

Package ‘rscc’

July 23, 2025

Type Package

Title R Source Code Similarity Evaluation by Variable/Function Names

Version 0.2.1

Date 2022-01-20

Description

Evaluates R source codes by variable and/or functions names. Similar source codes should deliver similarity coefficients near one. Since neither the frequency nor the order of the used names is considered, a manual inspection of the R source code is required to check for similarity. Possible use cases include detection of code clones for improving software quality and of plagiarism amongst students' assignments.

License GPL-3

URL <https://github.com/sigbertklinke/rscc> (development version)

Imports crayon, formatR, highlight, igraph, tm

Encoding UTF-8

RoxygenNote 7.1.2

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Sigbert Klinke [aut, cre]

Maintainer Sigbert Klinke <sigbert@hu-berlin.de>

Repository CRAN

Date/Publication 2022-01-20 12:02:42 UTC

Contents

as_igraph	2
browse	3
documents	3
freq_table	4
matrix2dataframe	5

same_file	6
similarity_coeff	6
sims	7
sim_coeff	8
sourcecode	9
tfidf	9
Index	11

as_igraph	<i>as.igraph</i>
-----------	------------------

Description

Converts a data frame of similarity coefficients into a graph.

Usage

```
as_igraph(x, tol = 100 * .Machine$double.eps, tol1 = 8 * tol, ...)
```

Arguments

- x a similarity object
- tol numeric scalar >= 0. Smaller differences are not considered, see [all.equal.numeric](#).
- tol1 numeric scalar >= 0. `isSymmetric.matrix()` ‘pre-tests’ the first and last few rows for fast detection of ‘obviously’ asymmetric cases with this tolerance. Setting it to length zero will skip the pre-tests.
- ... further parameters used by [igraph::graph_from_adjacency_matrix](#)

Value

an igraph object

Examples

```
files <- list.files(path=system.file("examples", package="rsc"), pattern="*.R$", full.names = TRUE)
prgs <- sourcecode(files, title=basename(files))
docs <- documents(prgs)
simm <- similarities(docs)
# a similarity coefficients equal to zero does not create an edge!
g <- as_igraph(simm, diag=FALSE)
# thicker edges have higher similarity coefficients
plot(g, edge.width=1+3*igraph::E(g)$weight)
```

browse	<i>browse</i>
--------	---------------

Description

Creates a temporary HTML file with source codes and opens it into a browser using `browseURL`. Note that the source code is reformatted.

Usage

```
browse(prgs, simdf, n = (simdf[, 3] > 0), width.cutoff = 60, css = NULL)
```

Arguments

<code>prgs</code>	sourcecode object
<code>simdf</code>	similarity object
<code>n</code>	integer: comparisons to show (default: <code>simf[,3]>0</code>)
<code>width.cutoff</code>	integer: an integer in [20, 500]: if a line's character length is at or over this number, the function will try to break it into a new line (default: 60)
<code>css</code>	character: file name of CSS style for highlighting the R code

Value

invisibly the name of the temporary HTML file

Examples

```
# example files are taken from https://CRAN.R-project.org/package=SimilaR
files <- list.files(system.file("examples", package="rsc"), "*.R$", full.names=TRUE)
prgs <- sourcecode(files)
simm <- similarities(documents(prgs))
simdf <- matrix2dataframe(simm)
if (interactive()) browse(prgs, simdf)
```

documents	<i>documents</i>
-----------	------------------

Description

Creates word vectors from parsed source code objects. If

- `type=="vars"` then the names of `all.vars(.)`,
- `type=="funs"` then the names of `setdiff(all.names(.), all.vars(.))`, and
- `type=="names"` then the names of `all.names(.)`

are used.

Usage

```
documents(
  prgs,
  type = c("vars", "funs", "names"),
  ignore.case = TRUE,
  minlen = 2,
  ...
)
```

Arguments

prgs	prgs sourcecode object
type	character: either "vars", "funs", "names" (default: "vars")
ignore.case	logical: If TRUE, case is ignored for computing (default: TRUE)
minlen	integer: minimal name length to be considered (default: 2)
...	unused

Value

a

Examples

```
# example files are taken from https://CRAN.R-project.org/package=SimilaR
files <- list.files(system.file("examples", package="rsc"), "*.R$", full.names=TRUE)
prgs <- sourcecode(files, basenane=TRUE)
docs <- documents(prgs)
docs
```

freq_table

freq_table

Description

Computes a frequency table of words and documents.

Usage

```
freq_table(docs, ...)
```

Arguments

docs	documents object
...	unused

Value

a matrix with similarities

Examples

```
# example files are taken from https://CRAN.R-project.org/package=SimilaR
files <- list.files(system.file("examples", package="rsc"), "*.R$", full.names=TRUE)
prgs <- sourcecode(files, basename=TRUE)
docs <- documents(prgs)
freq_table (docs)
```

matrix2dataframe	<i>matrix2dataframe</i>
------------------	-------------------------

Description

Converts a numeric matrix to a data frame with decreasing or increasing values: First column row index, second column col index and third column the value. If the matrix is symmetric, only the upper triangle is taken into account.

Usage

```
matrix2dataframe(
  m,
  decreasing = TRUE,
  tol = 100 * .Machine$double.eps,
  tol1 = 8 * tol,
  ...
)
```

Arguments

<code>m</code>	numeric: a matrix of values
<code>decreasing</code>	logical: should the sort order be increasing or decreasing (default: TRUE)
<code>tol</code>	numeric scalar ≥ 0 . Smaller differences are not considered, see all.equal.numeric .
<code>tol1</code>	numeric scalar ≥ 0 . <code>isSymmetric.matrix()</code> ‘pre-tests’ the first and last few rows for fast detection of ‘obviously’ asymmetric cases with this tolerance. Setting it to length zero will skip the pre-tests.
<code>...</code>	further arguments passed to methods; the matrix method passes these to all.equal . If the row and column names of object are allowed to differ for the symmetry check do use <code>check.attributes = FALSE</code> !

