\_Sequences(x, filename)
\end{verbatim}

**Arguments**

\begin{itemize}
  \item \texttt{x} \hspace{1cm} NM Id of the sequence.
  \item \texttt{filename} \hspace{1cm} filename specifies the name of output file.
\end{itemize}

**Value**

It will return a text file containing sequence.

**Author(s)**

S. Ramachandran

**See Also**

\texttt{get\_NMids}, \texttt{\ldots}

**Examples**

\begin{verbatim}
## Not run: get\_Sequences("NM\_012238.4", "SIRT1")
\end{verbatim}

\textbf{Give\_Sentences} \hspace{1cm} \textit{To extract sentences from the Abstracts}

**Description**

\texttt{Give\_Sentences} will help to extract the sentence containing query term/s from the large corpus.

**Usage**

\begin{verbatim}
Give\_Sentences(m, abs)
\end{verbatim}

**Arguments**

\begin{itemize}
  \item \texttt{m} \hspace{1cm} m a character term.
  \item \texttt{abs} \hspace{1cm} abs an S4 object of class Abstracts.
\end{itemize}
Value

It will return a list object containing sentences

Author(s)

S.Ramachandran

See Also

Give_Sentences_PMC

Examples

## Not run: Give_Sentences_PMC(PMC4039032, "atherosclerosis")
head_abbrev

To extract the abbreviated term.

Description

head_abbrev is used to find expansion for which abbreviation is used. It will help to find the falsely matching abbreviations from the abstracts.

Usage

head_abbrev(limits, term, pmid, abs)

Arguments

limits specifies the limit up to which expansion should be displayed. Default is 50

term is the query term (abbreviation)

pmid describes the PMID

abs an S4 object of class Abstracts.

Value

It will return a list.

Author(s)

S.Ramachandran

Examples

## Not run: head_abbrev(50, "AR", "16893912", myabs)

HGNC-class

HGNC Class for package.

Description

"HGNC"

Objects from the Class

Objects can be created by calls of the form new("HGNC", ...).
Slots

HGNCID: Object of class "character"
ApprovedSymbol: Object of class "character"
ApprovedName: Object of class "character"
Status: Object of class "character"
PreviousSymbols: Object of class "character"
Aliases: Object of class "character"
Chromosome: Object of class "character"
AccessionNumbers: Object of class "character"
RefSeqIDs: Object of class "character"

Author(s)

Dr.S.Ramachandran, Ab Rauf Shah

See Also

Abstracts

Examples

showClass("HGNC")

HGNC2UniprotID  R Data containing HGNC2UniprotID data mapping.

Description

This dataset contains HGNC2UniprotID from Uniprot and is used in uniprotfn() function of this package, to get the information of a gene from the Uniprot.

Usage

data(HGNC2UniprotID)

Format

The format is: chr "HGNC2UniprotID"

Details

The dataset contains HGNC2UniprotID
HGNCdata

References


Examples

data(HGNC2UniprotID)

HGNCdata

R Data containing HGNC data.

Description

This dataset contains data from Human Gene Nomenclature Committee i.e HGNC ID, HGNC approved symbol, approved name, gene synonyms, chromosome no., accession numbers and RefSeq ids.

Usage

data(HGNCdata)

Format

The format is: chr "HGNCdata"

Details

The dataset contains HGNCdata

References


Examples

data(HGNCdata)
input_for_find_intro_conc_html

fetch the abstracts using E-utilities.

Description

it helps in searching and fetching the abstracts from E-utilities using PMIDs.

Usage

input_for_find_intro_conc_html(y, all)

Arguments

y
  an S4 object of class Abstracts

all
  is logical if true, will include title and author also.

Details

it takes an S4 object as input and uses its PMIDs to fetch the abstracts from E-utilities. The output will be used as input for find_intro_conc_html as it contains neat data i.e. abstracts only.

