Package 'baRcodeR'

December 7, 2019

Title Label Creation for Tracking and Collecting Data from Biological Samples

Version 0.1.4

Description Tools to generate unique identifier codes and printable barcoded labels for the management of biological samples. The creation of unique ID codes and printable PDF files can be initiated by standard commands, user prompts, or through a GUI addin for R Studio. Biologically informative codes can be included for hierarchically structured sampling designs.

License GPL-3

URL https://docs.ropensci.org/baRcodeR (website)
https://github.com/ropensci/baRcodeR

BugReports https://github.com/ropensci/baRcodeR/issues

Depends qrcode, R (>= 3.4.0)

Imports DT (>= 0.3), grDevices, grid, miniUI (>= 0.1.1), rstudioapi, shiny (>= 0.13)

Suggests covr, knitr, rmarkdown, shinytest, testthat (>= 2.1.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-CA

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Yihan Wu [aut, cre] (<https://orcid.org/0000-0002-1202-4208>), Robert Colautti [aut] (<https://orcid.org/0000-0003-4213-0711>), Emily Bao [ctb], Lluís Revilla Sancho [rev] (<https://orcid.org/0000-0001-9747-2570>), Rayna Harris [rev]

Maintainer Yihan Wu <yihan.wu@queensu.ca>

Repository CRAN

Date/Publication 2019-12-07 00:10:02 UTC

R topics documented:

`cheatsheet` *baRcodeR Cheatsheet*

Description

This addin links to a downloadable PDF version of the baRcodeR cheatsheet.

Usage

```
cheatsheet()
```

Value

Opens webpage of PDF

Examples

```
if(interactive()){
baRcodeR::cheatsheet()
}
```

`create_PDF` *Make barcodes and print labels*

Description

Input vector or data.frame of ID codes to produce a PDF of QR codes which can be printed. This is a wrapper function for `custom_create_PDF`. See details of `custom_create_PDF` on how to format text labels if needed.

Usage

```
create_PDF(user = FALSE, Labels = NULL, name = "LabelsOut",
  type = "matrix", ErrCorr = "H", Fsz = 12, ...)
```

Arguments

user	logical. Run function using interactive mode (prompts user for parameter values) Default is FALSE
Labels	vector or data frame object containing label names (i.e. unique ID codes) with either UTF-8 or ASCII encoding.
name	character. Name of the PDF output file. Default is "LabelsOut". A file named name.pdf will be saved to the working directory by default. Use "dirname/name" to produce a file called name.pdf in the dirname directory.
type	character. Choice of "linear" code 128 or "matrix" QR code labels. Default is "matrix".
ErrCorr	error correction value for matrix labels only. Level of damage from low to high: "L", "M", "Q", "H". Default is "H". See details for explanation of values.
Fsz	numerical. Sets font size in points. Longer ID codes may be shrunk to fit if truncation is not used for matrix labels. Default font size is 5. ID codes are also shrunk automatically to fit on the label if actual size is bigger than label dimensions.
...	advanced arguments to modify the PDF layout. See custom_create_PDF for arguments. The advanced options can be accessed interactively with user = TRUE and then entering TRUE when prompted to modify advanced options.

Details

The default PDF setup is for ULINE 1.75" * 0.5" WEATHER RESISTANT LABEL for laser printer; item # S-19297 (uline.ca). The page format can be modified using the ... (advanced arguments) for other label types.

Value

a PDF file containing QR-coded labels, saved to the default directory.

See Also

custom_create_PDF

Examples

```
## data frame
example_vector <- as.data.frame(c("ao1", "a02", "a03"))

## Not run:
## run with default options
## pdf file will be "example.pdf" saved into a temp directory

temp_file <- tempfile()

create_PDF(Labels = example_vector, name = temp_file)

## view example output from temp folder
```

```
system2("open", paste0(temp_file, ".pdf"))

## End(Not run)

## run interactively. Overrides default pdf options
if(interactive()){
    create_PDF(user = TRUE, Labels = example_vector)
}

## Not run:
## run using a data frame, automatically choosing the "label" column
example_df <- data.frame("level1" = c("a1", "a2"), "label" = c("a1-b1",
"a1-b2"), "level2" = c("b1", "b1"))
create_PDF(user = FALSE, Labels = example_df, name = file.path(tempdir(), "example_2"))

## End(Not run)

## Not run:
## run using an unnamed data frame
example_df <- data.frame(c("a1", "a2"), c("a1-b1", "a1-b2"), c("b1", "b1"))
## specify column from data frame
create_PDF(user = FALSE, Labels = example_df[,2], name = file.path(tempdir(), "example_3"))

## End(Not run)
## Not run:
## create linear (code128) label rather than matrix (2D/QR) labels
example_df <- data.frame(c("a1", "a2"), c("a1-b1", "a1-b2"), c("b1", "b1"))
## specify column from data frame
create_PDF(user = FALSE, Labels = example_df, name = file.path(tempdir(),
"example_4", type = "linear"))

## End(Not run)
```

`custom_create_PDF` *Make barcodes and print labels*

Description

Input a vector or data frame of ID codes to produce a PDF of barcode labels that can then be printed. The PDF setup is for the ULINE 1.75" * 0.5" WEATHER RESISTANT LABEL for laser printer; item # S-19297 (uline.ca). See details for how to format text labels properly.

