

# Package ‘PubScore’

June 15, 2021

**Type** Package

**Title** Automatic calculation of literature relevance of genes

**Version** 1.4.0

**Description**

Calculates the importance score for a given gene. The importance score is the total counts of articles in the pubmed database that are a result for that gene AND each term of a list.

**Imports** ggplot2, igraph, ggrepel, rentrez, progress, graphics, dplyr, utils, methods, intergraph, network, sna

**Suggests** FCBF, plotly, SummarizedExperiment, SingleCellExperiment, knitr, rmarkdown, testthat (>= 2.1.0), BiocManager, biomaRt

**biocViews** GeneSetEnrichment, GeneExpression, SystemsBiology, Genetics, Epigenetics, BiomedicalInformatics, Visualization, SingleCell

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 4.0.0)

**git\_url** <https://git.bioconductor.org/packages/PubScore>

**git\_branch** RELEASE\_3\_13

**git\_last\_commit** 5e46d7d

**git\_last\_commit\_date** 2021-05-19

**Date/Publication** 2021-06-15

**Author** Tiago Lubiana [aut, cre],  
Helder Nakaya [aut, ths]

**Maintainer** Tiago Lubiana <tiago.lubiana.alves@usp.br>

**R topics documented:**

<code>.getSimulation_test</code>	2
<code>.query_pubmed</code>	3
<code>all_counts</code>	3
<code>gene2pubmed_db</code>	4
<code>getScore</code>	4
<code>get_all_counts</code>	5
<code>get_literature_score</code>	6
<code>heatmapViz</code>	7
<code>hgcn_entrez_reference</code>	7
<code>initialize, PubScore-method</code>	8
<code>networkViz</code>	9
<code>plot_literature_graph</code>	9
<code>plot_literature_score</code>	10
<code>pubscore</code>	11
<code>PubScore-class</code>	11
<code>set_all_counts&lt;-</code>	12
<code>test_score</code>	13

**Index** **15**


---

`.getSimulation_test`    *Auxiliary function for the test method*

---

**Description**

Auxiliary function for the test method

**Usage**

```
.getSimulation_test(pub, ambiguous = c(), n_simulations)
```

**Arguments**

<code>pub</code>	An object of class PubScore
<code>ambiguous</code>	A character vector with possible ambiguous gene names
<code>n_simulations</code>	The number of simulations to run.

**Value**

A data-frame with a simulation of literature scores for random samplings

---

.query\_pubmed                    #' .query\_pubmed

---

### Description

Auxiliary function for getting the list score

### Usage

```
.query_pubmed(search_topic, wait_time = 0, ret_max = 1)
```

### Arguments

search_topic	Item to search on PubMed via rentrez
wait_time	Time between searches
ret_max	Number of IDs to be returned. Defaults to 1.

### Value

The rentrez search result (a list)

---

all\_counts                    *all\_counts*

---

### Description

A dataframe with all pubmed counts for the genes in the Dengue dataset in relation to the term "Dengue".

### Usage

```
data(all_counts)
```

### Format

An object of class `data.frame`

### Details

Outcome of the `test_score` method of the `pubscore` class. As this function may take a long time, this dataset speeds up the vignette.

Contains: 3 columns: `#tax_id`: The reference ID for the taxon. All are 9606 (humans). `GeneID`: The Entrez ID code for a given gene. `PubMedID`: A PubMed ID for a paper that mentions the gene in the "Gene ID" column.

1335548 rows: gene-paper associations in the `gene2pubmed` database.

---

gene2pubmed_db	<i>human genes on gene2pubmed_db</i>
----------------	--------------------------------------

---

### Description

A subset of the gene2pubmed database downloaded via FTP from <ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2pubmed.gz>.  
 # The subset contains only the rows corresponding to humans (#tax\_id = 906) The table was downloaded in October 2019.

### Usage

```
data(gene2pubmed_db)
```

### Format

An object of class `data.frame`

### Details

Contains: 3 columns: #tax\_id: The reference ID for the taxon. All are 9606 (humans). GeneID: The Entrez ID code for a given gene. PubMedID: A PubMed ID for a paper that mentions the gene in the "Gene ID" column.

1335548 rows: gene-paper associations in the gene2pubmed database.

### References

Maglott, Donna, et al. 'Entrez Gene: gene-centered information at NCBI.' Nucleic acids research 33.suppl\_1 (2005): D54-D58.