Value

a list containing abstracts and PMID

Author(s)

S. Ramachandran, Jyoti Rani

References


See Also

find_intro_conc_html

Examples

## Not run: test=input_for_find_intro_conc_html(abs)
## here 'abs' is an S4 object of class Abstracts.
\textit{local\_uniprotfun} \hspace{1cm} \textit{To Get Information from Uniprot.}

\section*{Description}
It is an auxiliary function for \textit{altnamesfun}.

\section*{Usage}
\begin{verbatim}
local\_uniprotfun(y)
\end{verbatim}

\subsection*{Arguments}
\begin{itemize}
\item \texttt{y} \hspace{1cm} a character value containing HGNC Gene symbol
\end{itemize}

\section*{Value}
It writes an output file named "x.txt" which will be used as input in \textit{altnamesfun}().

\section*{Author(s)}
S.Ramachandran, Jyoti Rani

\section*{See Also}
uniprotfun

\section*{Examples}
\begin{verbatim}
## Not run: local\_uniprotfun("TLR4")
## here it will generate an output file named "x.txt" containing
## result for TLR4.
\end{verbatim}

\section*{names\_fn} \hspace{1cm} \textit{To extract the sentences in abstracts containing gene names from HGNC.}

\section*{Description}
\texttt{names\_fn} matches the gene symbols to gene names and extract from HGNC.

\section*{Usage}
\begin{verbatim}
names\_fn(genes, data, abs, filename, terms)
\end{verbatim}
Arguments

- genes: genes is output of gene_atomization or a table containing HGNC gene symbols in first column with its frequency in second column.
- data: data is HGNC data table with all 49 features (columns) available from the website https://www.genenames.org/
- abs: abs an S4 object of class Abstracts.
- filename: filename specifies the name of output file.
- terms: terms second query term to be searched in the same sentence (co-occurrence) of abstracts.

Value

It returns an output file containing genes with their corresponding gene names and sentences with co-occurrences if any.

Author(s)

S.Ramachandran

Examples

```r
## Not run:
names_fn(genes, data, diabetes_abs, "names", c("diabetic nephropathy", "DN"))
```

Usage

```r
official_fn(genes, abs, filename, terms)
```

Arguments

- genes: genes is output of gene_atomization, or a table containing HGNC gene symbols in first column with its frequency in second column.
- abs: abs an S4 object of class Abstracts.
- filename: filename specifies the name of output file.
- terms: terms second query term to be searched in the same sentence (co-occurrence) of abstracts.
**pmids_to_abstracts**

**Value**

It will return a text file containing corresponding official gene symbol.

**Author(s)**

S.Ramachandran

**Examples**

```r
## Not run: official_fn(genes, diabetes_abs, "genes", c("diabetic nephropathy", "DN"))
## End(Not run)
## genes output of gene_atomization()
```

---

**pmids_to_abstracts** To Find and match the PMID's to the abstracts.

**Description**

`pmids_to_abstracts` is used to extract the abstract/s of query PMID/s.

**Usage**

`pmids_to_abstracts(x, abs)`

**Arguments**

- `x` : a numeric vector containing PMIDs
- `abs` : abs an S4 object of class Abstracts.

**Value**

It will return an S4 object of class abstracts containing abstracts for query PMIDs.

**Author(s)**

S.Ramachandran

**Examples**

```r
## Not run: pmids_to_abstracts(26878666, abs)
```
previousabs_fn To Retrieve the Abstracts from the large corpus for given years.

Description

This function is used to extract the abstracts from the large corpus excluding the years and under a given theme. Its output is used in other functions like BWI and genes_BWI

Usage

previousabs_fn(yrs_to_exclude, theme, parentabs)

Arguments

yrs_to_exclude

yrs_to_exclude is abstracts for the list of years we want to exclude from the corpus

theme

theme is a character value specifying the themes for the Abstracts.

parentabs

parentabs an S4 object containing the Abstracts.

Value

It returns an S4 object containing the abstracts of the given year.

Author(s)

S.Ramachandran

See Also

currentabs_fn

Examples

## Not run: test = previousabs_fn(as.character(2015:2010), "atherosclerosis", diabetesabs
## here we will get the abstracts before 2010 for 'atherosclerosis'
## from the large corpus diabetesabs.
prevsymbol_fn

To extract the sentences containing Previous symbols of HGNC genes.