Usage

```
custom_create_PDF(user = FALSE, Labels = NULL, name = "LabelsOut",
  type = "matrix", ErrCorr = "H", Fsz = 12, Across = TRUE,
  ERows = 0, ECols = 0, trunc = TRUE, numrow = 20, numcol = 4,
  page_width = 8.5, page_height = 11, width_margin = 0.25,
  height_margin = 0.5, label_width = NA, label_height = NA,
  x_space = 0, y_space = 0.5)
```

```
qrcode_make(Labels, ErrCorr)

code_128_make(Labels)
```

Arguments

user	logical. Run function using interactive mode (prompts user for parameter values) Default is FALSE
Labels	vector or data frame object containing label names (i.e. unique ID codes) with either UTF-8 or ASCII encoding.
name	character. Name of the PDF output file. Default is "LabelsOut". A file named name.pdf will be saved to the working directory by default. Use "dirname/name" to produce a file called name.pdf in the dirname directory.
type	character. Choice of "linear" code 128 or "matrix" QR code labels. Default is "matrix".
ErrCorr	error correction value for matrix labels only. Level of damage from low to high: "L", "M", "Q", "H". Default is "H". See details for explanation of values.
Fsz	numerical. Sets font size in points. Longer ID codes may be shrunk to fit if truncation is not used for matrix labels. Default font size is 5. ID codes are also shrunk automatically to fit on the label if actual size is bigger than label dimensions.
Across	logical. When TRUE, print labels across rows, left to right. When FALSE, print labels down columns, top to bottom. Default is TRUE.
ERows	number of rows to skip. Default is 0. Example: setting ERows to 6 will begin printing at row 7. ERows and ECols are useful for printing on partially-used label sheets.
ECols	number of columns to skip. Default is 0. Example: setting ECols to 2 will put the first label at column 3. ERows and ECols are useful for printing on partially-used label sheets.
trunc	logical. Text is broken into multiple lines for longer ID codes, to prevent printing off of the label area. Default is TRUE. If trunc = FALSE, and text is larger than the physical label, the text will be shrunk down automatically.
numrow	numerical. Number of rows per page. Default is 20.
numcol	numerical. Number of columns per page. Default is 4.
page_width	numerical. Width of page (in inches). Default is set to 8.5.
page_height	numerical. Height of page (in inches). Default is set to 11.
width_margin	numerical. The width margin of the page (in inches). Default is 0.25.
height_margin	numerical. The height margin of the page (in inches). Default is 0.5.
label_width	numerical. The width of label (in inches). Will be calculated as (page_width -2 * width_margin)/numcol if label_width is set as NULL.
label_height	numerical. The height of the label (in inches). Will be calculated as (page_height -2 * height_margin)/numrow if label_height is set as NULL.

x_space — numerical. A value between 0 and 1. This sets the distance between the QR code and text of each label. Only applies when type = "matrix". Default is 0.

y_space — numerical. The height position of the text on the physical label as a proportion of the label height. Only applies when type = "matrix". A value between 0 and 1. Default is 0.5.

Details

qrcode_make is the helper function for generating a QR code matrix. code_128_make is the helper function for generating a linear barcode according to code 128 set B. custom_create_PDF is the main function which sets page layout, and creates the PDF file.

Correction levels for QR codes refer to the level of damage a label can tolerate before the label become unreadable by a scanner (L = Low (7%), M = Medium (15%), Q = Quantile (25%), H = High (30%)). So a label with L correction can lose up to at most 7 while a H label can lose up to 30 can be printed at smaller sizes compared to H codes.

The escape characters \n and \s (and the hex equivalents \x0A and \x20 can be used to format text labels. Tab character \t (\x09) does not work for QR codes and should be replaced by a number of space characters. See the package vignette for examples.

If ECol or ERow is greater than numcol and numrow, the labels will be printed starting on the second page.

Value

a PDF file containing QR-coded labels, saved to the default directory.

See Also

create_PDF

Examples

```
## this is the same examples used with create_PDF
## data frame
example_vector <- as.data.frame(c("ao1", "a02", "a03"))

## Not run:
## run with default options
## pdf file will be "example.pdf" saved into a temp directory

temp_file <- tempfile()

custom_create_PDF(Labels = example_vector, name = temp_file)

## view example output from temp folder
system2("open", paste0(temp_file, ".pdf"))

## End(Not run)

## run interactively. Overrides default pdf options
```

```
if(interactive()){
    custom_create_PDF(user = TRUE, Labels = example_vector)
}

## Not run:
## run using a data frame, automatically choosing the "label" column
example_df <- data.frame("level1" = c("a1", "a2"), "label" = c("a1-b1",
"a1-b2"), "level2" = c("b1", "b1"))
custom_create_PDF(user = FALSE, Labels = example_df, name = file.path(tempdir(), "example_2"))

## End(Not run)

## Not run:
## run using an unnamed data frame
example_df <- data.frame(c("a1", "a2"), c("a1-b1", "a1-b2"), c("b1", "b1"))
## specify column from data frame
custom_create_PDF(user = FALSE, Labels = example_df[,2], name = file.path(tempdir(), "example_3"))

## End(Not run)
## Not run:
## create linear (code128) label rather than matrix (2D/QR) labels
example_df <- data.frame(c("a1", "a2"), c("a1-b1", "a1-b2"), c("b1", "b1"))
## specify column from data frame
custom_create_PDF(user = FALSE, Labels = example_df, name = file.path(tempdir(),
"example_4", type = "linear"))

## End(Not run)
```

