

Package ‘landscapetools’

October 1, 2018

Type Package

Title Landscape Utility Toolbox

Version 0.4.0

Description Provides utility functions for some of the less-glamorous tasks involved in landscape analysis. It includes functions to coerce raster data to the common tibble format and vice versa, it helps with flexible reclassification tasks of raster data and it provides a function to merge multiple raster. Furthermore, 'landscapetools' helps landscape scientists to visualize their data by providing optional themes and utility functions to plot single landscapes, rasterstacks, -bricks and lists of raster.

License GPL-3

Encoding UTF-8

LazyData true

ByteCompile true

Depends R (>= 3.1.0)

URL <https://ropensci.github.io/landscapetools/>

BugReports <https://github.com/ropensci/landscapetools/issues>

RoxygenNote 6.1.0

Imports magrittr, checkmate, dplyr, extrafont, ggplot2, purrr, raster, rasterVis, tibble, tidyr, viridis

Suggests testthat, covr

NeedsCompilation no

Author Marco Sciaini [aut, cre] (<<https://orcid.org/0000-0002-3042-5435>>),
Matthias Fritsch [aut],
Google [cph] (Roboto Condensed Font),
Laura Graham [rev] (Laura reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/188>),
Jeffrey Hollister [rev] (Jeffrey reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/188>)

Maintainer Marco Sciaini <sciaini.marco@gmail.com>

Repository CRAN

Date/Publication 2018-10-01 17:30:03 UTC

R topics documented:

landscapetools-package	2
fbmmap	3
font_rc	3
grdmap	4
rndmap	4
theme_nlm	5
util_binarize	9
util_classify	10
util_facetplot	11
util_import_roboto_condensed	12
util_merge	13
util_plot	14
util_raster2tibble	15
util_rescale	15
util_tibble2raster	16

Index	18
--------------	-----------

landscapetools-package

landscapetools

Description

landscapetools provides utility functions to work with landscape data (raster* Objects).

Author(s)

Maintainer: Marco Sciaini <sciaini.marco@gmail.com> (0000-0002-3042-5435)

Authors:

- Matthias Fritsch <matthias.fritsch@forst.uni-goettingen.de>

Other contributors:

- Google (Roboto Condensed Font) [copyright holder]
- Laura Graham (Laura reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/188>) [reviewer]
- Jeffrey Hollister (Jeffrey reviewed the package for rOpenSci, see <https://github.com/ropensci/onboarding/issues/188>) [reviewer]

See Also

Useful links:

- <https://ropensci.github.io/landscapetools/>
- Report bugs at <https://github.com/ropensci/landscapetools/issues>

fbmmap	<i>Example map (fractional brownian motion).</i>
--------	--

Description

An example map to show landscapetools functionality generated with the `nlm_fbm()` algorithm.

Usage

```
fbmmap
```

Format

A raster layer object.

Source

Simulated neutral landscape models with R. <https://github.com/ropensci/NLMR/>

font_rc	<i>Roboto Condensed font name R variable aliases</i>
---------	--

Description

```
font_rc == "Roboto Condensed"  
font_fc_light == "Roboto Condensed Light"
```

Usage

```
font_rc  
  
font_rc_light
```

Format

length 1 character vector

Note

`font_rc_light` (a.k.a. "Roboto Condensed Light") is not available on Windows and will throw a warning if used in plots.

grdmap

Example map (planar gradient).

Description

An example map to show landscapetools functionality generated with the `nlm_planargradient()` algorithm.

Usage

grdmap

Format

A raster layer object.

Source

Simulated neutral landscape models with R. <https://github.com/ropensci/NLMR/>

rndmap

Example map (random).

Description

An example map to show landscapetools functionality generated with the `nlm_random()` algorithm.

Usage

rndmap

Format

A raster layer object.

Source

Simulated neutral landscape models with R. <https://github.com/ropensci/NLMR/>

 theme_nlm

theme_nlm

Description

Opinionated ggplot2 theme to visualize NLM raster.

