Package ‘SimilaR’

June 26, 2020

Version 1.0.8

Date 2020-06-26

Title R Source Code Similarity Evaluation

Description An implementation of a novel method to quantify the similarity of the code-base of R functions by means of program dependence graphs. Possible use cases include detection of code clones for improving software quality and of plagiarism amongst students' assignments.

URL https://github.com/bartoszukm/SimilaR

BugReports https://github.com/bartoszukm/SimilaR/issues

Type Package

Depends R (>= 3.1.0)

License GPL (>= 3)

Encoding UTF-8

Imports Rcpp (>= 0.12.0), stringi

Suggests testthat

LinkingTo Rcpp (>= 0.12.0), BH

SystemRequirements C++11

RoxygenNote 7.1.0

NeedsCompilation yes

Author Maciej Bartoszuk [aut, cre] (<https://orcid.org/0000-0001-6088-8273>), Marek Gagolewski [aut] (<https://orcid.org/0000-0003-0637-6028>)

Maintainer Maciej Bartoszuk <bartoszuk@rexamine.com>

Repository CRAN

Date/Publication 2020-06-26 09:40:02 UTC

R topics documented:

SimilaR-package *The SimilaR Package*

Description

See SimilaR_fromDirectory() for details.

Author(s)

Maciej Bartoszuk, Marek Gagolewski

SimilaR_fromDirectory *Quantify the Similarity of Pairs of R Functions*

Description

An implementation of the SimilaR algorithm - a method to quantify the similarity of R functions based on Program Dependence Graphs. Possible use cases include detection of code clones for improving software quality and of plagiarism among students' homework assignments.

SimilaR_fromDirectory scans for function definitions in all *.R source files in a given directory and performs pairwise comparisons.

SimilaR_fromTwoFunctions compares the code-base of two function objects.

Usage

```
SimilaR_fromDirectory(
  dirname,
  returnType = c("data.frame", "matrix"),
  fileTypes = c("function", "file"),
  aggregation = c("tnorm", "sym", "both")
)

SimilaR_fromTwoFunctions(
  function1,
  function2,
  functionNames,
  returnType = c("data.frame", "matrix"),
  aggregation = c("tnorm", "sym", "both")
)
```

Arguments

`dirname` path to a directory with source files named `*.R`

`returnType` `"data.frame"` or `"matrix"`; indicates the output object type

`fileTypes` `"function"` or `"file"`; indicates which pairs of functions extracted from the source files in `dirname` should be compared; `"function"` compares each function against every other function; `"file"` compares only the functions defined in different source files

`aggregation` `"sym"`, `"tnorm"`, or `"both"`; specifies which model of similarity asymmetry should be used; `"sym"` means that one (overall) similarity degree is computed; `"both"` evaluates and returns the degree to which the first function in a function pair is similar ("contained in", "is subset of") to the second one, and, separately, the extent to which the second function is similar to the first one; `"tnorm"` computes two similarity values and aggregates them to a single number

`function1` a first function object to compare

`function2` a second function object to compare

`functionNames` optional functions' names to be included in the output

Details

Note that, depending on the `"aggregation"` argument, the method may either return a single value, representing the overall (symmetric) similarity between a pair of functions, or or two different values, measuring the (non-symmetric) degrees of "subsethood". The user might possibly wish to aggregate these two values by means of some custom aggregation function.

Value

If `returnType` is equal to "data.frame", a data frame that gives the information about the similarity of the inspected pairs of functions, row by row, is returned. The data frame has the following columns:

- `name1` - the name of the first function in a pair
- `name2` - the name of the second function in a pair
- `SimilaR` - values in the [0,1] interval as returned by the SimilaR algorithm; 1 denotes that the functions are equivalent, while 0 means that they are totally dissimilar; if `aggregation` is equal to `"both"`, two similarity values are given: the one with suffix `"12"` quantifies the degree to which the first function is a subset of the second, and the another one with suffix `"21"` measures the extent to which the second function is a subset of the first one
- `decision` - 0 or 1; 1 means that two functions are classified as similar and 0 otherwise.

Rows in the data frame are sorted with respect to the `SimilaR` column (descending). Of course, `SimilaR_fromTwoFunctions` gives a data frame with only one row.

If `returnType` is equal to `"matrix"`, a square matrix is returned. The element at index (i,j) equals to the similarity degree between the i-th and the j-th function. When `aggregation` is equal to `"sym"` or `"tnorm"`, the matrix is symmetric. Column names and row names of the matrix are generated from the names of the functions being compared.

References

Bartoszuk M., A source code similarity assessment system for functional programming languages based on machine learning and data aggregation methods, Ph.D. thesis, Warsaw University of Technology, Warsaw, Poland, 2018.

Bartoszuk M., Gagolewski M., *Binary aggregation functions in software plagiarism detection*, In: *Proc. FUZZ-IEEE'17*, IEEE, 2017.

Bartoszuk M., Beliakov G., Gagolewski M., James S., *Fitting aggregation functions to data: Part II - Idempotentization*, In: Carvalho J.P. et al. (Eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II (Communications in Computer and Information Science 611)*, Springer, 2016, pp. 780-789. doi:10.1007/978-3-319-40581-0_63.

Bartoszuk M., Beliakov G., Gagolewski M., James S., *Fitting aggregation functions to data: Part I - Linearization and regularization*, In: Carvalho J.P. et al. (Eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II (Communications in Computer and Information Science 611)*, Springer, 2016, pp. 767-779. doi:10.1007/978-3-319-40581-0_62.

Bartoszuk M., Gagolewski M., *Detecting similarity of R functions via a fusion of multiple heuristic methods*, In: Alonso J.M., Bustince H., Reformat M. (Eds.), *Proc. IFSA/EUSFLAT 2015*, Atlantis Press, 2015, pp. 419-426.

Bartoszuk M., Gagolewski M., *A fuzzy R code similarity detection algorithm*, In: Laurent A. et al. (Eds.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part III (CCIS 444)*, Springer-Verlag, Heidelberg, 2014, pp. 21-30.

Examples

```
f1 <- function(x) {x*x}
f2 <- function(x,y) {x+y}

## A data frame is returned: 1 row, 4 columns
SimilaR_fromTwoFunctions(f1,
                         f2,
                         returnType = "data.frame",
                         aggregation = "tnorm")

## Custom names in the returned data frame
SimilaR_fromTwoFunctions(f1,
                         f2,
                         functionNames = c("first", "second"),
                         returnType = "data.frame",
                         aggregation = "tnorm")

## A data frame is returned: 1 row, 5 columns
SimilaR_fromTwoFunctions(f1,
                         f2,
                         returnType = "data.frame",
                         aggregation = "both")

## A non-symmetric square matrix is returned,
## with 2 rows and 2 columns
SimilaR_fromTwoFunctions(f1,
                         f2,
                         returnType = "matrix",
                         aggregation = "both")


## Typical example, where we wish to compare the functions from different files,
## but we do not want to compare the functions from the same file.
## There will be one value describing the overall similarity level.
SimilaR_fromDirectory(system.file("testdata","data",package="SimilaR"),
                                 returnType = "data.frame",
                                 fileTypes="file",
                                 aggregation = "sym")

## In this example we want to compare every pair of functions: even those
## defined in the same file. Two (non-symmetric) similarity degrees
## are reported.
SimilaR_fromDirectory(system.file("testdata","data2",package="SimilaR"),
                      returnType = "data.frame",
                      fileTypes="function",
                      aggregation = "both")
```

Index