

# Package ‘pubmed.mineR’

June 24, 2015

**Type** Package

**Title** Text Mining of PubMed Abstracts

**Version** 1.0.4

**Date** 2015-06-23

**Author** Jyoti Rani, S.Ramachandran, Ab Rauf Shah

**Maintainer** S. Ramachandran <ramu@igib.in>

**Description**

Text mining of PubMed Abstracts (text and XML) from <http://www.ncbi.nlm.nih.gov/pubmed>.

**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\_BWI.R' 'BWI.R'  
'currentabs\_fn.R' 'previousabs\_fn.R' 'altnamesfun.R'  
'subsetabs.R'

**License** GPL-3

**LazyLoad** yes

**LazyData** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-06-24 11:15:43

**R topics documented:**

Abstracts-class	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
currentabs_fn	12
Find_conclusion	13
find_intro_conc_html	14
genes_BWI	15
GeneToEntrez	16
Genewise	16
Genewise-methods	17
gene_atomization	17
getabs	18
getabs-methods	19
getabsT	19
getabsT-methods	20
get_original_term	20
get_original_term2	21
HGNC-class	21
HGNC2UniprotID	22
HGNCdata	23
input_for_find_intro_conc_html	24
local_uniprotfun	25
previousabs_fn	25
printabs	26
pubtator_function	27
pubtator_result_list_to_table	28
readabs	28
ready	29
removeabs	30
removeabs-methods	30
searchabsL	31
searchabsL-methods	32
searchabsT	32
searchabsT-methods	33
sendabs	33
sendabs-methods	34

SentenceToken . . . . .	35
subabs . . . . .	35
subabs-methods . . . . .	36
subsetabs . . . . .	37
subsetabs-methods . . . . .	37
tdm_for_lsa . . . . .	38
uniprotfun . . . . .	38
whichcluster . . . . .	39
wordcluster . . . . .	40
wordscusterview . . . . .	41
word_atomizations . . . . .	42
xmlgene_atomizations . . . . .	43
xmlreadabs . . . . .	43
xmlword_atomizations . . . . .	44
Yearwise . . . . .	45
Yearwise-methods . . . . .	46
<b>Index</b>	<b>47</b>

---

Abstracts-class	<i>Class "Abstracts" Abstract Class</i>
-----------------	---

---

## 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

[searchabsL](#) [getabs](#) [contextSearch](#) [Genewise](#) [Yearwise](#) [combineabs](#) [subabs](#) [subsetabs](#) [readabs](#)

**Examples**

```
showClass("Abstracts")
```

---

altnamesfun

*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 the HGNC gene symbol.

**Value**

It returns a list of alternative names of given Genes.

**Author(s)**

S.Ramachandran

**References**

UniProt Consortium. "The universal protein resource (UniProt)." Nucleic acids research 36.suppl 1 (2008): D190-D195. <http://www.uniprot.org/>

**See Also**

[uniprotfun](#), ~~~

**Examples**

```
## Not run: test = altnamesfun(x)
## here x is the HGNC gene symbol for which alternative name is required.
```

---

BWI*To Get the Buzz Word Index of terms form the Abstracts.*

---

**Description**

This function is used to get the Buzz word index value for the terms. Buzz word became popular for a period of time.

**Usage**

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

**Arguments**

current	current an S4 object containing the Abstracts for the year we want to study. Output from currentabs_fn()
previous	previous an S4 object containing the Abstracts for years previous to our year of study. Output from previousabs_fn().
n	n is a character value 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**

Jensen, Lars Juhl, Jasmin Saric, and Peer Bork. "Literature mining for the biologist: from information retrieval to biological discovery." *Nature reviews genetics* 7.2 (2006): 119-129.

**See Also**

[genes\\_BWI](#)

**Examples**

```
## Not run: BWI(current, previous, n, N)
## here 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 word and theme respectively
```

---

cleanabs	<i>To clean the result of searchabsL</i>
----------	--

---

**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	<i>Methods for Function cleanabs</i>
------------------	--------------------------------------

---

**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 resulted abstracts.

---

cluster_words	<i>To Find the highest frequency of words within clusters</i>
---------------	---

---

**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

**Author(s)**

S. Ramachandran

**See Also**

[wordscluster](#)

**Examples**

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

---

combineabs	<i>To combine the abstracts</i>
------------	---------------------------------

---

**Description**

combineabs will automatically combine two abstracts of two objects.

