

# Package ‘dupree’

November 10, 2019

**Type** Package

**Title** Identify Duplicated R Code in a Project

**Version** 0.2.0

**Author** Russ Hyde, University of Glasgow

**Maintainer** Russ Hyde <russ.hyde.data@gmail.com>

**Description** Identifies code blocks that have a high level of similarity within a set of R files.

**URL** <https://github.com/russHyde/dupree>

**BugReports** <https://github.com/russHyde/dupree/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat, knitr, rmarkdown, tidyr

**Imports** dplyr, purrr, tibble, magrittr, methods, stringdist (>= 0.9.5.5), lintr (>= 2.0.0), rlang

**RoxygenNote** 6.1.1

**Collate** 'dupree.R' 'dupree\_classes.R' 'dupree\_data\_validity.R' 'dupree\_code\_enumeration.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-11-10 19:00:02 UTC

## R topics documented:

dupree . . . . .	2
dupree_dir . . . . .	3
dupree_package . . . . .	4
EnumeratedCodeTable-class . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

 dupree

*Detect code duplication between the code-blocks in a set of files*


---

## Description

This function identifies all code-blocks in a set of files and then computes a similarity score between those code-blocks to help identify functions / classes that have a high level of duplication, and could possibly be refactored.

## Usage

```
dupree(files, min_block_size = 20, ...)
```

## Arguments

files	A set of files over which code-duplication should be measured.
min_block_size	'dupree' uses a notion of non-trivial symbols. These are the symbols / code-words that remain after filtering out really common symbols like '<', ',', etc. After filtering out these symbols from each code-block, only those blocks containing at least 'min_block_size' symbols are used in the inter-block code-duplication measurement.
...	Unused at present.

## Details

Code-blocks under a size threshold are disregarded before analysis (the size threshold is controlled by `min_block_size`); and only top-level code blocks are considered.

Every sufficiently large code-block in the input files will be present in the results at least once. If code-block X and code-block Y are present in a row of the resulting data-frame, then either X is the closest match to Y, or Y is the closest match to X (or possibly both) according to the similarity score; as such, some code-blocks may be present multiple times in the results.

Similarity between code-blocks is calculated using the longest-common-subsequence (lcs) measure from the package `stringdist`. This measure is applied to a tokenised version of the code-blocks. That is, each function name / operator / variable in the code blocks is converted to a unique integer so that a code-block can be represented as a vector of integers and the lcs measure is applied to each pair of these vectors.

## Value

A tibble. Each row in the table summarises the comparison between two code-blocks (block 'a' and block 'b') in the input files. Each code-block in the pair is indicated by: i) the file (`file_a` / `file_b`) that contains it; ii) its position within that file (`block_a` / `block_b`; 1 being the first code-block in a given file); and iii) the line where that code-block starts in that file (`line_a` / `line_b`). The pairs of code-blocks are ordered by decreasing similarity. Any match that is returned is either the top hit for block 'a' or for block 'b' (or both).

**Examples**

```
# To quantify duplication between the top-level code-blocks in a file
example_file <- system.file("extdata", "duplicated.R", package = "dupree")
dup <- dupree(example_file, min_block_size = 10)
dup

# For the block-pair with the highest duplication, we print the first four
# lines:
readLines(example_file)[dup$line_a[1] + c(0:3)]
readLines(example_file)[dup$line_b[1] + c(0:3)]

# The code-blocks in the example file are rather small, so if
# `min_block_size` is too large, none of the code-blocks will be analysed
# and the results will be empty:
dupree(example_file, min_block_size = 40)
```

---

dupree_dir	<i>'dupree_dir' - run duplicate-code detection over all R-files in a directory</i>
------------	--

---

**Description**

'dupree\_dir' - run duplicate-code detection over all R-files in a directory

**Usage**

```
dupree_dir(path, min_block_size = 20, filter = NULL, ...,
           recursive = TRUE)
```

**Arguments**

path	A directory. All files in this directory that have a ".R", ".r" or ".Rmd" extension will be checked for code duplication.
min_block_size	'dupree' uses a notion of non-trivial symbols. These are the symbols / code-words that remain after filtering out really common symbols like '<-', ',', etc. After filtering out these symbols from each code-block, only those blocks containing at least 'min_block_size' symbols are used in the inter-block code-duplication measurement.
filter	A pattern for use in grep - this is used to keep only particular files: eg, filter = "classes" would compare files with 'classes' in the filename
...	Further arguments for grep. For example, 'filter = "test", invert = TRUE' would disregard all files with 'test' in the file-path.
recursive	Should we consider files in subdirectories as well?

**See Also**

dupree

---

dupree_package	<i>'dupree_package' - run duplicate-code detection over all files in a package's 'R' directory</i>
----------------	--

---

**Description**

'dupree\_package' - run duplicate-code detection over all files in a package's 'R' directory

**Usage**

```
dupree_package(package, min_block_size = 20)
```

**Arguments**

package	The name or path to the package that is to be checked.
min_block_size	'dupree' uses a notion of non-trivial symbols. These are the symbols / code-words that remain after filtering out really common symbols like '<', ',', etc. After filtering out these symbols from each code-block, only those blocks containing at least 'min_block_size' symbols are used in the inter-block code-duplication measurement.

**See Also**

dupree

---

EnumeratedCodeTable-class

*An S4 class to represent the code blocks as strings of integers*

---

**Description**

An S4 class to represent the code blocks as strings of integers

**Slots**

blocks A tbl\_df with columns 'file', 'block', 'start\_line' and 'enumerated\_code'

# Index

[dupree](#), [2](#)

[dupree\\_dir](#), [3](#)

[dupree\\_package](#), [4](#)

[EnumeratedCodeTable-class](#), [4](#)