

Package ‘mindr’

December 16, 2018

Version 1.1.9

Date 2018-12-16

Title Convert Files Between Markdown or Rmarkdown Files and Mindmaps

Author Peng Zhao

Maintainer Peng Zhao <pzhao@pzhao.net>

Depends R (>= 3.0.0)

Imports htmlwidgets, knitr, jsonlite

Suggests

Description Convert Markdown (‘.md’) or Rmarkdown (‘.Rmd’) files into Free-Mind mindmap (‘.mm’) files, and vice versa. Free-Mind mindmap (‘.mm’) files can be opened by or imported to common mindmap software such as ‘Free-Mind’ (<http://freemind.sourceforge.net/wiki/index.php/Main_Page>) and ‘XMind’ (<<http://www.xmind.net>>).

License MIT + file LICENSE

URL <https://github.com/pzhaonet/mindr>

BugReports <https://github.com/pzhaonet/mindr/issues>

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-16 20:10:10 UTC

R topics documented:

dir2	2
get_body	3
get_eqloc	3
get_foldername	4
get_heading	4
get_heading2	5
get_heading3	5

markmap	6
markmapOption	7
markmapOutput	8
md2mm	8
md2r	9
mdtxt2mmtxt	10
mm2md	10
mm2r	11
outline	12
r2md	13
r2mm	13
r2rmd	14
rename2	15
renderMarkmap	15
rmd2r	16
rmvcode	16
tree2mm	17
writeLines2	18

Index 19

dir2	<i>Convert a folder structure into a mindmap.</i>
------	---

Description

Convert a folder structure into a mindmap.

Usage

```
dir2(path = getwd(), savefilename = "mindr", output = c("mm", "txt", "md"),
     backup = TRUE)
```

Arguments

path	character. the path of the folder.
savefilename	character. Valid when savefile == TRUE.
output	a file with the folder structure.
backup	logical. Whether the existing target file, if any, should be saved as backup.

Value

a mindmap file, which can be viewed by common mindmap software, such as 'FreeMind' (http://freemind.sourceforge.net/wiki/index.php/Main_Page) and 'XMind' (<http://www.xmind.net>).

Examples

```
dir2(NA)
```

get_body	<i>get the body out of given strings</i>
----------	--

Description

get the body out of given strings

Usage

```
get_body(pattern = "^#[^ ]*", text)
```

Arguments

pattern	The definition of the body text
text	the given strings

Value

integer. the index of the body text in the given strings.

get_eqloc	<i>Get the index of equations in a string vector</i>
-----------	--

Description

Get the index of equations in a string vector

Usage

```
get_eqloc(eq_begin, eq_end)
```

Arguments

eq_begin	the beginning index of an equation
eq_end	the end index of an equation

Value

a index vector

get_foldername	<i>Get the folder name of a given complete path</i>
----------------	---

Description

Get the folder name of a given complete path

Usage

```
get_foldername(path)
```

Arguments

path	The complete path
------	-------------------

Value

The folder name

get_heading	<i>get the headings out of given strings</i>
-------------	--

Description

get the headings out of given strings

Usage

```
get_heading(pattern = "^#+ ", text)
```

Arguments

pattern	The definition of the headings
text	the given strings

Value

integer. the index of the headings in the given strings.

get_heading2 *get the headings out of given strings*

Description

get the headings out of given strings

Usage

```
get_heading2(pattern = "^#= #+ ", text)
```

Arguments

pattern	The definition of the headings
text	the given strings

Value

integer. the index of the headings in the given strings.

get_heading3 *get the headings out of given strings*

Description

get the headings out of given strings

Usage

```
get_heading3(pattern = "^#' #+ ", text)
```

Arguments

pattern	The definition of the headings
text	the given strings

Value

integer. the index of the headings in the given strings.

 markmap

Create a markmap widget

Description

This function, modified from <https://github.com/seifer08ms/Rmarkmap>, creates a markmap widget using htmlwidgets. The widget can be rendered on HTML pages generated from R Markdown, Shiny, or other applications.

Usage

```
markmap(root = NA, input = c(".md", ".mm"), path = ".", remove_curly_bracket = FALSE,
        width = NULL, height = NULL, elementId = NULL, options = markmapOption(),
        bookdown_style = TRUE, method = c("regexpr", "pandoc"))
```

Arguments

root	character. a string displayed as the root of the mind map
input	character, The format of the input files
path	character. The path of the folder which contains the input file(s).
remove_curly_bracket	logical. Whether to remove #ID in the headers of the markdown file (usually in a 'bookdown' https://github.com/rstudio/bookdown project).
width	the width of the markmap
height	the height of the markmap
elementId	character.
options	the markmap options
bookdown_style	logical. whether the markdown files are in bookdown style, i.e. index.Rmd at the beginning, # (PART), # (APPENDIX) and # References as an upper level of normal # title
method	"regexpr" uses regular expressions, 'pandoc' uses pandoc to find the headings.