Description

prevsymbol_fn will return the sentences containing previous symbols of the genes from the abstracts using HGNC data.

Usage

prevsymbol_fn(genes, data, abs, filename, terms)

Arguments

genes genes is output of gene_atomization, or a table containing HGNC gene symbols in first column with its frequency in second column.
data data is HGNC data table with all 49 features (columns) available from the website https://www.genenames.org/
abs abs an S4 object of class Abstracts.
filename filename specify the name of output file
terms terms second query term to be searched in the same sentence (co-occurrence) of abstracts.

Value

It returns a text file containing gene symbol with corresponding previous symbols.

Author(s)

S.Ramachandran

See Also

names_fn, official_fn

Examples

## Not run:
prevsymbol_fn(genes, data, diabetes_abs, "prevsym", c("diabetic nephropathy", "DN"))
## End(Not run)
printabs

To print the total number of abstracts in an S4 object of class Abstracts, its start and end.

Description

It gives overview of the abstracts in an S4 object of class Abstracts.

Usage

printabs(object)

Arguments

object An S4 object of class Abstracts.

Value

prints the total number of abstracts in an S4 object with additional information.

Author(s)

S. Ramachandran

Examples

## Not run: printabs(myabs)
## here myabs is an S4 object of class Abstracts.

pubtator_function

function for text annotation using PubTator

Description

pubtator_function is used to extract specific information from an abstract like Gene, chemical, and diseases etc.

Usage

pubtator_function(x)

Arguments

x numeric value PMID.
pubtator_result_list_to_table

Details

`pubtator_function` allow users to get information about 'Gene', 'Chemical' and 'Disease' for given PMID. It uses online tool PubTator on R platform. It also removes redundancy from the output. It takes one PMID at once, for multiple PMIDs user can use `lapply()` function.

Value

It returns a list object containing Gene, Chemical, Disease and PMID. The corresponding concept id numbers are joined by a ‘>’ character. This is useful for further data mining

Author(s)

S.Ramachandran, Jyoti Rani

References

Wei CH et. al., Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts, Database (Oxford), bas041, 2012

Examples

```r
## Not run: test = pubtator_function(17922911)
## here pubtator_function() will extract the information from this given pmid.
```

---

pubtator_result_list_to_table

*Function to Convert Pubtator result from list into Table*

Description

This function is used to collect the outputs of `pubtator_function()` after using `lapply` over multiple PMIDs. This function enables to convert it into table for easy reading and further analysis.

Usage

`pubtator_result_list_to_table(x)`

Arguments

`x` here `x` is list output of `pubtator_function()`. 
Value

It returns table for pubtator_function output.

Author(s)

S.Ramachandran, Jyoti Rani

See Also

pubtator_function

Examples

## Not run: test = pubtator_result_list_to_table(x)
##here x is the output of pubtator_function

Description

readabs will automatically read the abstracts from the pubmed file.

Usage

readabs(x)

Arguments

x Text file of PubMed abstracts. (Abstracts downloaded from PubMed)

Details

The saved file from a general pubmed search as text file is read via readabs().

Value

An S4 object of class "Abstracts", and a text file with tab delimited headers Journal, Abstract, PMID written with file name "newabs.txt".

Author(s)

S.Ramachandran

Examples

## Not run: readabs("pubmed_result.txt")
##here pubmed_result.txt is the text file of abstracts saved from PubMed.
readabsnew

To read Abstracts

Description

readabsnew will automatically read the abstracts from the pubmed text file.

Usage

readabsnew(x)

Arguments

x Text file of PubMed abstracts. (Abstracts downloaded from PubMed)

Details

The saved file from a general pubmed search as text file is read via readabsnew().

Value

An S4 object of class "Abstracts" and a text file with tab delimited headers Journal, Abstract, PMID written with file name "newabs.txt".

Author(s)

S.Ramachandran

Examples

## Not run: readabsnew("pubmed_result.txt")
##here pubmed_result.txt is the text file of abstracts saved from PubMed.

ready

To Initiate the Classes.