Usage

```
theme_nlm(base_family = "Roboto Condensed", base_size = 11.5,
  plot_title_family = base_family, plot_title_size = 18,
  plot_title_face = "bold", plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows")
    "Roboto Condensed" else "Roboto Condensed Light", subtitle_size = 13,
  subtitle_face = "plain", subtitle_margin = 15,
  strip_text_family = base_family, strip_text_size = 12,
  strip_text_face = "plain", strip.background = "grey80",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed"
    else "Roboto Condensed Light", caption_size = 9,
  caption_face = "plain", caption_margin = 10,
  axis_text_size = base_size, axis_title_family = base_family,
  axis_title_size = 9, axis_title_face = "plain",
  axis_title_just = "rt", plot_margin = ggplot2::unit(c(0, 0, 0, 0),
    "lines"), grid_col = "#cccccc", grid = TRUE, axis_col = "#cccccc",
  axis = FALSE, ticks = FALSE, legend_title = "Z",
  legend_labels = NULL, legend_text_size = 8, legend_title_size = 10,
  ratio = 1, viridis_scale = "D", ...)
```

```
theme_nlm_discrete(base_family = "Roboto Condensed", base_size = 11.5,
  plot_title_family = base_family, plot_title_size = 18,
  plot_title_face = "bold", plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows")
    "Roboto Condensed" else "Roboto Condensed Light", subtitle_size = 13,
  subtitle_face = "plain", subtitle_margin = 15,
  strip_text_family = base_family, strip_text_size = 12,
  strip_text_face = "plain", strip.background = "grey80",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed"
    else "Roboto Condensed Light", caption_size = 9,
  caption_face = "plain", caption_margin = 10,
  axis_text_size = base_size, axis_title_family = base_family,
  axis_title_size = 9, axis_title_face = "plain",
  axis_title_just = "rt", plot_margin = ggplot2::unit(c(0, 0, 0, 0),
    "lines"), grid_col = "#cccccc", grid = TRUE, axis_col = "#cccccc",
  axis = FALSE, ticks = FALSE, legend_title = "Z",
  legend_labels = NULL, legend_text_size = 8, legend_title_size = 10,
  ratio = 1, viridis_scale = "D", ...)
```

```

theme_nlm_grey(base_family = "Roboto Condensed", base_size = 11.5,
  plot_title_family = base_family, plot_title_size = 18,
  plot_title_face = "bold", plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows")
  "Roboto Condensed" else "Roboto Condensed Light", subtitle_size = 13,
  subtitle_face = "plain", subtitle_margin = 15,
  strip_text_family = base_family, strip_text_size = 12,
  strip_text_face = "plain", strip.background = "grey80",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed"
  else "Roboto Condensed Light", caption_size = 9,
  caption_face = "plain", caption_margin = 10,
  axis_text_size = base_size, axis_title_family = base_family,
  axis_title_size = 9, axis_title_face = "plain",
  axis_title_just = "rt", plot_margin = ggplot2::unit(c(0, 0, 0, 0),
  "lines"), grid_col = "#cccccc", grid = TRUE, axis_col = "#cccccc",
  axis = FALSE, ticks = FALSE, legend_title = "Z",
  legend_labels = NULL, legend_text_size = 8, legend_title_size = 10,
  ratio = 1, ...)

theme_nlm_grey_discrete(base_family = "Roboto Condensed",
  base_size = 11.5, plot_title_family = base_family,
  plot_title_size = 18, plot_title_face = "bold",
  plot_title_margin = 10, subtitle_family = if (.Platform$OS.type ==
  "windows") "Roboto Condensed" else "Roboto Condensed Light",
  subtitle_size = 13, subtitle_face = "plain", subtitle_margin = 15,
  strip_text_family = base_family, strip_text_size = 12,
  strip_text_face = "plain", strip.background = "grey80",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed"
  else "Roboto Condensed Light", caption_size = 9,
  caption_face = "plain", caption_margin = 10,
  axis_text_size = base_size, axis_title_family = base_family,
  axis_title_size = 9, axis_title_face = "plain",
  axis_title_just = "rt", plot_margin = ggplot2::unit(c(0, 0, 0, 0),
  "lines"), grid_col = "#cccccc", grid = TRUE, axis_col = "#cccccc",
  axis = FALSE, ticks = FALSE, legend_title = "Z",
  legend_labels = NULL, legend_text_size = 8, legend_title_size = 10,
  ratio = 1, ...)

theme_facetplot(base_family = "Roboto Condensed", base_size = 11.5,
  plot_title_family = base_family, plot_title_size = 18,
  plot_title_face = "bold", plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows")
  "Roboto Condensed" else "Roboto Condensed Light", subtitle_size = 13,
  subtitle_face = "plain", subtitle_margin = 15,
  strip.background = "grey80", caption_family = if (.Platform$OS.type
  == "windows") "Roboto Condensed" else "Roboto Condensed Light",
  caption_size = 9, caption_face = "plain", caption_margin = 10,
  ratio = 1, viridis_scale = "D", ...)