Value

A HTML widget object rendered from a given document.

Examples

```
path <- system.file("examples/md", package = "mindr")
markmap(path = path)
markmap(path = path, remove_curly_bracket = TRUE)
```

markmapOption	<i>Options for markmap creation</i>
---------------	-------------------------------------

Description

This function is taken from <https://github.com/seifer08ms/Rmarkmap>.

Usage

```
markmapOption(preset = NULL, nodeHeight = 20, nodeWidth = 180, spacingVertical = 10,  
              spacingHorizontal = 120, duration = 750, layout = "tree", color = "gray",  
              linkShape = "diagonal", renderer = "boxed", ...)
```

Arguments

preset	the name of built-in theme for markmap. If present, any other parameters will be ignored.
nodeHeight	the height of nodes in the markmap.
nodeWidth	the width of nodes in the markmap.
spacingVertical	space of vertical.
spacingHorizontal	space of horizontal.
duration	duration time for animation.
layout	layout mode of makrmap. Currently, only 'tree' is accepted.
color	color of markmap. A character color value ,either 'gray' or a categorical colors including 'category10','category20','category20b' and 'category20c'.
linkShape	link shape of markmap. A character value, either 'diagonal' or 'bracket'.
renderer	rendered shaped of markmap. A character value ,either 'basic' or 'boxed'.
...	other options.

Details

Currently,markmap have 'default' and 'colorful' themes. 'colorful' themes have three different parameters from default themes: nodeHeight: 10, renderer: 'basic',color: 'category20'

Functions

- markmapOption: Options for markmap creation

See Also

<https://github.com/dundalek/markmap/blob/master/lib/view.mindmap.js> for details.

Examples

```
path <- system.file('examples/md', package = 'mindr')
markmap(path = path, remove_curly_bracket = TRUE,
  options = markmapOption(preset = 'colorful')) # 'colorful' theme
markmap(path = path, remove_curly_bracket = TRUE,
  options = markmapOption(color = 'category20b',
  linkShape = 'bracket')) # 'colorful' theme
markmap(path = path, remove_curly_bracket = TRUE,
  options = markmapOption(color = 'category20b',
  linkShape = 'diagonal',
  renderer = 'basic')) # 'colorful' theme
```

markmapOutput	<i>Shiny bindings for markmap</i>
---------------	-----------------------------------

Description

Output function for using markmap within Shiny applications and interactive Rmd documents. This function is taken from <https://github.com/seifer08ms/Rmarkmap>.

Usage

```
markmapOutput(outputId, width = "100%", height = "400px")
```

Arguments

outputId	output variable to read from
width	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
height	See 'width'.

md2mm	<i>Convert markdown or rmarkdown files to mindmap files.</i>
-------	--

Description

Convert markdown or rmarkdown files to mindmap files.

Usage

```
md2mm(pattern = "*.[R]*md$", title = NA, path = ".", remove_curly_bracket = FALSE,
  savefilename = NA, backup = TRUE, bookdown_style = TRUE, keep_eq = FALSE,
  method = c("regexpr", "pandoc"))
```


Arguments

pattern	an optional regular expression for filtering the input files. See <code>help(dir)</code> .
title	character. The title of the output file.
path	character. The path of the folder which contains the input file(s).
remove_curly_bracket	logical. Whether to remove #ID in the headers of the markdown file (usually in a 'bookdown' https://github.com/rstudio/bookdown project).
savefilename	character. Valid when <code>savefile == TRUE</code> .
backup	logical. Whether the existing target file, if any, should be saved as backup.
bookdown_style	logical. whether the markdown files are in bookdown style, i.e. <code>index.Rmd</code> at the beginning, # (PART), # (APPENDIX) and # References as an upper level of normal # title
keep_eq	logical. whether to keep LaTeX equations.
method	"regexpr" uses regular expressions, 'pandoc' uses pandoc to find the headings.

Value

a mindmap file, which can be viewed by common mindmap software, such as 'FreeMind' (http://freemind.sourceforge.net/wiki/index.php/Main_Page) and 'XMind' (<http://www.xmind.net>).

Examples

```
path <- system.file("examples/md", package = "mindr")
md2mm(path = path)
md2mm(path = path, remove_curly_bracket = TRUE)
```

md2r

*Convert .md or .Rmd files into a .R script***Description**

Convert .md or .Rmd files into a .R script

Usage

```
md2r(filepattern = "*.R]*md$", path = ".", savefilename = NA, backup = TRUE,
      heading = " -----", body = "#")
```

Arguments

filepattern	the pattern of the file names
path	the path of the folder which contains the .Rmd or .md files
savefilename	the destined file name
backup	logical. whether backup the existent file
heading	the indicator of the headings
body	the indicator of the body text

Value

a .R script

Examples

```
path <- system.file("examples/md", package = "mindr")
md2r(path = path)
```

mdtxt2mmtxt

Convert markdown text to mindmap text.

