

- ✓ Perceptually uniform
- ✓ Perceptually ordered
- ✓ Colour-vision-deficiency (CVD) friendly
- ✓ Readable as black and white print
- ✓ Provided in all major formats
- ✓ Citable & reproducible
- \* Cyclic
- \* No white; no black

Crameri, F. (2018), Scientific colour-maps, Zenodo, [doi:10.5281/zenodo.1243862](https://doi.org/10.5281/zenodo.1243862)

Crameri, F. (2018), Geodynamic diagnostics, scientific visualisation and StagLab 3.0, Geosci. Model Dev., 11, 2541-2562, [doi:10.5194/gmd-11-2541-2018](https://doi.org/10.5194/gmd-11-2541-2018)

Crameri, F., G.E. Shephard, and P.J. Heron (2020), The misuse of colour in science communication, Nature Communications, 11, 5444. [doi: 10.1038/s41467-020-19160-7](https://doi.org/10.1038/s41467-020-19160-7)

---

\* [www.fabiocrameri.ch/romaO](http://www.fabiocrameri.ch/romaO)



# Contents

<b>1</b>	<b>Creators</b>	<b>4</b>
<b>2</b>	<b>Sources of inspiration</b>	<b>5</b>
<b>3</b>	<b>Acknowledgement</b>	<b>5</b>
<b>4</b>	<b>Instructions</b>	<b>6</b>
4.1	ArcGIS Pro . . . . .	6
4.2	d3 . . . . .	6
4.3	Ferret . . . . .	6
4.4	Fledermaus & Qimera . . . . .	6
4.5	GIMP/Inkscape . . . . .	6
4.6	GMT . . . . .	6
4.7	Gnuplot . . . . .	7
4.8	ImageJ/Fiji . . . . .	7
4.9	Kingdom . . . . .	7
4.10	Mathematica . . . . .	7
4.11	MatLab . . . . .	8
4.12	Ncview . . . . .	9
4.13	Originlab . . . . .	9
4.14	Paraview . . . . .	9
4.15	Petrel . . . . .	9
4.16	Photoshop . . . . .	10
4.17	Plotly . . . . .	10
4.18	Python . . . . .	10
4.19	QGIS . . . . .	11
4.20	R . . . . .	12
4.21	SKUA-GOCAD . . . . .	12
4.22	VisIt . . . . .	12
<b>5</b>	<b>Software with built-in versions</b>	<b>12</b>
<b>6</b>	<b>References</b>	<b>13</b>
<b>7</b>	<b>Version history</b>	<b>13</b>
<b>8</b>	<b>License</b>	<b>15</b>

# 1 Creators

Author: [Fabio Crameri](#)

Contributors: [Grace Shephard](#) – Conversion to .cpt format, promotion of Scientific colour maps, and other contributions

[Phil Heron](#) – Promotion of Scientific colour maps

[Clint Conrad](#) – Wider compatibility of .cpt format

[Matteo Albano](#) – Conversion to .clr format; QGIS compatibility

[Casper Pranger](#) – Mathematica compatibility

[Alexis Plunder](#) – Wider compatibility of .xml format

[Krister Stræte Karlsen](#) – User instruction for use with python

[Philippe Rivière](#) – Conversion instruction for d3

[Emilia](#) – Plotly versions

[Thomas Lin Pedersen](#) – The ‘scico’ package for use with R

[Paul Wessel](#) – Built-in version for ‘GMT’

[Wolfgang Schwanghart](#) – Built-in version for ‘TopoToolbox’

[Chad Greene](#) – MatLab file exchange version

[Sean Trim](#) – Conversion to .pal format

[George Edward Campbell](#) – Conversion to .lut format

[Christophe Leterrier](#) – NeuroCyto LUTs Fiji add-on

[Kirstie Wright](#) – User instruction for use with Petrel

[Craig Williams](#) – Style file for ArcGIS Pro

[Jennifer Levett](#) – Conversion to SKUA-GOCAD .xcmap format

[Sam Hatfield](#) – Conversion to Ncview .ncmap format

[Patrick Brockmann](#) – Conversion to Ferret .spk format

[Thomas Morrow](#) – Conversion to QPS .cmap format

[Mark Wieczorek](#) – Import init file for Python

[Anthony Jamelot](#) – Additions to import init file for Python

[Andy Emery](#) – Conversion to Kingdom .clm and .clb format

[Benjamin Witschas](#) – Conversion to Originlab .pal format

[Steven Reddy](#) – Conversion to Photoshop .grd format

[Callum Rollo](#) – Python package via pip and anaconda

## 2 Sources of inspiration

The ‘**endrainbow**’ campaign initiated by Ed Hawkins.

The **Colorbrewer** colour maps, the **MPL** colour maps, the **cividis** colour map, the **CMOcean** colour maps, and the **CET** colour maps.