Description

ready will initiate the classes neccessary for other functions.

Usage

ready()

Details

This function is neccessary to initiate the classes which are needed for the implementation of other functions.
Value
  
  classes

Author(s)
  
  S. Ramachandran

Examples

## Not run: ready()
removeabs-methods

To remove abstracts of a term from the data.

Description
removeabs  This function will search for the abstracts containing the given term to remove them from the data.

Methods
signature(object = "Abstracts")  This method depicts its function, it will remove the abstracts from the data, and the number of abstracts removed will be written the text file named "dataout.txt"

searchabsL  To Search the abstracts of term(s) in a combination mode.

Description
searchabsL will search for abstracts for the given term(s). Multiple combinations are allowed.

Usage
searchabsL(object, yr, include, restrict, exclude)

Arguments
object  An S4 object of class Abstracts
yr  character vector specifies the year of search
include  character vector specifies the terms contained in the abstracts.
restrict  character vector specifies the term contained in the abstracts for which search should be restricted.
exclude  character vector specifies the terms contained in the abstracts for excluding these abstracts from the search results.

Details
In the arguments except for the object all other arguments have "NONE" as default. To export or write the result of searchabsL() we use sendabs() function.

Value
An object of class Abstracts satisfying the term combinations, In addition a text file named "out.txt" reporting the number of abstracts for given query term combinations.
Author(s)
S. Ramachandran

See Also
searchabsT

Examples

```r
## Not run: searchabsL(myabs, include="term")
searchabsL(myabs, yr="2013")
searchabsL(myabs, restrict="term")
searchabsL(myabs, exclude="term")
searchabsL(myabs, include="term", exclude="term2")
## End(Not run)
## Here myabs is the object of class Abstracts containing data,
## "term" is the query term to be search.
```

searchabsL-methods Searching Abstracts

Description
searchabsL will automatically search the abstracts from the data for the given terms or their combination of several terms.

Methods

```r
signature(object = "Abstracts")
```

searchabsL will search the abstracts for the given term or combinations of several terms. In this method the argument "include" uses the boolean operator 'OR' and is liberal whereas the 'restrict' and 'exclude' use the boolean operator 'AND' to specify additional filters. If the restriction to individual terms are desired then they can be individually searched and then the multiple abstracts can be combined using combineasb() function.

searchabsT To Search Abstracts

Description
searchabsT is similar to searchabsL() but performs more specific search. It performs case sensitive search.

Usage

```r
searchabsT(object, yr, include, restrict, exclude)
```
Arguments

object  An S4 object of class Abstracts
yr  character vector specifies the year(s) of search.
include  character vector specifies the term(s) for which abstracts to be searched.
restrict  character vector specifies the term(s) contained in the abstracts for which search should be restricted.
exclude  character vector specifies the term(s) contained in the abstracts for excluding these abstracts from our search results.

Details

In the arguments except the object all arguments have "NONE" as default. Use sendabs() function to write the results in a tab delimited text file.

Value

An object of class Abstracts meeting the term and the term combinations. A text file reporting the number of abstracts for the query terms and their combinations is also written with the filename "out.txt".

Author(s)

Dr.S.Ramachandran

See Also

searchabsL

Examples

```r
## Not run: searchabsT(myabs, yr="2013")
searchabsT(myabs, include="term")
searchabsT(myabs, restrict="term")
searchabsT(myabs, exclude="term")
searchabsT(myabs, yr="2013", include="term")
## End(Not run)
## Here myabs is an S4 object of class Abstracts containing the abstracts to search,
## "term" is the query term to be search.
```
searchabsT-methods searchabsT Searching abstracts

Description

searchabsT will perform a specific search for the given term.

Methods

signature(object = "Abstracts") It is similar to the searchabsL method, but it is more specific than searchabsL, it is case sensitive, however searchabsL is not.

sendabs To send abstracts

Description

sendabs will send the abstracts into a tab delimited text file with the fields Journal, Abstract, and PMID.

