

Package: cmocean (via r-universe)

September 20, 2026

Encoding UTF-8
Version 0.3-2.9999
Title Beautiful Colour Maps for Oceanography
Imports grDevices
Depends R (>= 3.1.0)
Suggests knitr, rmarkdown
Enhances ggplot2
Description Perceptually uniform palettes for commonly used variables in oceanography as functions taking an integer and producing character vectors of colours. See Thyng, K.M., Greene, C.A., Hetland, R.D., Zimmerle, H.M. and S.F. DiMarco (2016) <doi:10.5670/oceanog.2016.66> for the guidelines adhered to when creating the palettes.
License MIT + file LICENSE
URL <https://matplotlib.org/cmocean/>
BugReports <https://github.com/aitap/cmocean/issues>
VignetteBuilder knitr, rmarkdown
Repository <https://aitap.r-universe.dev>
Date/Publication 2025-07-27 21:02:46 UTC
RemoteUrl <https://github.com/aitap/cmocean>
RemoteRef HEAD
RemoteSha 15ba18b5d6ccbc78c4fa7ec7a19532aeede81cd0

Contents

cmocean-package	2
cmocean	2
scale_colour_cmocean	5
Index	7

cmocean-package	<i>Beautiful Colour Maps for Oceanography</i>
-----------------	---

Description

Perceptually uniform palettes for commonly used variables in oceanography as functions taking an integer and producing character vectors of colours. See Thyng, K.M., Greene, C.A., Hetland, R.D., Zimmerle, H.M. and S.F. DiMarco (2016) <doi:10.5670/oceanog.2016.66> for the guidelines adhered to when creating the palettes.

References

Thyng, K. M., Greene, C. A., Hetland, R. D., Zimmerle, H. M., and DiMarco, S. F. (2016). True colors of oceanography. *Oceanography*, 29(3), 10.

See Also

[cmocean](#) for the classic-style palette-generating function, [scale_colour_cmocean](#) for **ggplot2** helpers.

cmocean	<i>Return a cmocean palette function</i>
---------	--

Description

Produce a palette function for a palette described by the first paramter.

Usage

```
cmocean(
  name, version = getOption("cmocean-version", "2.0"),
  clip = 0, start = clip/2, end = 1 - clip/2,
  direction = 1, alpha = 1
)
```

Arguments

name	the name of the colormap to return, see Details for possible values.
version	specific cmocean version you would like to use; one of: "0.03", "0.1", "0.2", "0.2.1", "0.2.2", "0.3", "1.0", "1.1", "1.2", "2.0". Alternatively, you can use <code>options(`cmocean-version` = "...")</code> to set the desired version as a global option. Explicitly set argument overrides the global option; the global option overrides the default of the latest version stored in the package.
clip	fraction of the colormap to clip as a number between 0 and 1 (divided equally among both sides).

<code>start</code>	fraction of the colormap to clip from its start (overrides <code>clip</code>).
<code>end</code>	fraction of the colormap where it should stop, in $[0, 1]$ (overrides <code>clip</code> ; <code>start < end</code> .)
<code>direction</code>	set to -1 to reverse the palette.
<code>alpha</code>	opacity of the palette in the range $[0, 1]$.

Details

Names of available palettes depend on the version you would like to use:

thermal (≥ 1.0), Temperature (< 1.0) Sequential with dark blue representing lower, cooler values and transitioning through reds to yellow representing increased warmer values.

haline (≥ 1.0), Salinity (< 1.0) Sequential, and might be used with dark blue representing lower salinity or fresher water, transitioning through greens to light yellow representing increased salinity or saltier water. This colormap is based on matplotlib's "YlGnBu", but was recreated from scratch using the 'viscm' tool.

solar (≥ 1.0), PAR (< 1.0) Sequential from dark brown for low values to increasingly bright yellow to potentially represent an increase in radiation in the water.

gray (≥ 1.0), Gray (< 1.0 , inverted) Sequential from black to white, with uniform steps through perceptual colorspace. This colormap is generic to be used for any sequential dataset.

oxy (≥ 1.0), Oxygen (< 1.0) Sequential for most of the colormap, representing the normal range of oxygen saturation in ocean water, and diverging 80% of the way into the colormap to represent a state of supersaturation. The bottom 20% of the colormap is colored reddish to highlight hypoxic or low oxygen water, but to still print relatively seamlessly into grayscale in case the red hue is not important for an application. The top 20% of the colormap, after the divergence, is colored yellow to highlight the supersaturated water.

deep (≥ 1.0), Bathymetry (< 1.0) Sequential from light yellow to potentially represent shallower water through pale green to increasingly dark blue and purple to represent increasing depth.

dense (≥ 1.0), Density (< 1.0) Sequential with whitish-blue for low values and increasing in purple with increasing value, which could be used to represent an increase in water density.

algae (≥ 1.0), Chlorophyll (< 1.0) Sequential with whitish-green for low values and increasing in green with increasing value, which could be used to represent an increase in chlorophyll in the water.

matter (≥ 1.0), CDOM (< 1.0) Sequential with whitish-yellow for low values and increasing in pink with increasing value, and could be used to represent an increase in material in the water.

turbid (≥ 1.0), Turbidity (< 1.0) Sequential from light to dark brown and could be used to represent an increase in sediment in the water.