**Usage**

```
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**

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

---

combineabs-methods	Abstracts <i>Method to Combine Abstracts</i>
--------------------	--

---

**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	<i>R Data containing words which frequently in text</i>
------------------	---

---

**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](https://en.wikipedia.org/wiki/Most_common_words_in_English)

**Examples**

```
data(common_words_new)
```

---

contextSearch	<i>For Context Search</i>
---------------	---------------------------

---

**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 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**

An R object and 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

**References**

[https://en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity)

**See Also**

[tdm\\_for\\_lsa](#)

**Examples**

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

---

cos_sim_calc_boot	<i>Cosine Similarity Calculation by Boot Strapping</i>
-------------------	--

---

**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 <code>tdm_for_lsa</code> 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](#)

**Examples**

```
## 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.
```

---

currentabs_fn	<i>To Retrive the Abstracts for year.</i>
---------------	---

---

**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 we want to extract the abstracts of theme "Atherosclerosis"
## from the large corpus of diabetes i.e. diabetesabs.
```

---

Find_conclusion	<i>To find the conclusion from the abstract(s).</i>
-----------------	---

---

**Description**

This function is designed for the user convinience, so that user can get the conclusion from the abstract(s) with out reading the whole abstract(s).

**Usage**

```
Find_conclusion(y)
```

**Arguments**

y                      An S4 object of class 'Abstract'.

**Value**

A list containing conclusion 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.

## Details

find\_intro\_conc\_html provide an HTML file containing space separated introduction and conclusion part from the abstracts of given query term as well as gives a link direct to PubMed for resulted 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")
## here 'abs' is an S4 object of class Abstracts
## and 'diet' is a term to be search from the abstracts
```

genes\_BWI

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

This function gives 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 the 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. Output from gene_atomization().

**Value**

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

**Author(s)**

S.Ramachandran

**See Also**

[BWI](#)

**Examples**

```
## Not run: test = genes_BWI(currentabs, previousabs, theme, genes)
## currentabs is an S4 object containig the Abstracts for the year we want to study.
## previousabs is an S4 object containig 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 list of genes output from gene_atomization().
```

---

GeneToEntrez	<i>Data containing Entrez Ids</i>
--------------	-----------------------------------

---

**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**

```
data(GeneToEntrez)
```

**Format**

The format is: chr "GeneToEntrez"

**Examples**

```
data(GeneToEntrez)
```

---

Genewise	<i>To Search the number of abstracts for Genes</i>
----------	--

---

**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 vector of gene names(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**

```
## Not run: Genewise(x, "TLR4")
## here 'x' contains the S4 object of Abstracts.
```

---

Genewise-methods	<i>method to find the abstracts for the given gene.</i>
------------------	---

---

**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, containiing abstracts. It will write a text file named "dataout.txt", containing the number of abstracts for the query gene terms

---

gene_atomization	<i>To Extract Genes from the Abstracts</i>
------------------	--

---

**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 occurence.

**Author(s)**

S.Ramachandran, Jyoti Rani

**Examples**

```
## Not run: gene_atomization(x)
## here x is an S4 object of class 'Abstracts' containing the abstracts
```

---

getabs

*To get Abstracts for a given term.*

---

**Description**

getabs will automatically fetch the abstracts containing the query term. A base function of the package pubmed.mineR.

**Usage**

```
getabs(object, x, y)
```

**Arguments**

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

**Details**

getabs() is used to find and extract the abstracts for any given term, from the large a large corpus of abstracts. It uses regexpr based search strategy.

**Value**

An S4 object of class 'Abstracts', containing the result abstracts for the given term.

**Author(s)**

Dr.S.Ramachandran

**Examples**

```
## Not run: getabs(x, "term")
## x is an S4 object of class abstracts containing the abstracts.
```

---

getabs-methods	getabs <i>To Get abstracts for a term</i>
----------------	---

---

**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	<i>To get Abstracts for a given term.</i>
---------	---

---

**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")
```

---

getabsT-methods	<i>To Get Abstracts</i>
-----------------	-------------------------

---

**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_original_term	<i>To get the original terms from the corpus.</i>
-------------------	---

---

**Description**

get\_original\_term is used to get the exact term as it is present in corpus.

**Usage**

```
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**

```
## 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	<i>To get the original terms from the corpus.</i>
--------------------	---

---

**Description**

get\_original\_term2 is used to get the exact term as it is present in corpus. It is originated from get\_original\_term to give more accurate result. It takes one term at once. For multiple terms we can use lapply.