Usage

sendabs(object, x)

Arguments

object An S4 object of class 'Abstracts'
x "filename.txt" to write the abstracts

Details

A general writing function for object of class 'Abstracts'

Value

A tab delimited text file with headers Journal, Abstract, PMID.

Author(s)

S.Ramachandran, Jyoti Rani

Examples

## Not run: sendabs(myabs,"myabs.txt")
## here myabs is the S4 object of class 'Abstracts' and
## 'abs.txt' is the file where abstracts will be written.
sendabs-methods     To send the Data into a File

Description

sendabs will write the data of an object of class 'Abstracts' into a tab delimited text file with header Journal, Abstract, and PMID.

Methods

signature(object = "Abstracts") sendabs will send the data into a text file. It writes a tab delimited text file for PubMed abstracts containing Journal, Abstract, and PMID.

SentenceToken     To Tokenize the sentences

Description

SentenceToken will tokenize abstracts into individual sentences.

Usage

SentenceToken(x)

Arguments

x is a character string; could be an output from paste

Details

This function is necessary for extracting sentences from abstracts, used by contextSearch function. The tokenization principle follows the overall strategy as described in contextSearch.

Value

A character vector of sentences

Author(s)

S.Ramachandran

Examples

## Not run: SentenceToken(x)
space_quasher  
\textit{Removes extra spaces between words.}

**Description**

\texttt{space\_quasher} will automatically remove extra spaces between words. Therefore only one space between any pair of words will be left.

**Usage**

\texttt{space\_quasher(x)}

**Arguments**

- \texttt{x}  
  
  \texttt{x} is a text with single or multiple sentences given within double quotes.

**Details**

The extra spaces between words in sentences is quashed to one via \texttt{space\_quasher()}. 

**Value**

Sentences(s) in which extra spaces between any pair of words are quashed to one.

**Author(s)**

S.Ramachandran

**Examples**

```r
## Not run: space\_quasher("I am a ghostbuster. I have the tools required to hunt ghosts")
##here pubmed\_result.txt is the text file of abstracts saved from PubMed.
```

---

**subabs**  
\textit{To find sub-abstracts}

**Description**

\texttt{subabs} will automatically extract the sub-abstracts from large set of abstracts.

**Usage**

\texttt{subabs(object, start, end)}
**Arguments**

- **object**: An S4 object of class Abstracts
- **start**: integer, specifies starting limit of the range to perform search
- **end**: integer, specifies end limit of the range to perform search

**Details**

From a large number of abstracts wish to extract a subset of abstracts into a separate object.

**Value**

An R object of class 'Abstracts' containing the extracted abstracts meeting a given range.

**Author(s)**

Jyoti Rani, S.Ramachandran

**Examples**

```r
## Not run: subabs(myabs,1,5)
## Here 'myabs' is an S4 object of class 'Abstracts',
## 1 and 5 are the start and end respectively.
```

---

**subabs-methods**  
*Getting subabstracts*

**Description**

`subabs` subabs will extract the sub abstracts corresponding to a given range, from the whole data.

**Methods**

`signature(object = "Abstracts")` From an S4 object of class 'Abstracts' the subabs function is able to extract the abstracts corresponding to a given range.
subsetabs-methods

To make subsets of large corpus.

Description

It is used to divide the large corpus into a given range.

Usage

subsetabs(object, indices)

Arguments

object object is an S4 object containing Abstracts.
indices indices a numeric range (e.g. 1:10, c(1,5,7,9,10)).

Value

It returns an S4 object of extracted Abstracts.

Author(s)

S. Ramachandran.

Examples

## Not run: test = subsetabs(diabetesabs, 1:50)
## here we want to extract the Abstracts ranges from 1 to 50
## from the large corpus of diabetes.

subsetabs-methods

To make subset of Abstracts.