Peter **Kovesi**’s work, in particular, has helped to develop the Scientific Colour Maps: some of the **many excellent, openly accessible scripts** were used as a basis for the applied colour-map diagnostics and to make the file conversion to .tbl and .act formats.

## 3 Acknowledgement

! → Please acknowledge the free use of the colour maps.

e.g., *"The scientific colour map romaO (Crameri 2018) is used in this study to prevent visual distortion of the data and exclusion of readers with colour-vision deficiencies (Crameri et al., 2020)."*

The software : **Crameri, F. (2018a), Scientific colour-maps. Zenodo. <http://doi.org/10.5281/zenodo.1243862>**

The research : **Crameri, F., G.E. Shephard, and P.J. Heron (2020), The misuse of colour in science communication, Nature Communications, 11, 5444. doi: [10.1038/s41467-020-19160-7](https://doi.org/10.1038/s41467-020-19160-7)**

## 4 Instructions

### 4.1 ArcGIS Pro

#### 4.1.1 style file

Download the style file for the ArcGIS Pro provided by [Craig Williams](#) on <https://www.arcgis.com/home/>.

### 4.2 d3

#### 4.2.1 .xml format

An instruction to convert the .xml format to d3's internal representation is provided by Philippe Rivière at <https://beta.observablehq.com/@fil/colormaps>.

### 4.3 Ferret

#### 4.3.1 .spk format

To use the .spk colour map files in Ferret, follow the instructions given on the official homepage: [https://ferret.pmel.noaa.gov/Ferret/documentation/users-guide/customizing-plots/COLOR#\\_VPID\\_247](https://ferret.pmel.noaa.gov/Ferret/documentation/users-guide/customizing-plots/COLOR#_VPID_247).

### 4.4 Fledermaus & Qimera

#### 4.4.1 .cmap format

To use the .cmap colour map files in the QPS software Fledermaus and Qimera, download the external package from [www.fabiocrameri.ch/colourmaps](http://www.fabiocrameri.ch/colourmaps) or via the direct [link](#).

### 4.5 GIMP/Inkscape

#### 4.5.1 .gpl format

To import the .gpl palettes, launch GIMP and go to **Windows > Dockable Dialogs > Palettes** to open the Palettes dialog. Then right-click anywhere on the list of palettes and select **Import Palette**. In the *\*Import a New Palette\** dialog, select the *\*Palette file\** radio button and then the button just to the right of the folder icon.

Then, navigate to and select the desired .gpl file in the corresponding folder. Clicking the *\*Import\** button will add the scientific colour map to the existing list of palettes.

### 4.6 GMT

**Note:** GMT 6.0.0 and later offers built-in scientific colour maps (see Section 5).

#### 4.6.1 .cpt format

The file `davos.cpt` can be resampled for a given z-value range with the Generic Mapping Tools (GMT; <http://www.generic-mapping-tools.org>) command `"makecpt"`.

For example to resample for an array from -2000 to 2000 in 100 increments you could generate a new file with:

```
$makecpt -Cdavos.cpt -T-2000/2000/100 > davos_resampled.cpt
```

## 4.7 Gnuplot

### 4.7.1 .pal format

Launch the Gnuplot shell and load the specific .pal file (e.g., batlow) into Gnuplot with:

```
user@computer gnuplot
gnuplot> load "batlow.pal"
```

## 4.8 ImageJ/Fiji

### 4.8.1 .lut format

The .lut colour-map file (e.g., \*batlow.lut\*) can be imported to ImageJ or Fiji by placing it in the \*luts\* folder (to reveal folder location in Fiji: **File > Show Folder > LUTs**). Upon restart of ImageJ, the scientific colour map(s) should then be available under **Image > Lookup Tables**.

Alternatively, the colour-map .lut file may be applied using either (a) **File > Open**, (b) **File > Import > LUT**, or (c) drag and drop the .lut file onto the ImageJ window. To view available LUTs: **Image > Color > Display LUTs**.

### 4.8.2 NeuroCyto LUTs add-on

Detailed information about how to use a simple add-on that adds a handy LUTs drop-down menu to the Fiji user interface is given on <https://forum.image.sc/t/neurocyto-luts-update-site/26244>.

## 4.9 Kingdom

### 4.9.1 .clm format

On any screen, select **Show color bar** from the toolbar. Above the colour bar that appears, select **Select?**, then under **Files of type** choose **Color Bars (\*.CLM)**, then navigate to the location the colour-map files are stored.

The continuous Scientific Colour Maps are also provided externally in Kingdom's native file format, .clb, for easier implementation. The .clb files are available separately on [www.fabiocrameri.ch/colourmaps](http://www.fabiocrameri.ch/colourmaps). To import them in Kingdom, select, on any screen, **Show color bar** from the toolbar. Above the colour bar that appears, select **Select?**, then navigate to the location the colour-map files are stored.