---

getScore	<i>Retrieve the literature_score attribute</i>
----------	--

---

### Description

Retrieve the literature\_score attribute

### Usage

```
getScore(pub)
```

```
## S4 method for signature 'PubScore'  
getScore(pub)
```

### Arguments

pub                    Object of class PubScore

**Value**

A "numeric" with the literature score for this gene x term combination

**Examples**

```
# Create a new pubscore object
pub <- pubscore(genes = c('cd4', 'cd8'), terms_of_interest = c('blabla', 'immunity'))
plot(networkViz(pub))
```

---

get_all_counts	<i>Retrieve the all_counts attribute</i>
----------------	--

---

**Description**

Retrieve the all\_counts attribute

**Usage**

```
get_all_counts(pub)

## S4 method for signature 'PubScore'
get_all_counts(pub)
```

**Arguments**

pub                    Object of class PubScore

**Value**

A dataframe containing the counts table for all genes.

**Examples**

```
# Create a new pubscore object
pub <- pubscore(genes = c('cd4', 'cd8'), terms_of_interest = c('blabla', 'immunity'))
plot(networkViz(pub))
```

---

```
get_literature_score  get_literature_score
```

---

### Description

Calculates the importance score for a given gene. The importance score is the total counts of articles in the pubmed database that are a result for that gene AND each term of a list

### Usage

```
get_literature_score(  
  genes,  
  terms_of_interest,  
  gene2pubmed = FALSE,  
  return_all = FALSE,  
  wait_time = 0,  
  show_progress = TRUE,  
  verbose = FALSE  
)
```

### Arguments

genes	A vector with multiple genes.
terms_of_interest	A list of terms of interest related to the topic you want to find the relevance for
gene2pubmed	logical defining if gene2pubmed db is going to be used. If used, the vector of genes has to be of HUMAN genes in the hgcn_symbol format.
return_all	Only to be used with gene2pubmed! logical defining if the all_counts table is going to be returned here. Usually it is generated by the test_score function.
wait_time	How long should be the interval between two requests to the ENTREZ database when it fails. Defaults to 0. In seconds.
show_progress	If TRUE, a progress bar is displayed. Defaults to TRUE.
verbose	If TRUE, will display the index of the search occuring. Defaults to FALSE.

### Value

A dataframe with the literature scores.

### Examples

```
genes <- c('CD8A', 'CD4')  
terms_of_interest <-  
  c(  
    "CD4 T cell",  
    "CD14+ Monocyte",  
    "B cell",
```

```

      "CD8 T cell",
      "FCGR3A+ Monocyte",
      "NK cell",
      "Dendritic cell",
      "Megakaryocyte",
      'immunity'
    )
  get_literature_score(genes, terms_of_interest)

```

---

heatmapViz

*Retrieve the heatmap attribute*


---

### Description

Retrieve the heatmap attribute

### Usage

```

heatmapViz(pub)

## S4 method for signature 'PubScore'
heatmapViz(pub)

```

### Arguments

pub                    Object of class PubScore

### Value

A "gg" object, from ggplot2, containing a heatmap from the counts table.

### Examples

```

#Create a new pubscore object
pub <- pubscore(genes = c('cd4', 'cd8'), terms_of_interest = c('blabla', 'immunity'))
plot(heatmapViz(pub))

```

---

hgcn\_entrez\_reference    *hgcn\_entrez\_reference*


---

### Description

Contains the result of a query to the biomaRt service done in October, 2019.

### Usage

```

data(hgcn_entrez_reference)

```

**Format**

An object of class `data.frame`

**Details**

2 columns: `entrezgene_id` (containing the Entrez ids) and `hgnc_symbol` (containing gene symbols from the HUGO gene nomenclature consortium)

20491 rows, for the mapping between the two nomenclatures for human genes.

**References**

Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package `biomaRt`. Steffen Durinck, Paul T. Spellman, Ewan Birney and Wolfgang Huber, *Nature Protocols* 4, 1184-1191 (2009).

---

```
initialize, PubScore-method
      initialize
```

---

**Description**

`initialize`

**Usage**

```
## S4 method for signature 'PubScore'
initialize(.Object, genes, terms_of_interest, gene2pubmed = FALSE)
```

**Arguments**

<code>.Object</code>	The object of signature <code>PubScore</code> that is going to be created
<code>genes</code>	The genes to which you want to calculate and visualize the literature score.
<code>terms_of_interest</code>	A list of terms of interest related to the topic you want to find the relevance.
<code>gene2pubmed</code>	Logical (TRUE / FALSE) defining if <code>gene2pubmed</code> db is going to be used.

**Value**

A object of the `PubScore` class



---

networkViz	<i>Retrieve the network attribute</i>
------------	---------------------------------------

---

**Description**

Retrieve the network attribute

**Usage**

```
networkViz(pub)

## S4 method for signature 'PubScore'
networkViz(pub)
```

**Arguments**

pub                    Object of class PubScore

**Value**

A "gg" object, from ggplot2, containing a network from the counts table.