Description

subsetabs is used to subset of Abstracts from the large corpus. Its output is used in other functions like currentabs_fn and previousabs_fn

Methods

signature(object = "Abstracts") subsetabs will divide the large corpus into subset.
### tdm_for_lsa

create Term Document Matrix for lsa analysis

#### Description

lsa package take "Term Document Matrix" as input, so it is needed to create a 'tdm' for Abstracts and tdm_for_lsa do the same as it find out the frequency of given term in each abstract and each abstract is considered as separate document. It prepares term document matrix of terms in the 'abstracts' corpus.

#### Usage

```
tdm_for_lsa(object, y)
```

#### Arguments

- **object**: An S4 object of class 'Abstracts'
- **y**: a character vector specifying the terms

#### Value

a Term Document Matrix (Numerical matrix) containing the raw frequencies of given terms in each abstract.

#### Author(s)

Jyoti Rani

#### Examples

```
## Not run: y = c("insulin", "inflammation", "obesity")
tdm_for_lsa(myabs,y)
## End(Not run)
```

### uniprotfun

To get information about gene from the UniProt.

#### Description

uniprotfun will access the UniProt data for a given gene as per HGNC approved gene symbols.

#### Usage

```
uniprotfun(y)
```
whichcluster

Arguments

y a HGNC approved gene symbol as character

Details

This function retrieves data from the UniProt. At present uniprotfun() works with only HGNC approved gene symbols.

Value

A text file written with filename as the 'query' name suffixed with .txt

Author(s)

S.Ramachandran

Examples

## Not run: uniprotfun("SIRT1")

whichcluster To fetch the cluster for words

Description

whichcluster is used to get the cluster in which a given word (term) occurs.

Usage

whichcluster(clusterobject, y)

Arguments

clusterobject an R object containing the clusters of words output by wordscluster function.

y a character string of query terms.

Value

a list containing the number of cluster under which given term occurs.

Author(s)

S.Ramachandran

See Also

wordscluster
wordscluster

**Examples**

```r
## Not run: test<-whichcluster(x, "diabetes")
## here x is an R object output form wordscluster function.
## and "diabetes" is the term for which cluster number is to be searched.
## End(Not run)
```

wordscluster To cluster the words

**Description**

`wordscluster` is used to cluster the words, using the levenshtein distance concept, which are coming together in combination with either 'prefixes' or 'suffixes' or other compound words. The first word, usually of lowest length, could be 'stemmed' word in many cases drastically so, is considered as representative for that cluster.

**Usage**

```r
wordscluster(lower, upper)
```

**Arguments**

- `lower`: lower limit for characters in word. Default = 5.
- `upper`: upper limit of characters in word. Default = 30

**Details**

This function is usefull for dampening the 'explosion' of words output from word_atomizations. This step enables easy examination of the terms.

**Value**

A list object of words clustered together and a text filenamed "resulttable.txt" with the columns cluster number, cluster size and representatives of clusters.

**Note**

The function may run faster when the lower limits are reduced but 'risks' producing plenty of 'decoy' situations. Their frequencies are very rare. Decoy situations: Some 'words' with part identity to other smaller words will runaway with smaller words. This event creates an unfavorable situation whereby the generated 'clusters' of words become difficult to interpret. This situation can be minimized by increasing the lower limit of word length, however at the cost of lowering computational speed. An example is: the word hypercholesterolemia runaway with the smaller word 'lester' which could be another name. In this instance increasing the lower limit will be more usefull. Words longer than 30 characters are usually names of chemical compounds in IUPAC system of nomenclature.
Author(s)
S.Ramachandran, Jyoti Rani

See Also
whichcluster word_atomizations

Examples
## Not run:
test=wordscluster(5, 10)
## here it will start making cluster of words of length with minimum of 5 characters
## and maximum of 10 characters.
## End(Not run)

wordsclusterview
To view the words in cluster

Description
wordsclusterview is used to view the words comes in cluster formed by wordscluster function.

Usage
wordsclusterview(words_cluster, all)

Arguments
words_cluster
an R object containing output of wordscluster

all
is logical and default is FALSE, if set to TRUE includes those with one member word.

Details
The first 5 words and 5 words near the median nd 5 words at the tail end are shown for clusters with more than 15 members. In case of cluster size less than 15, all the words are written in output.