## 4.10 Mathematica

### 4.10.1 .mat format

```
ColorMapSuitePath = "/Path/To/ColourMapSuite/";
```

```

ColorMapSuite[name_String] := ColorMapSuite[name, -1]
ColorMapSuite[name_String, e1_] := With[{
  list =
    Transpose@{Subdivide[0, 1, 255],
      RGBColor @@@
      First@Import[
        ColorMapSuitePath <> "/" <> name <> "/" <> name <> ".mat"]}]
  },
  Blend[list, {##}][[e1]]] &
]

```

The function call `ColorMapSuite["name", i = -1]` returns a lambda function whose *i*th argument is used to define color (see the Manual for `ColorFunction` for details). "name" should be replaced with the name (in quotes) of the color scheme, e.g. "davos". Be sure to set the variable `ColorMapSuitePath` to the path where your `ColorMapSuite` is installed.

General rules are:

- 1D plots of 1D functions/data: no (default) argument *i* suffices
- 2D plots of 2D functions/data: no (default) argument *i* suffices
- 3D plots of 2D functions/data: use *i* = 3
- 3D plots of 3D functions/data: use *i* = 4 (results might be worse than default Mathematica color functions, possibly due to lack of surface normal mapping)

```

ContourPlot[Sin[x] Sin[y], {x, 0, 2 Pi},
  {y, 0, 2 Pi}, ColorFunction -> ColorMapSuite["davos"]]

```

## 4.11 MatLab

### 4.11.1 .mat format)

Load the colour map into MatLab, either by adding the .mat file to the MatLab search path and using the command:

```
load('davos.mat');
```

or by specifying the full file path to the .mat file:

```
load('~/.work/Colormaps/davos.mat');
```

Then use it, for example, with:

```

figure(1)
colormap(davos)
colorbar

```

### 4.11.2 File-exchange app

A convenient MatLab package provided by Chad Greene containing the full scientific colour-map suite is available on [MatLab file exchange](#).



## 4.12 Ncview

### 4.12.1 .ncmap format

The colour map .ncmap files can live in the following places:

1. `NCVIEW_LIB_DIR`, which is determined at installation time. A reasonable choice is `/usr/local/lib/ncview`.
2. In a directory named by the environmental variable `NCVIEWBASE`.
3. If there is no environmental variable `NCVIEWBASE`, then in `$HOME`.
4. In the current working directory.

Then when you open Ncview, it should automatically have all of the colour maps available.

## 4.13 Originlab

### 4.13.1 .pal format

To use the .pal colour map files in the Originlab software, download the external package from [www.fabiocrameri.ch/colourmaps](http://www.fabiocrameri.ch/colourmaps) or via the direct [link](#). The .PAL files can then be copied to the origin palette folder and used similarly to the Originlab default color palettes.

## 4.14 Paraview

### 4.14.1 .xml format

Using Scientific colour maps in Paraview is done via the following procedure:

Click **Edit** color map panel. Once the colour map settings open, click the folder with the heart (i.e., **Choose Preset**), then **Import**, and then choose the PARAVIEW.xml format (e.g., `batlow_PARAVIEW.xml`). The colour map is now loaded and saved in Paraview, so one can now simply search for the colour map name (e.g., `batlow`) in the search field for the colour maps. Click on the desired colour map and hit **Apply**.

## 4.15 Petrel

### 4.15.1 .alut format

To import colour maps, select the **templates** pane and **colour tables** folder.

Then select the folder to import into (or insert a new folder) and right click **import on selection**.

Select **colour tables (alut files) (\*.alut)** to view and select all suitable colour maps for import.

Accept default settings **trim colour control points** and **trim opacity control points** and finally use as any other colour table within Petrel.

## 4.16 Photoshop

### 4.16.1 .grd format

The .grd format to read into Photoshop provided by Steven Reddy can be found at: [www.geoscienceatomprobe.org/downloads.html](http://www.geoscienceatomprobe.org/downloads.html)

## 4.17 Plotly

### 4.17.1 .py format

Plotly versions of the scientific colour maps are provided by Emilia are available at <https://github.com/empet/scientific-colorscales>.

The plotly scientific colour maps (see the file `scicolorscales.py`) were created by converting the provided .py file of each colour map.

Direct applications and some scientific tests are illustrated in this Jupyter Notebook: <http://nbviewer.jupyter.org/github/empet/scientific-colorscales/blob/master/Tests-for-scientific-colorscales.ipynb>.

## 4.18 Python

### 4.18.1 Package (pip and anaconda)

The convenient python package, <https://pypi.org/project/cmcrameri/>, by Callum Rollo is available through pip and anaconda.