**Examples**

```
# Create a new pubscore object
pub <- pubscore(genes = c('cd4', 'cd8'), terms_of_interest = c('blabla', 'immunity'))
plot(networkViz(pub))
```

---

```
plot_literature_graph #'plot_literature_graph'
```

---

**Description**

Plot a graph inspired in CEMiTool's graphs

**Usage**

```
plot_literature_graph(
  plot_counts,
  name,
  color = "#B30000FF",
  max_number_of_labels = 10
)
```

**Arguments**

<code>plot_counts</code>	The dataframe returned from the <code>get_literature_score</code> function
<code>name</code>	The name of the plot.
<code>color</code>	The color of the plot. Defaults to a shade of red ("B30000FF").
<code>max_number_of_labels</code>	The max number of gene labels to show. Defaults to 10.

**Value**

A plotly/ggplot2 object is either returned or directly plotted

**Examples**

```
gene <- c('CD4', 'CD14', "AIF1", "ACVR1", "CDY2A")
terms_of_interest <- c("CD4 T cell", "CD14+ Monocyte")
literature_counts <- get_literature_score(gene, terms_of_interest)
pl <- plot_literature_graph(literature_counts, name = 'test')
pl
```

---

`plot_literature_score` *plot\_literature\_score*

---

**Description**

Plots a non-clusterized heatmap of the article counts for the combination of gene list and list of terms NOTE: the object has to be exactly the one returned by `get_literature_score.R`. Otherwise `ggplot2` will not be able to identify the correct columns.

**Usage**

```
plot_literature_score(plot_counts, return_ggplot = FALSE, is_plotly = FALSE)
```

**Arguments**

<code>plot_counts</code>	The dataframe returned from the <code>get_literature_score</code> function
<code>return_ggplot</code>	If TRUE, returns a <code>ggplot2</code> object instead of plotting. Defaults to FALSE.
<code>is_plotly</code>	If TRUE, a interactive plot is plotted in the place o static <code>ggplot</code> . Defaults to FALSE.

**Value**

A `ggplot2` object is either returned or directly plotted

**Examples**

```
gene <- c('CD4', 'CD14', "AIF1", "ACVR1", "CDY2A")
terms_of_interest <- c("CD4 T cell", "CD14+ Monocyte", "B cell",
  "CD8 T cell", "FCGR3A+ Monocyte", "NK cell", "Dendritic cell",
  "Megakaryocyte", 'immunity')
literature_counts <- get_literature_score(gene, terms_of_interest)
P <- plot_literature_score(literature_counts, return_ggplot = TRUE)
plot(P)
```

pubscore

*PubScore fundamental analysis***Description**

Runs the initialization and the basic functions for querying pubmed and getting the literature scores.

**Usage**

```
pubscore(terms_of_interest, genes, gene2pubmed = FALSE)
```

**Arguments**

terms_of_interest	A list of terms of interest related to the topic you want to find the relevance for
genes	A vector with multiple genes.
gene2pubmed	Logical (TRUE / FALSE) defining if gene2pubmed db is going to be used. Defaults to FALSE.

**Value**

Object of class PubScore

PubScore-class

*An S4 class to represent PubScore results***Description**

The S4 class to PubScore and its basic initialize and show methods.

**Slots**

terms\_of\_interest A list of terms of interest related to the topic you want to find the relevance.  
 genes The genes to which you want to calculate and visualize the literature score.  
 date The date when the object was initialized. PubScore counts will likely increase with time.  
 gene2pubmed Logical (TRUE / FALSE) noting if gene to pubmed was used or not.  
 counts A data frame with the counts retrieved on PubMed  
 network A visualization of the results found in a network  
 heatmap A visualization of the results found in a heatmap

---

set\_all\_counts<-        *Set the all\_counts attribute*

---

**Description**

Set the all\_counts attribute

**Usage**

```
set_all_counts(pub) <- value

## S4 replacement method for signature 'PubScore'
set_all_counts(pub) <- value
```

**Arguments**

pub                    Object of class PubScore  
 value                 The table with all gene x term article counts from the "test\_score" method.

**Value**

A dataframe containing the counts table for all genes.