Description

Convert markdown text to mindmap text.

Usage

```
mdtxt2mmtxt(title = "my title", mdtxt = "", keep_eq = FALSE)
```

Arguments

title	character. The title of the output file.
mdtxt	character. The markdown text to convert.
keep_eq	logical. whether to keep LaTeX equations.

Value

a mindmap text.

Examples

```
mdtxt2mmtxt(mdtxt = c("# Chapter 1", "## Section 1.1", "## Section 1.2"))
```

mm2md

Convert a mind map (.mm) into markdown headers.

Description

Convert a mind map (.mm) into markdown headers.

Usage

```
mm2md(pattern = "*.mm$", path = ".", savefile = TRUE, savefilename = "mindr.md",
       backup = TRUE)
```

Arguments

pattern	an optional regular expression for filtering the input files. See <code>help(dir)</code> .
path	character. The path of the folder which contains the input file(s).
savefile	logical. Whether to save the output as a markdown file.
savefilename	character. Valid when <code>savefile == TRUE</code> .
backup	logical. Whether the existing target file, if any, should be saved as backups.

Value

a vector of strings showing outline of a markdown document or book.

Examples

```
path <- system.file("examples/mm", package = "mindr")
mm2md(path = path)
```

mm2r	<i>Convert .mm into a .R script</i>
------	-------------------------------------

Description

Convert .mm into a .R script

Usage

```
mm2r(filepattern = "*.mm$", path = ".", savefilename = NA, backup = TRUE,
      heading = " -----")
```

Arguments

filepattern	the pattern of the file names
path	the path of the folder which contains the .Rmd or .md files
savefilename	the destined file name
backup	logical. whether backup the existent file
heading	the indicator of the headings

Value

a .R script

Examples

```
path <- system.file("examples/mm", package = "mindr")
mm2r(path = path)
```

 outline

Extract headers of markdown or rmarkdown files as an outline.

Description

Extract headers of markdown or rmarkdown files as an outline.

Usage

```
outline(pattern = "*.[R]*md", path = ".", remove_curly_bracket = FALSE, savefile = TRUE,
  savefilename = "outline.md", backup = TRUE, bookdown_style = TRUE, keep_eq = FALSE,
  method = c("regexpr", "pandoc"))
```

Arguments

pattern	an optional regular expression for filtering the input files. See <code>help(dir)</code> .
path	character. The path of the folder which contains the input file(s).
remove_curly_bracket	logical. Whether to remove #ID in the headers of the markdown file (usually in a 'bookdown' https://github.com/rstudio/bookdown project).
savefile	logical. Whether to save the output as a markdown file.
savefilename	character. Valid when <code>savefile == TRUE</code> .
backup	logical. Whether the existing target file, if any, should be saved as backups.
bookdown_style	logical. whether the markdown files are in bookdown style, i.e. <code>index.Rmd</code> at the beginning, # (PART), # (APPENDIX) and # References as an upper level of normal # title
keep_eq	logical. whether to keep LaTeX equations.
method	"regexpr" uses regular expressions, 'pandoc' uses pandoc to find the headings.

Value

a vector of strings showing outline of a markdown document or book.

Examples

```
path <- system.file("examples/md", package = "mindr")
outline(path = path)
outline(path = path, remove_curly_bracket = TRUE)
```

r2md *Convert .R scripts into a .md/.Rmd file*

Description

Convert .R scripts into a .md/.Rmd file

Usage

```
r2md(filepattern = "*.R$", path = ".", savefilename = NA, backup = TRUE, body = "#' ")
```

Arguments

filepattern	the pattern of the script file names
path	the path of the folder which contains the .R scripts
savefilename	the destined file name
backup	logical. whether backup the existent file
body	the indicator of the body text

Value

a markdown file

Examples

```
r2md()
```

r2mm *Convert .R scripts into a .mm file*

Description

Convert .R scripts into a .mm file

Usage

```
r2mm(filepattern = "*.R$", path = ".", title = NA, savefilename = NA)
```

Arguments

filepattern	the pattern of the script file names
path	the path of the folder which contains the .R scripts
title	title of the mindmap
savefilename	the destined file name

Value

an R markdown file

Examples

```
path <- system.file("examples/r", package = "mindr")
r2mm(path = path)
```

r2rmd

Convert .R scripts into a .Rmd file

Description

Convert .R scripts into a .Rmd file

Usage

```
r2rmd(filepattern = "*.R$", path = ".", savefilename = NA)
```

Arguments

filepattern	the pattern of the script file names
path	the path of the folder which contains the .R scripts
savefilename	the destined file name

Value

an R markdown file

Examples

```
path <- system.file("examples/r", package = "mindr")
r2rmd(path = path)
```

rename2	<i>Rename a file automatically with a time stamp</i>
---------	--

Description

Rename a file automatically with a time stamp

Usage

```
rename2(filename, connect = "-")
```

Arguments

filename	character.
connect	the connecting character in the time stamp

Value

a new file name

renderMarkmap	<i>Shiny bindings for markmap</i>
---------------	-----------------------------------

Description

Render function for using markmap within Shiny applications and interactive Rmd documents. This function is taken from <https://github.com/seifer08ms/Rmarkmap>.