speed (≥ 1.0), Speed (< 1.0) Sequential from light greenish yellow representing low values to dark yellowish green representing large values. This colormap is the positive half of the delta colormap.

amp (≥ 1.0), WaveHeight (< 1.0) Sequential from whitish to dark red and could be used to represent an increase in wave height values. This colormap is the positive half of the balance colormap.

tempo (≥ 1.0), WavePeriod (< 1.0) Sequential from whitish to dark teal and could be used to represent an increase in wave period values. This colormap is the negative half of the `curl` colormap.

phase (≥ 1.0), Phase (< 1.0) Circular, spanning all hues at a set lightness value. This map is intended to be used for properties such as wave phase and tidal phase which wrap around from 0° to 360° to 0° and should be represented without major perceptual jumps in the colormap.

balance (≥ 1.0), FreeSurface (< 1.0) Diverging with dark blue to off-white to dark red representing negative to zero to positive values; this could be used to represent sea surface elevation, with deviations in the surface elevations as shades of color away from neutral off-white. In this case, shades of red have been chosen to represent sea surface elevation above the reference value (often mean sea level) to connect with warmer water typically being associated with an increase in the free surface.

delta (≥ 1.0), Velocity (< 1.0) Diverging from darker blues to just off-white through shades of yellow green and could be used to represent diverging velocity values around a critical value (usually zero). This colormap was inspired by Francesca Samsel's similar colormap, but generated from scratch using the 'viscm' tool.

curl (≥ 1.0), Vorticity (< 1.0) Diverging from darker teal to just off-white through shades of magenta and could be used to represent diverging vorticity values around a critical value (usually zero).

ice (≥ 1.0) Sequential from very dark blue (almost black) to very light blue (almost white). A use for this could be representations of sea ice.

rain (≥ 2.0) Sequential from light, dry colors to blue, wet colors, and could be used to plot amounts of rainfall.

topo (≥ 2.0) The topo colormap has two distinct parts: one that is shades of blue and yellow to represent water depths (this is the deep colormap) and one that is shades of browns and greens to represent land elevation.

diff (≥ 2.0) Diverging, with one side shades of blues and one side shades of browns.

tarn (≥ 2.0) Diverging, with one side dry shades of browns and the other a range of greens and blues. The positive end of the colormap is meant to reflect the colors in rain and thus be a complementary colormap to rain for rain anomaly (around 0 or some other critical value).

Value

A function taking an integer and returning a character vector with '#RRGGBB' colours. Currently, it does not make sense to request more than 256 of them.

Examples

```
image(volcano, col = cmocean('thermal')(100))
image(volcano, col = cmocean('Temperature', '0.03')(128))
image(volcano, col = cmocean('thermal', clip = .2)(256))
image(volcano, col = cmocean('thermal', start = .1, end = .7)(256))
image(volcano, col = cmocean('thermal', direction = -1)(256))
image(volcano, col = cmocean('thermal', alpha = .5)(256))
```

scale_colour_cmocean *cmocean colour scales for ggplot2*

Description

Helper functions to allow the colour scale to be used effectively in **ggplot2**.

Usage

```
scale_colour_cmocean(  
  ..., alpha = 1, start = 0, end = 1, direction = 1, discrete = FALSE,  
  name = "thermal", version = getOption('cmocean-version', '2.0')  
)  
scale_color_cmocean(  
  ..., alpha = 1, start = 0, end = 1, direction = 1, discrete = FALSE,  
  name = "thermal", version = getOption('cmocean-version', '2.0')  
)  
scale_fill_cmocean(  
  ..., alpha = 1, start = 0, end = 1, direction = 1, discrete = FALSE,  
  name = "thermal", version = getOption('cmocean-version', '2.0')  
)
```

Arguments

...	parameters to ggplot2 functions <code>discrete_scale</code> or <code>scale_fill_gradientn</code> .
alpha	opacity of the palette in the range $[0, 1]$.
start	fraction of the colormap to clip from its start, in $[0, 1]$.
end	fraction of the colormap where it should stop, in $[0, 1]$; <code>start < end</code> .
direction	set to <code>-1</code> to reverse the palette.
discrete	set to <code>TRUE</code> to generate a discrete palette instead of a continuous one.
name	the name of one of the available cmocean colormaps, see cmocean for the full list.
version	specific cmocean version you would like to use, passed as-is to cmocean . Defaults to the latest available version of the palette set.

Value

A ggplot object to be used with other ggplot objects.

Author(s)

As requested by Ilja Kocken in <https://github.com/aitap/cmocean/issues/5>.

See Also

[cmocean](#)

Examples

```
if (require('ggplot2')) {  
  dat <- data.frame(  
    a = 1:10, b = 11:20, c = rnorm(10)  
  )  
  
  ggplot(dat, aes(x = a, y = b, fill = c)) +  
    geom_raster() + scale_fill_cmocean()  
}
```

Index

* **color**

cmocean, [2](#)

cmocean-package, [2](#)

cmocean, [2](#), [2](#), [5](#)

cmocean-package, [2](#)

scale_color_cmocean

(scale_colour_cmocean), [5](#)

scale_colour_cmocean, [2](#), [5](#)

scale_fill_cmocean

(scale_colour_cmocean), [5](#)