**Examples**

```
terms_of_interest <- c('Dengue')
pub <- pubscore(terms_of_interest = terms_of_interest, genes = c("CD4", "CD8", "CD14") )
print(getScore(pub))
data("all_counts")
set_all_counts(pub) <- all_counts
```

---

test_score	<i>Test the literature enrichment score</i>
------------	---

---

## Description

Test the literature enrichment score

## Usage

```
test_score(
  pub,
  total_genes,
  show_progress = TRUE,
  remove_ambiguous = TRUE,
  verbose = FALSE,
  nsim = 1e+05,
  ambiguous_terms = c("PC", "JUN", "IMPACT", "ACHE", "SRI", "SET", "CS", "PROC", "MET",
    "SHE", "CAD", "DDT", "PIGS", "SARS", "REST", "GC", "CP", "STAR", "SI", "GAN", "MARS",
    "SDS", "AGA", "NHS", "CPE", "POR", "MAX", "CAT", "LUM", "ANG", "POLE", "CLOCK",
    "TANK", "ITCH", "SDS", "AES", "CIC", "FST", "CAPS", "COPE", "F2", "AFM", "SPR",
    "PALM", "C2", "BAD", "GPI", "CA2", "SMS", "INVS", "WARS", "HP", "GAL", "SON", "AFM",
    "BORA", "MBP", "MAK", "MALL", "COIL", "CAST ")
)

## S4 method for signature 'PubScore'
test_score(
  pub,
  total_genes,
  show_progress = TRUE,
  remove_ambiguous = TRUE,
  verbose = FALSE,
  nsim = 1e+05,
  ambiguous_terms = c("PC", "JUN", "IMPACT", "ACHE", "SRI", "SET", "CS", "PROC", "MET",
    "SHE", "CAD", "DDT", "PIGS", "SARS", "REST", "GC", "CP", "STAR", "SI", "GAN", "MARS",
    "SDS", "AGA", "NHS", "CPE", "POR", "MAX", "CAT", "LUM", "ANG", "POLE", "CLOCK",
    "TANK", "ITCH", "SDS", "AES", "CIC", "FST", "CAPS", "COPE", "F2", "AFM", "SPR",
    "PALM", "C2", "BAD", "GPI", "CA2", "SMS", "INVS", "WARS", "HP", "GAL", "SON", "AFM",
    "BORA", "MBP", "MAK", "MALL", "COIL", "CAST ")
)
```

## Arguments

pub	Object of class PubScore
total_genes	A list of all the possible genes in your study. Usually all the names in the rows of an "exprs" object.
show_progress	If TRUE, a progress bar is displayed. Defaults to True.

`remove_ambiguous` If TRUE, ambiguously named genes (such as "MARCH") will be removed. Defaults to TRUE.

`verbose` If TRUE, will display the index of the search occurring. Defaults to false.

`nsim` The number of simulations to run. Defaults to 100000.

`ambiguous_terms` A character vector of the ambiguous terms to use instead of the default. The default includes 60 genes pre-selected as ambiguous (as IMPACT, MARCH and ACHE).

### Value

A "gg" object, from ggplot2, containing a network from the counts table.

### Examples

```
# Create a new pubscore object
pub <- pubscore(genes = c('cd4', 'cd8'),
  terms_of_interest = c('blabla', 'immunity'))
pub <- test_score(pub,
  total_genes = c('notagene1', 'notagene2', 'cd4', 'cd8'),
  remove_ambiguous = TRUE)
```

# Index

- \* **datasets**,
  - all\_counts, 3
  - gene2pubmed\_db, 4
  - hgcn\_entrez\_reference, 7
- \* **literature**
  - all\_counts, 3
  - gene2pubmed\_db, 4
  - hgcn\_entrez\_reference, 7
- \* **pubmed**,
  - all\_counts, 3
  - gene2pubmed\_db, 4
  - hgcn\_entrez\_reference, 7
- \* **test**,
  - all\_counts, 3
  - gene2pubmed\_db, 4
  - hgcn\_entrez\_reference, 7
- .getSimulation\_test, 2
- .query\_pubmed, 3
  
- all\_counts, 3
  
- gene2pubmed\_db, 4
- get\_all\_counts, 5
- get\_all\_counts, PubScore-method  
(get\_all\_counts), 5
- get\_literature\_score, 6
- getScore, 4
- getScore, PubScore-method (getScore), 4
  
- heatmapViz, 7
- heatmapViz, PubScore-method  
(heatmapViz), 7
- hgcn\_entrez\_reference, 7
  
- initialize, PubScore-method, 8
  
- networkViz, 9
- networkViz, PubScore-method  
(networkViz), 9
  
- plot\_literature\_graph, 9
  
- plot\_literature\_score, 10
- pubscore, 11
- PubScore-class, 11
  
- set\_all\_counts<-, 12
- set\_all\_counts<-, PubScore-method  
(set\_all\_counts<-), 12
  
- test\_score, 13
- test\_score, PubScore-method  
(test\_score), 13