**Usage**

```
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**

```
## Not run: test = get_original_term("hba1c", diababs)
## here it will return accurate formation of hba1c i.e. HbA1c from diababs.
```

---

HGNC-class	<i>HGNC Class for package.</i>
------------	--------------------------------

---

**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

**References**

UniProt Consortium. "The universal protein resource (UniProt)." Nucleic acids research 36.suppl 1 (2008): D190-D195. <http://www.uniprot.org/>

**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**

Povey, Sue, et al. "The HUGO gene nomenclature committee (HGNC)." Human genetics 109.6 (2001): 678-680. <http://www.genenames.org/>

**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**

<code>y</code>	an S4 object of class Abstracts
<code>all</code>	is logical if true, will include title and author otherwise only abstracts.

**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**

[literature/http://eutils.ncbi.nlm.nih.gov/](http://eutils.ncbi.nlm.nih.gov/literature/)

**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.
```



---

local_uniprotfun	<i>To Get Information from Uniprot.</i>
------------------	---

---

**Description**

It is an auxiliary function for altnamesfun.

**Usage**

```
local_uniprotfun(y)
```

**Arguments**

y                      y a character value containing HGNC Gene symbol

**Value**

It writes an output file named "x.txt" which will be used as input in altnamesfun().

**Author(s)**

S.Ramachandran, Jyoti Rani

**See Also**

[uniprotfun](#)

**Examples**

```
## Not run: local_uniprotfun("TLR4")
## here it will generate an output file named x.txt containing
## result for TLR4.
```

---

previousabs_fn	<i>To Retrive the Abstracts from the large corpus for given years.</i>
----------------	--

---

**Description**

This function is used to extract the abstracts from the large corpus excluding the years under study. 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 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 = currentabs_fn(as.character(2015:2010), "atherosclerosis", diabetesabs)
## here we will get the abstracts before 2010 for 'atherosclerosis' from the large corpus .
```

---

<code>printabs</code>	<i>To print the total number of abstracts in an S4 object of class Abstracts, its start and end</i>
-----------------------	---

---

**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(res1)
## here 'res1' is an S4 object of class Abstracts.
```

---

pubtator_function	<i>function for text annotation using online PubTator</i>
-------------------	---

---

## 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 describing 'PMID'.

## 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.

## Author(s)

S.Ramachandran, Jyoti Rani

## References

Wei CH et. al., PubTator: a Web-based text mining tool for assisting Biocuration, Nucleic acids research, 2013, 41 (W1): W518-W522. doi: 10.1093/nar/gkt44

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

Wei CH et. al., PubTator: A PubMed-like interactive curation system for document triage and literature curation, in Proceedings of BioCreative 2012 workshop, Washington DC, USA, 145-150, 2012

## Examples

```
## 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. The `pubtator_function()` gives output in a list. 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 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
```

---

readabs

*To read Abstracts*

---

### 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_filename.txt")
##here x 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	<i>To remove abstracts for the query term.</i>
-----------	--

---

### Description

removeabs will report the number of abstracts removed for the given query term.

### Usage

```
removeabs(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 specific

### Details

removeabs() finds the abstracts for the given term and remove them from the large set of abstracts. A text file of file name "dataout.txt" will be written containing the number of abstracts removed.

### Value

An S4 object of class Abstracts and a text file named "dataout.txt"

### Author(s)

S.Ramachandran, Jyoti Rani

### Examples

```
## Not run: removeabs(x, "term", TRUE)
```

---

removeabs-methods	removeabs <i>To remove abstracts of a term from the data.</i>
-------------------	---

---

### 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	<i>To Search the abstracts of term(s) in a combination mode.</i>
------------	--

---

## 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

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

---

searchabsL-methods	<i>Searching Abstracts</i>
--------------------	----------------------------

---

### Description

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

### Methods

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	<i>To Search Abstracts</i>
------------	----------------------------

---

### Description

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

### Usage

```
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 als written with the filename "out.txt".

**Author(s)**

Dr.S.Ramachandran

**See Also**

[searchabsT](#)

**Examples**

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

---

searchabsT-methods	searchabsT <i>Searching abstracts</i>
--------------------	---------------------------------------

---

**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	<i>To send abstracts</i>
---------	--------------------------

---

**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(x,"abs.txt")
## here 'x' 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	<i>To Tokenize the sentences</i>
---------------	----------------------------------

---

**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)
```

---

subabs	<i>To find sub-abstracts</i>
--------	------------------------------

---

**Description**

subabs will automatically extract the sub-abstracts from large set of abstracts.

**Usage**

```
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 asbstracts 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**

```
## Not run: subabs(x,1,5)
## Here 'x' is an S4 object of class 'Abstracts',
## 1 and 5 are the start and end point 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	<i>To make subsets of large corpus.</i>
-----------	---

---

**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	<i>To make subset of Abstracts.</i>
-------------------	-------------------------------------

---

**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	<i>create Term Document Matrix for lsa analysis</i>
-------------	---

---

### 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	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(diab_abs,y)
## End(Not run)
```

---

uniprotfun	<i>To get information about gene from the UniProt.</i>
------------	--

---

### Description

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

### Usage

```
uniprotfun(y)
```

**Arguments**

y                      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.