Value
It returns a text file named word_cluster_view.txt

Author(s)
S.Ramachandran, Jyoti Rani
word_associations

See Also

wordscluster

Examples

## Not run: test = wordsclusterview(cluster)
# here cluster is output from wordscluster
## End(Not run)

word_associations Extracts the words associated (to the left and to the right) with a given word

Description

word_associations will automatically extract associated words for a given word, namely the words immediately to the left and to the right. The given word is usually in the middle except for those cases, where the given word occurs either at the start or the end of the sentence.

Usage

word_associations(term, abs)

Arguments

term is a single word

abs an S4 object of class Abstracts

Details

Certain words are qualified by authors in various ways. For example, physical therapy, gene therapy etc. This function is useful in extracting these qualified words in the form of available associated words. Useful for preparing terms to be given in co_occurrence_fn(). There could be other uses also.

Value

comp1 A list of all the word pairs in a given set of abstracts.

Author(s)

S. Ramachandran

References

word_atomizations

See Also

Give_Sentences

Examples

```r
## Not run: word_atomizations("therapy",myabs
##
```

---

**word_atomizations**  
Atomization of words

**Description**

`word_atomizations` will automatically break the whole text into words and rank them according to their frequency of occurrence.

**Usage**

`word_atomizations(m)`

**Arguments**

- `m`  
  An S4 object of class Abstracts

**Details**

`word_atomizations()` will break down the whole text into words after removing the extra white space, punctuation marks and very common English words.

**Value**

A text file containing words with their frequencies

**Author(s)**

S. Ramachandran, Jyoti Sharma

**Examples**

```r
## Not run: word_atomizations(myabs)
## here myabs is the object containing abstracts.
```
xmlgene_atomizations

Gene atomization of xml abstracts.

Description

xmlgene_atomizations is used to fetch the list of genes from the xml abstracts.

Usage

xmlgene_atomizations(m)

Arguments

m       an S4 object of class Abstracts, output from xmlreadabs.

Value

a list containing genes from the text with their frequency of occurrence.

Author(s)

S.Ramachandran, Jyoti Sharma

See Also

xmlreadabs

Examples

## Not run: test = xmlgene_atomizations(xmlabs)
## xmlabs is an S4 object of class Abstracts i.e. output of xmlreadabs

xmlreadabs

To read the abstracts from the PubMed saved in XML format.

Description

xmlreadabs is modified form of readabs as it reads the abstracts downloaded/saved in XML format from PubMed. This is helpful to give clean and better result after preprocessing i.e. word_atomizations, wordscluster etc.

Usage

xmlreadabs(file)
xmlword_atomizations

Word atomizations of abstracts from xml format.

Description

xmlword_atomizations is used to process the abstracts from PubMed in XML format.

Usage

xmlword_atomizations(m)

Arguments

m an S4 object of class Abstracts resulted from xmlreadabs.

Value

a list containing words from the text with their frequencies.

Note

xmlword_atomizations cannot work on output of readabs.

Author(s)

S. Ramachandran
Yearwise

See Also

xmlreadabs

Examples

## Not run: test = xmlword_atomizations(xmlabs)
## here xmlabs is an S4 object i.e. output of xmlreadabs

Yearwise

To Search abstracts Year wise

Description

Yearwise reports the no. of abstracts in a year.

Usage

Yearwise(object, year)

Arguments

object An S4 object of class Abstracts.
year a character vector specifies the year.

Details

Yearwise() is useful to find the no. of abstracts for the given year.

Value

A text file containing the no. of abstracts for given Year(s)

Author(s)

Dr.S.Ramachandran

Examples

## Not run: Yearwise(myabs, "2011") or
Yearwise(myabs, c("2011", "2013", "2009")
## End(Not run)
## Here myabs is the object containing PubMed abstracts.
Yearwise-methods  Yearwise *Year wise extraction of Abstracts*

---

**Description**

*Yearwise* will report the abstracts for given year(s).

**Methods**

```r
signature(object = "Abstracts")
```

This method "Yearwise" is written to fetch the abstracts yearly.