Usage

```
renderMarkmap(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	An expression that generates a markmap
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

rmd2r	<i>Convert .md or .Rmd files into a .R script</i>
-------	---

Description

Convert .md or .Rmd files into a .R script

Usage

```
rmd2r(filepattern = "*.[R]*md$", path = ".", savefilename = NA, backup = TRUE,
      heading = " -----", chunkheading = FALSE)
```

Arguments

filepattern	the pattern of the file names
path	the path of the folder which contains the .Rmd or .md files
savefilename	the destined file name
backup	logical. whether backup the existent file
heading	the indicator of the headings
chunkheading	logical. whether treat chunk options as headings (ending with ---)

Value

a .R script

Examples

```
path <- system.file("examples/r", package = "mindr")
rmd2r(path = path)
```

rmvcode	<i>check whether a digital number is within a given range</i>
---------	---

Description

check whether a digital number is within a given range

Usage

```
rmvcode(index, loc)
```

Arguments

index	integer. a row number in a markdown file
loc	integer vector. the row numbers of the code block indicator, e.g. triple backticks

Value

logical.

tree2mm

Convert a directory tree to a mindmap file.

Description

Convert a directory tree to a mindmap file.

Usage

```
tree2mm(tree, savefilename = "mindr", backup = TRUE, n_root = 1)
```

Arguments

tree	character. The directory tree.
savefilename	character. Valid when savefile == TRUE.
backup	logical. Whether the existing target file, if any, should be saved as backup.
n_root	numeric. Which element is the root of the tree.

Value

a mindmap file, which can be viewed by common mindmap software, such as 'FreeMind' (http://freemind.sourceforge.net/wiki/index.php/Main_Page) and 'XMind' (<http://www.xmind.net>).

Examples

```
et2 <- c("/Root name", "/Path A", "/Path A/Product A", "/Path A/Product A/Process A",
"/Path A/Product A/Process A/Step A", "/Path A/Product A/Process A/Step A/Record 1",
"/Path A/Product A/Process A/Step A/Record 1/Analyses",
"/Path A/Product A/Process A/Step A/Record 1/Analyses/Object 1",
"/Path A/Product A/Process A/Step A/Record 1/Analyses/Object 1/Type: data source",
"/Path A/Product A/Process A/Step A/Record 1/Analyses/Object 1/Version: 3",
"/Path A/Product A/Process A/Step A/Record 1/Analyses/Object 2",
"/Path A/Product A/Process A/Step A/Record 1/Analyses/Object 3",
"/Path A/Product A/Process A/Step A/Record 1/Setup Parts",
"/Path A/Product A/Process A/Step A/Record 1/Setup Parts/Par 1",
"/Path A/Product A/Process A/Step A/Record 1/Setup Parts/Par 2",
"/Path A/Product A/Process A/Step A/Record 1/Setup Parts/Par 3",
"/Path B", "/Path C")
tree2mm(et2)
```

writeLines2	<i>Write txt files avoiding overwriting existent files.</i>
-------------	---

Description

Write txt files avoiding overwriting existent files.

Usage

```
writeLines2(text, filename, backup = TRUE)
```

Arguments

text	The text to write.
filename	The destined file name
backup	Logical.

Value

a txt file

Index

[dir2](#), [2](#)

[get_body](#), [3](#)
[get_eqloc](#), [3](#)
[get_foldername](#), [4](#)
[get_heading](#), [4](#)
[get_heading2](#), [5](#)
[get_heading3](#), [5](#)

[markmap](#), [6](#)
[markmapOption](#), [7](#)
[markmapOutput](#), [8](#)
[md2mm](#), [8](#)
[md2r](#), [9](#)
[mdtxt2mmtxt](#), [10](#)
[mm2md](#), [10](#)
[mm2r](#), [11](#)

[outline](#), [12](#)

[r2md](#), [13](#)
[r2mm](#), [13](#)
[r2rmd](#), [14](#)
[rename2](#), [15](#)
[renderMarkmap](#), [15](#)
[rmd2r](#), [16](#)
[rmvcode](#), [16](#)

[tree2mm](#), [17](#)

[writeLines2](#), [18](#)