**Author(s)**

S.Ramachandran

**Examples**

```
## Not run: uniprotfun(x)
```

---

whichcluster	<i>To fetch the cluster for words</i>
--------------	---------------------------------------

---

**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 term.

**Value**

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

**Author(s)**

S.Ramachandran

**See Also**

[wordscluster](#)

**Examples**

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

wordcluster

*To cluster the words***Description**

wordcluster 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**

```
wordcluster(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 'explotion' 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 filenameed "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 'runaway' situations. Their frequencies are very rare. Runaway 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 runsaway 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 compnunds in IUPAC system of nomenclature.



**Author(s)**

S.Ramachandran, Jyoti Rani

**See Also**

[whichcluster word\\_atomizations](#)

**Examples**

```
## Not run:
test=wordsccluster(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	<i>To view the words in cluster</i>
------------------	-------------------------------------

---

**Description**

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

**Usage**

```
wordsclusterview(words_cluster, all)
```

**Arguments**

words_cluster	an R object containing output of wordsccluster
all	is logical and default is FALSE, if set TRUE including those with one member word.

**Details**

The first 5 words and 5 words near the median and 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

**See Also**

[wordsccluster](#)

**Examples**

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

---

word_atomizations	<i>Atomization of words</i>
-------------------	-----------------------------

---

**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**

```
## Not run: word_atomizations(x)
## here x is the object containing abstracts.
```

---

xmlgene_atomizations	<i>Gene atomization of xml abstracts.</i>
----------------------	---

---

**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	<i>To read the abstracts from the PubMed saved in XML format.</i>
------------	---

---

**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)
```

**Arguments**

file                    an XML file saved from PubMed.

**Value**

an S4 object of class Abstracts containing journals, abstracts and PMID.

**Author(s)**

S.Ramachandran

**See Also**

[readabs](#)

**Examples**

```
## Not run: test_run = xmlreadabs("pubmed_result.xml")
## here "pubmed_result.xml" is an xml format file downloaded from PubMed.
```

---

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

**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(x, "2011") or
Yearwise(x, c("2011", "2013", "2009"))
## End(Not run)
## Here 'x' is the object containing data of PubMed abstracts.
```

---

Yearwise-methods

Yearwise *Year wise extraction of Abstracts*

---

**Description**

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

**Methods**

signature(object = "Abstracts") This method "Yearwise" is written to fetch the abstracts yearly.

# Index

## \*Topic **Functions**

sendabs, [33](#)

## \*Topic **Function**

cleanabs, [6](#)

cluster\_words, [7](#)

combineabs, [7](#)

contextSearch, [9](#)

cos\_sim\_calc, [10](#)

cos\_sim\_calc\_boot, [11](#)

Find\_conclusion, [13](#)

find\_intro\_conc\_html, [14](#)

gene\_atomization, [17](#)

Genewise, [16](#)

get\_original\_term, [20](#)

getabs, [18](#)

getabsT, [19](#)

input\_for\_find\_intro\_conc\_html, [24](#)

printabs, [26](#)

readabs, [28](#)

removeabs, [30](#)

searchabsL, [31](#)

searchabsT, [32](#)

SentenceToken, [35](#)

subabs, [35](#)

uniprotfun, [38](#)

whichcluster, [39](#)

wordscluster, [40](#)

wordsclusterview, [41](#)

xmlgene\_atomizations, [43](#)

xmlreadabs, [43](#)

xmlword\_atomizations, [44](#)

Yearwise, [45](#)

## \*Topic **classes**

Abstracts-class, [3](#)

HGNC-class, [21](#)

## \*Topic **currentabs\_fn**

currentabs\_fn, [12](#)

## \*Topic **datasets**

common\_words\_new, [9](#)

GeneToEntrez, [16](#)

HGNC2UniprotID, [22](#)

HGNCdata, [23](#)

## \*Topic **function**

altnamesfun, [4](#)

BWI, [5](#)

genes\_BWI, [15](#)

pubtator\_function, [27](#)

ready, [29](#)

tdm\_for\_lsa, [38](#)

word\_atomizations, [42](#)