`make_labels` *baRcodeR GUI*

Description

This addin will allow you to interactive create ID codes and generate PDF files of QR codes.

Usage

```
make_labels()
```

Value

Opens RStudio addin gadget window for making labels and barcodes in a GUI

Examples

```
if(interactive()){
library(baRcodeR)
make_labels()
}
```

uniqID_hier_maker *Make hierarchical ID codes*

Description

Generate hierarchical ID codes for barcode labels. Hierarchical codes have a nested structure: e.g. Y subsamples from each of X individuals. Use uniqID_maker for sequential single-level labels. Can be run in interactive mode, prompting user for input. The data.frame can be saved as CSV for (i) the create_PDF function to generate printable QR-coded labels; and (ii) to downstream data collection using spreadsheet, relational database, etc.

Usage

```
uniqID_hier_maker(user = FALSE, hierarchy, end = NULL, digits = 2)
```

Arguments

user	logical. Run function using interactive mode (prompts user for parameter values). Default is FALSE
hierarchy	list. A list with each element consisting of three members a vector of three elements (string, beginning value, end value). See examples. Used only when user=FALSE)
end	character. A string to be appended to the end of each label.
digits	numerical. Default is 2. Number of digits to be printed, adding leading 0s as needed. This will apply to all levels when user=FALSE. When the max number of digits in the ID code is greater than number of digits defined in digits, then digits is automatically increased to avoid errors.

Value

data.frame of text labels in the first column, with additional columns for each level in the hierarchy list, as defined by the user.

See Also

uniqID_maker

Examples

```
if(interactive()){
## for interactive mode
uniqID_hier_maker(user = TRUE)
}

## how to make hierarchy list

## create vectors for each level in the order string_prefix, beginning_value,
```

```
## end_value and combine in list

a <- c("a", 3, 6)
b <- c("b", 1, 3)
c <- list(a, b)
Labels <- uniqID_hier_maker(hierarchy = c)
Labels

## add string at end of each label
Labels <- uniqID_hier_maker(hierarchy = c, end = "end")
Labels
```

`uniqID_maker` *Generate a list of ID codes*

Description

Create ID codes consisting of a text string and unique numbers (string001, string002, ...). Can be run in interactive mode, prompting user for input. The data.frame output can be saved as CSV for (i) the `create_PDF` function to generate printable QR-coded labels; and (ii) to downstream data collection software (spreadsheets, relational databases, etc.)

Usage

```
uniqID_maker(user = FALSE, string = NULL, level, digits = 3,
  ending_string = NULL)
```

Arguments

`user`	logical. Run function using interactive mode (prompts user for parameter values). Default is `FALSE`
`string`	character. Text string for label. Default `null`.
`level`	integer vector. Defines the numerical values to be appended to the character string. Can be any sequence of numbers (see examples).
`digits`	numerical. Default is `2`. Number of digits to be printed, adding leading 0s as needed. This will apply to all levels when `user=FALSE`. When the numeric value of the label has a greater number of digits than `digits`, `digits` is automatically increased for the entire level. Default is `3`.
`ending_string`	a character string or vector of strings to attach to the label. If a vector is used, all combinations of that vector with a unique label will be produced.

Details

When the function is called with `user = TRUE`, a sequence of numbers is generated between the starting and ending number provided by the user. When `user = FALSE`, a vector of custom numbers can be provided. See example below.

Value

data.frame with text labels in the first column, along with string and numeric values in two additional columns.

See Also

uniqID_hier_maker

Examples

```
## sequential string of numbers in label
Labels <- uniqID_maker(string = "string", level = c(1:5), digits = 2)
Labels

## can also use nonsequential strings in input for levels
level <- c(1:5, 8:10, 999:1000)
Labels <- uniqID_maker(string = "string", level = level, digits = 4)
Labels


## Using the ending_string to produce labels with unique endings
## this is different from hierarchical labels with two levels as there
## is no numbering, just the text string


Labels <- uniqID_maker(string = "string", level = c(1:5), digits = 2, ending_string = "A")
Labels

Labels <- uniqID_maker(string = "string", level = c(1:5),
                       digits = 2, ending_string = c("A", "B"))
Labels


if(interactive()){
## function using user prompt does not use any of the other parameters
Labels <- uniqID_maker(user = TRUE)
Labels
}
```

Index