```

Arguments

base_family	base font family size
base_size	base font size
plot_title_family	plot title family
plot_title_size	plot title size
plot_title_face	plot title face
plot_title_margin	plot title ggplot2::margin
subtitle_family	plot subtitle family
subtitle_size	plot subtitle size
subtitle_face	plot subtitle face
subtitle_margin	plot subtitle ggplot2::margin bottom (single numeric value)
strip_text_family	facet facet label font family
strip_text_size	facet label font family, face and size
strip_text_face	facet facet label font face
strip.background	strip background
caption_family	plot caption family
caption_size	plot caption size
caption_face	plot caption face
caption_margin	plot caption ggplot2::margin
axis_text_size	axis text size
axis_title_family	axis title family
axis_title_size	axis title size
axis_title_face	axis title face
axis_title_just	axis title justification
plot_margin	plot ggplot2::margin (specify with 'ggplot2::margin')
grid_col	grid color
grid	grid TRUE/FALSE
axis_col	axis color

axis	axis TRUE/FALSE
ticks	ticks TRUE/FALSE
legend_title	Title of the legend (default "Z")
legend_labels	Labels for the legend ticks, if used with <code>util_plot</code> they are automatically derived.
legend_text_size	legend text size, default 8
legend_title_size	legend text size, default 10
ratio	ratio for tiles (default 1, if your raster is not a square the ratio should be <code>raster::nrow(x) / raster::ncol(x)</code>)
viridis_scale	Five options are available: "viridis - magma" (= "A"), "viridis - inferno" (= "B"), "viridis - plasma" (= "C"), "viridis - viridis" (= "D", the default option), "viridis - cividis" (= "E")
...	optional arguments to <code>ggplot2::theme</code>

Details

A focused theme to visualize raster data that sets a lot of defaults for the `ggplot2::theme`.

The theme can make use of the Roboto Condensed font (Open Source font from Google). If your local font library does not contain Roboto as a font, you can import it via `util_import_roboto_condensed` (highly recommended).

The functions are setup in such a way that you can customize your own one by just wrapping the call and changing the parameters. The theme itself is heavily influenced by `hrbrmstr` and his package `hrbrthemes` (<https://github.com/hrbrmstr/hrbrthemes/>).

See Also

`util_import_roboto_condensed`

Examples

```
# nolint start
## Not run:
# provided example map
x <- fbmmmap
# classify
y <- c(0.5, 0.15, 0.25)
y <- util_classify(x, y, c("1", "2", "3"))

# color + continuous
rasterVis::gplot(x) +
  ggplot2::geom_tile(ggplot2::aes(fill = value)) +
  ggplot2::labs(x = "Easting",
               y = "Northing") +
  theme_nlm() +
  ggplot2::ggtitle("Example map",
                  subtitle = "with continuous viridis color scale") +
  ggplot2::labs(caption = "Example map simulated with the R package NLMR.")
```



```

# grey + continuous
rasterVis::gplot(x) +
  ggplot2::geom_tile(ggplot2::aes(fill = value)) +
  ggplot2::labs(x = "Easting",
                y = "Northing") +
  theme_nlm_grey() +
  ggplot2::ggtitle("Example map",
                   subtitle = "with continuous grey color scale") +
  ggplot2::labs(caption = "Example map simulated with the R package NLMR.")

# color + discrete
rasterVis::gplot(y) +
  ggplot2::geom_tile(ggplot2::aes(fill = factor(value))) +
  ggplot2::labs(x = "Easting",
                y = "Northing") +
  theme_nlm_discrete() +
  ggplot2::ggtitle("Example map",
                   subtitle = "with discrete viridis color scale") +
  ggplot2::labs(caption = "Random map simulated with the R package NLMR.")

# grey + discrete
rasterVis::gplot(y) +
  ggplot2::geom_tile(ggplot2::aes(fill = factor(value))) +
  ggplot2::labs(x = "Easting",
                y = "Northing") +
  theme_nlm_grey_discrete() +
  ggplot2::ggtitle("Example map",
                   subtitle = "with discrete grey color scale") +
  ggplot2::labs(caption = "Random map simulated with the R package NLMR.")

# have a look at theme_facetplot
binary_maps <- util_binarize(x, c(0.3, 0.5, 0.7, 0.9))
util_facetplot(binary_maps)

# nolint end

## End(Not run)

```

util_binarize

Binarize continuous raster values

Description

Classify continuous raster values into binary map cells based upon given break(s).