## \*Topic **get\_original\_term2**

get\_original\_term2, [21](#)

## \*Topic **local\_uniprot\_fun**

local\_uniprotfun, [25](#)

## \*Topic **methods**

cleanabs-methods, [6](#)

combineabs-methods, [8](#)

contextSearch-methods, [10](#)

Genewise-methods, [17](#)

getabs-methods, [19](#)

getabsT-methods, [20](#)

removeabs-methods, [30](#)

searchabsL-methods, [32](#)

searchabsT-methods, [33](#)

sendabs-methods, [34](#)

subabs-methods, [36](#)

subsetabs-methods, [37](#)

Yearwise-methods, [46](#)

## \*Topic **previousabs\_fn**

previousabs\_fn, [25](#)

## \*Topic **pubtator\_result\_list\_to\_table**

pubtator\_result\_list\_to\_table, [28](#)

## \*Topic **subsetabs**

subsetabs, [37](#)

Abstracts, [22](#)

Abstracts-class, [3](#)

altnamesfun, [4](#)

BWI, [5](#), [15](#)

cleanabs, [6](#)

cleanabs, Abstracts-method  
(cleanabs-methods), [6](#)

cleanabs-methods, [6](#)

cluster\_words, [7](#)

combineabs, [3](#), [7](#)

combineabs, Abstracts-method  
(combineabs-methods), [8](#)

combineabs-methods, [8](#)

common\_words\_new, [9](#)

contextSearch, [3](#), [9](#)

contextSearch, Abstracts-method  
(contextSearch-methods), [10](#)

contextSearch-methods, [10](#)

cos\_sim\_calc, [10](#)

cos\_sim\_calc\_boot, [11](#)

currentabs\_fn, [12](#), [26](#)

Find\_conclusion, [13](#)

find\_intro\_conc\_html, [14](#), [24](#)

gene\_atomization, [17](#)

genes\_BWI, [5](#), [15](#)

GeneToEntrez, [16](#)

Genewise, [3](#), [16](#)

Genewise, Abstracts-method  
(Genewise-methods), [17](#)

Genewise-methods, [17](#)

get\_original\_term, [20](#), [21](#)

get\_original\_term2, [21](#)

getabs, [3](#), [18](#)

getabs, Abstracts-method  
(getabs-methods), [19](#)

getabs-methods, [19](#)

getabsT, [19](#)

getabsT, Abstracts-method  
(getabsT-methods), [20](#)

getabsT-methods, [20](#)

HGNC-class, [21](#)

HGNC2UniprotID, [22](#)

HGNCdata, [23](#)

input\_for\_find\_intro\_conc\_html, [14](#), [24](#)

local\_uniprotfun, [25](#)

previousabs\_fn, [13](#), [25](#)

printabs, [26](#)

pubtator\_function, [27](#), [28](#)

pubtator\_result\_list\_to\_table, [28](#)

readabs, [3](#), [28](#), [44](#)

ready, [29](#)

removeabs, [30](#)

removeabs, Abstracts-method  
(removeabs-methods), [30](#)

removeabs-methods, [30](#)

searchabsL, [3](#), [6](#), [31](#)

searchabsL, Abstracts-method  
(searchabsL-methods), [32](#)

searchabsL-methods, [32](#)

searchabsT, [31](#), [32](#), [33](#)

searchabsT, Abstracts-method  
(searchabsT-methods), [33](#)

searchabsT-methods, [33](#)

sendabs, [33](#)

sendabs, Abstracts-method  
(sendabs-methods), [34](#)

sendabs-methods, [34](#)

SentenceToken, [35](#)

subabs, [3](#), [35](#)

subabs, Abstracts-method  
(subabs-methods), [36](#)

subabs-methods, [36](#)

subsetabs, [3](#), [37](#)

subsetabs, Abstracts-method  
(subsetabs-methods), [37](#)

subsetabs-methods, [37](#)

tdm\_for\_lsa, [11](#), [12](#), [38](#)

uniprotfun, [4](#), [25](#), [38](#)

whichcluster, [39](#), [41](#)

word\_atomizations, [41](#), [42](#)

wordscluster, [7](#), [20](#), [39](#), [40](#), [41](#)

wordsclusterview, [41](#)

xmlgene\_atomizations, [43](#)

xmlreadabs, [43](#), [43](#), [45](#)

xmlword\_atomizations, [44](#)

Yearwise, [3](#), [45](#)

Yearwise, Abstracts-method  
(Yearwise-methods), [46](#)

Yearwise-methods, [46](#)