Value

a data frame with an attribute `matrix` with `m`

Examples

```
# non-symmetric
x <- matrix(runif(9), ncol=3)
matrix2dataframe(x)
```

same_file	<i>same_file</i>
-----------	------------------

Description

same_file

Usage

```
same_file(m, replacement = 0)
```

Arguments

m	matrix object with row- and columnnames
replacement	value for replacement (default: 0)

Value

matrix

Examples

```
m <- matrix(runif(25), ncol=5)
colnames(m) <- rownames(m) <- c(sprintf("m[%.f]", 1:3), sprintf("m2[%.f]", 1:2))
m
same_file(m)
```

similarity_coeff	<i>similarity_coeff</i>
------------------	-------------------------

Description

Computes a similarity coefficient based on the unique elements set1 and set2 in relation to setfull. If setfull is NULL then setfull is set to unique(c(set1, set2)). For more details, see the vignette vignette("rscc").

Usage

```
similarity_coeff(
  set1,
  set2,
  setfull = NULL,
  coeff = c("jaccard", "braun", "dice", "hamann", "kappa", "kulczynski", "ochiai",
    "phi", "russelrao", "matching", "simpson", "sneath", "tanimoto", "yule")
)
```

Arguments

set1	vector: elements to compare
set2	vector: elements to compare
setfull	vector: elements to compare (default: NULL)
coeff	character: coefficient to compute (default: "jaccard"), abbreviations can be used

Value

a numeric similarity coefficient

Examples

```
s1 <- 1:3
s2 <- 1:5
similarity_coeff(s1, s2)
s1 <- letters[1:3]
s2 <- LETTERS[1:5]
similarity_coeff(s1, s2)
```

sims

similarities

Description

sims and similarities both calculate for each pair of source code objects the similarity coefficients and return a data frame with the coefficients in descending order. A larger coefficient means a greater similarity.

Usage

```
sims(...)

similarities(
  docs,
  all = FALSE,
  coeff = c("jaccard", "braun", "dice", "hamann", "kappa", "kulczynski", "ochiai",
    "phi", "russelrao", "matching", "simpson", "sneath", "tanimoto", "yule")
)
```

Arguments

...	all parameters in sims are given to similarities
docs	document object
all	logical: should the similarity coefficients computed based on all sourcecode objects or just the two considered (default: FALSE)
coeff	character: coefficient to compute (default: "jaccard"), abbreviations can be used

Value

a data frame with the results

Examples

```
# example files are taken from https://CRAN.R-project.org/package=SimilaR
files <- list.files(system.file("examples", package="rsc"), "*.R$", full.names=TRUE)
prgs <- sourcecode(files, basenane=TRUE)
docs <- documents(prgs)
similarities(docs)
# further steps
# m <- similarities(docs)
# df <- matrix2dataframe(m)
# head(df, n=20)
# browse(prgs, df, n=5)
```

sim_coeff

sim_coeff

Description

Internal function for faster computation. No checks on input will be performed.

Usage

```
sim_coeff(set1, set2, setfull, coeff)
```

Arguments

set1	character: unique vector of words
set2	character: unique vector of words
setfull	character: unique vector of texts to compare
coeff	character: name of similarity coefficient to use

Value

value of similarity coefficient

sourcecode	<i>sourcecode</i>
------------	-------------------

Description

Reads and parses files with R source code.

Usage

```
sourcecode(x, ...)

## Default S3 method:
sourcecode(x, title = x, silent = FALSE, minlines = -1, ...)
```

Arguments

x	character: filenames
...	unused
title	character: vector of program titles (default: x)
silent	logical: should the report of messages be suppressed (default: FALSE)
minlines	integer: only expressions with minlines lines are considered (default: -1), if minlines<0 then whole files will be considered

Value

a sourcecode object

Examples

```
# example files are taken from https://CRAN.R-project.org/package=SimilaR
files <- list.files(system.file("examples", package="rsccl"), "*.R$", full.names=TRUE)
prgs <- sourcecode(files)
```

tfidf	<i>tfidf</i>
-------	--------------

Description

Computes the term frequency-inverse document frequency uses the cosine of the angles between the documents as similarity measure. Since R source code is provided no stemming or stop words are applied.

Usage

```
tfidf(docs)
```

Arguments

docs document object

Value

similarity matrix

Examples

```
files <- list.files(system.file("examples", package="rsc"), "*.R$", full.names = TRUE)
prgs  <- sourcecode(files, basename=TRUE, silent=TRUE)
docs  <- documents(prgs)
tfidf(docs)
# further steps
# m  <- tfidf(docs)
# df <- matrix2dataframe(m)
# head(df, n=20)
# browse(prgs, df, n=5)
```

Index

`all.equal`, [5](#)
`all.equal.numeric`, [2](#), [5](#)
`as_igraph`, [2](#)

`browse`, [3](#)

`documents`, [3](#)

`freq_table`, [4](#)

`igraph::graph_from_adjacency_matrix`, [2](#)

`matrix2dataframe`, [5](#)

`same_file`, [6](#)
`sim_coeff`, [8](#)
`similarities(sims)`, [7](#)
`similarity_coeff`, [6](#)
`sims`, [7](#)
`sourcecode`, [9](#)

`tfidf`, [9](#)