Usage

```
util_binarize(x, breaks)
```

Arguments

x	Raster* object
breaks	Vector with one or more break percentages

Details

Breaks are considered to be habitat percentages (p). If more than one percentage is given multiple layers are written in the same brick.

Value

RasterLayer / RasterBrick

Examples

```
breaks <- c(0.3, 0.5)
(binary_maps <- util_binarize(fbmap, breaks))
```

util_classify

util_classify

Description

Classify a raster into proportions based upon a vector of class weightings.

Usage

```
util_classify(x, weighting, level_names = NULL, real_land = NULL,
             mask_val = NULL)
```

Arguments

x	2D matrix
weighting	Vector of numeric values.
level_names	Vector of names for the factor levels.
real_land	Raster with real landscape
mask_val	Value to mask (refers to real_land)

Details

The number of elements in the weighting vector determines the number of classes in the resulting matrix. The classes start with the value 0. If non-numerical levels are required, the user can specify a vector to turn the numerical factors into other data types, for example into character strings (i.e. class labels). If the numerical vector of weightings does not sum up to 1, the sum of the weightings is divided by the number of elements in the weightings vector and this is then used for the classification.

For a given 'real' landscape the number of classes and the weightings are extracted and used to classify the given nlm landscape (any given weighting parameter is overwritten in this case!). If an optional mask value is given the corresponding class from the 'real' landscape is cut from the nlm landscape beforehand.

Value

RasterLayer

Examples

```
weight <- c(0.5, 0.25, 0.25)
util_classify(fbmap, weight,
             level_names = c("Land Use 1", "Land Use 2", "Land Use 3"))

## Not run:
rland <- util_classify(NLMR::nlm_planargradient(200,200),
                    c(.4,.2,.4),
                    c("Land use 1", "Water", "Land use 2"))

resu <- util_classify(fbmap, real_land = rland)
resu_mask <- util_classify(fbmap, real_land = rland, mask_val = 1)

visu <- list(
'1 nlm' = fbmap,
'2 real' = rland,
'3 result' = resu,
'4 result with mask' = resu_mask
)
util_facetplot(visu)

## End(Not run)
```

util_facetplot

util_facetplot() for visual overview

Description

Plot multiple raster (RasterStack, -brick or list of raster) side by side as facets.

Usage

```
util_facetplot(mpdata, nrow = NULL, ncol = NULL)
```

Arguments

`mpdata` Raster* Layer, Stack, Brick or a list of rasterLayers.
`nrow, ncol` Number of rows and columns.

Details

The output uses ggplots faceting and beforehand raster2tibble. Thus you will loose any spatial information (resolution, extent or reference system). Only raw tiles are displayed and the number of cells determines the size of the plot. This can lead to huge size differences between maps, but if you plot for example multiple maps from a time series side by side it works as intended. Depending on the size of the maps it is advisable to store the plot in an object and print it to a file. This will help with compressing and rendering the image.

Value

ggplot

Examples

```
## Not run:  
l1 <- NLMR::nlm_fbm(64, 64)  
l2 <- NLMR::nlm_planargradient(64, 64)  
l3 <- NLMR::nlm_randomrectangularcluster(ncol = 60, nrow = 60, minl = 5, maxl = 10)  
l4 <- NLMR::nlm_random(64, 64)  
  
bri1 <- raster::brick(l1, l2)  
names(bri1) <- c("FBM", "GRADIENT")  
util_facetplot(bri1)  
  
lst1 <- list(layer1 = l1,  
            layer2 = l2,  
            layer3 = l3,  
            layer4 = l4)  
util_facetplot(lst1)  
  
## End(Not run)
```

util_import_roboto_condensed

Import Roboto Condensed font for use in charts

Description

Roboto Condensed is a trademark of Google.

Usage

```
util_import_roboto_condensed()
```

Note

This will take care of ensuring PDF/PostScript usage. The location of the font directory is displayed after the base import is complete. It is highly recommended that you install them on your system the same way you would any other font you wish to use in other programs.

util_merge

util_merge

Description

Merge a primary raster with other rasters weighted by scaling factors.

Usage

```
util_merge(primary_nlm, secondary_nlm, scalingfactor = 1,  
           rescale = TRUE)
```

Arguments

primary_nlm	Primary Raster* object
secondary_nlm	A list or stack of Raster* objects that are merged with the primary Raster* object
scalingfactor	Weight for the secondary Raster* objects
rescale	If TRUE (default), the values are rescaled between 0-1.

Value

Rectangular matrix with values ranging from 0-1

Examples

```
(util_merge(grdmap, rndmap))
```

`util_plot`*util_plot*

Description

Plot a Raster* object with the NLMR default theme (as ggplot).

Usage

```
util_plot(x, xlab = "Easting", ylab = "Northing", discrete = FALSE,  
  ...)
```

```
util_plot_grey(x, xlab = "Easting", ylab = "Northing",  
  discrete = FALSE, ...)
```

Arguments

<code>x</code>	Raster* object
<code>xlab</code>	x axis label, default "Easting"
<code>ylab</code>	y axis label, default "Northing"
<code>discrete</code>	If TRUE, the function plots a raster with a discrete legend.
<code>...</code>	Arguments for theme_nlm

Value

ggplot2 Object

Examples

```
## Not run:  
x <- grdmap  
# classify  
y <- c(0.5, 0.15, 0.25)  
y <- util_classify(x, y, c("1", "2", "3"))  
  
util_plot(x)  
util_plot(y, discrete = TRUE)  
  
util_plot_grey(x)  
util_plot_grey(y, discrete = TRUE)  
  
## End(Not run)
```

util_raster2tibble *Converts raster data into tibble*

Description

Writes spatial raster values into tibble and adds coordinates.

Usage

```
util_raster2tibble(x)
```

Arguments

x Raster* object

Details

You will loose any resolution, extent or reference system. The output is raw tiles.

Value

a tibble

Examples

```
maptib <- util_raster2tibble(fbmmmap)
## Not run:
library(ggplot2)
ggplot(maptib, aes(x,y)) +
  coord_fixed() +
  geom_raster(aes(fill = z))

## End(Not run)
```

util_rescale *util_rescale*

Description

Linearly rescale element values in a raster to a range between 0 and 1.

Usage

```
util_rescale(x)
```

Arguments

x Raster* object

Details

Rasters generated by nlm_ functions are scaled between 0 and 1 as default, this option can be set to FALSE if needed.

Value

Raster* object with values ranging from 0-1

Examples

```
unscmap <- util_merge(grdmap, rndmap, rescale = FALSE)
util_rescale(unscmap)
```

util_tibble2raster *Converts tibble data into a raster*

Description

Writes spatial tibble values into a raster.

Usage

```
util_tibble2raster(x)
```

Arguments

x a tibble

Details

Writes tiles with coordinates from a tibble into a raster. Resolution is set to 1 and the extent will be c(0, max(x), 0, max(y)).

You can directly convert back the result from 'util_raster2tibble()' without problems. If you have altered the coordinates or otherwise played with the data, be careful while using this function.

Value

Raster* object

Examples

```
maptib <- util_raster2tibble(rndmap)
mapras <- util_tibble2raster(maptib)
all.equal(rndmap, mapras)
```

Index

*Topic **datasets**

- fbmmap, 3
- font_rc, 3
- grdmap, 4
- rndmap, 4

fbmmap, 3
font_rc, 3
font_rc_light (font_rc), 3

grdmap, 4

landscapetools
 (landscapetools-package), 2
landscapetools-package, 2

rndmap, 4

theme_facetplot (theme_nlm), 5
theme_nlm, 5, 14
theme_nlm_discrete (theme_nlm), 5
theme_nlm_grey (theme_nlm), 5
theme_nlm_grey_discrete (theme_nlm), 5

util_binarize, 9
util_classify, 10
util_facetplot, 11
util_import_roboto_condensed, 8, 12
util_merge, 13
util_plot, 8, 14
util_plot_grey (util_plot), 14
util_raster2tibble, 15
util_rescale, 15
util_tibble2raster, 16