Install with pip:

```
pip install cmcrameri
```

Install with conda:

```
conda config --add channels conda-forge
conda install cmcrameri
```

Usage example:

```
from cmcrameri import cm
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 100, 100)[None, :]
plt.imshow(x, aspect='auto', cmap=cm.batlow) # or any other colourmap
plt.axis('off')
plt.show()
```

### 4.18.2 init file

A simple init file located in `ScientificColourMaps6/+TOOLS/` can be used to make the whole suite of colour maps readily available in python: Place the `__init__.py` file in the main directory `ScientificColourMaps7/` and update your PYTHONPATH environment like this:

(for linux/bash)

```
export PYTHONPATH=$PYTHONPATH:/full/path/to/ScientificColourMaps7/
```

Then, in any python, you can import the palette collection, by using import ScientificColourMaps7 as SCM7, which allows for example commands like

```
plt.imshow(some_data, cmap=SCM7.berlin)      # Linear palette
plt.imshow(some_data, cmap=SCM7.berlin_r)    # Reversed, linear
plt.imshow(some_data, cmap=SCM7.berlin25_r)  # Reversed, 25 steps, discrete
```

#### 4.18.3 .txt format

##### Step 1: Load colour-map data

Load the colour-map data into Python using `numpy.loadtxt()`:

```
import numpy as np
cm_data = np.loadtxt("roma0.txt")
```

##### Step 2: Set up colour map

Use `matplotlib.colors.LinearSegmentedColormap()` to create a colour map that can be used with matplotlib.

```
from matplotlib.colors import LinearSegmentedColormap
roma0_map = LinearSegmentedColormap.from_list(?roma0?, cm_data)
```

##### Complete example:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

cm_data = np.loadtxt("roma0_RGB(0-1).txt")
roma0_map = LinearSegmentedColormap.from_list(?roma0?, cm_data)

x = np.linspace(0, 100, 100)[None, :]
plt.imshow(x, aspect=?auto?, cmap=roma0_map)
plt.axis(?off?)
plt.show()
```

#### 4.18.4 palettable library

The [palettable library](#) provides the Scientific Colour Maps (and other scientific colour maps) in a convenient way for use with e.g., matplotlib. Palettable is available on PyPI for installation via pip: `pip install palettable`. Palettable is compatible with Python 2.6, 2.7, and Python 3. For more instructions see [here](#).

## 4.19 QGIS

#### 4.19.1 .xml format

Load the colour map into QGIS in:

```
Settings > Style manager > Import/Export > Import symbol(s)
> select the xxx_QGIS.xml file.
```

## 4.20 R

### 4.20.1 scico package

`'scico'` (<https://travis-ci.org/thomasp85/scico>) – pronounced as “psycho” – is a small package developed by Thomas Lin Pedersen that provides access to the scientific colour maps within R. It provides colour palettes for `'ggplot2'` without requiring `'ggplot2'` to be installed.

`scico` can be installed from CRAN with `install.packages('scico')`. If you want the development version then install directly from GitHub:

```
# install.packages("devtools")
devtools::install_github("thomasp85/scico")
```

For further details and user instructions are included in a README file within `'scico'`.

## 4.21 SKUA-GOCAD

### 4.21.1 .xcmmap format

To import a colormap into a SKUA-GOCAD project, navigate to **File > Import > GOCAD Resources > Colormaps**.

Alternatively, for advanced users, to include a colormap as a resource in all new projects, insert the `.xcmmap` text into the `*colormaps.xml*` file located in `*/Gocad/lib/app-defaults`.

## 4.22 VisIt

### 4.22.1 .ct format

The file `davos.ct` can be imported to VisIt by placing the `.ct` file in the `.visit` directory, which can be found on macOS under e.g.,:

```
/Applications/VisIt.app/Contents/Resources/ ...
... 2.12.3/darwin-x86_64/resources/colortables
```

The colour map should appear in the built-in list after VisIt has been restarted.

## 5 Software with built-in versions

- [GMT](#) 6.0 and later
- [TopoToolbox](#) 2.2 and later
- [StagLab](#) 3.0 and later
- [SubMachine](#)
- [Geoscience ANALYST](#) 2.80 and later

## 6 References

Included colour-map diagnostics are based on:

**Kovesi (2015)**, Good Colour Maps: How to Design Them, CoRR, abs/1509.03700, <http://arxiv.org/abs/1509.03700>\* and related MatLab functions available at <https://www.peterkovesi.com/matlabfns/index.html#colour>.

For further details see:

**Crameri, F. (2018)**, Geodynamic diagnostics, scientific visualisation and StagLab 3.0, *Geosci. Model Dev.*, 11, 2541-2562, [doi:10.5194/gmd-11-2541-2018](https://doi.org/10.5194/gmd-11-2541-2018)

**Crameri, F., G.E. Shephard, and P.J. Heron (2020)**, The misuse of colour in science communication, *Nature Communications*, 11, 5444. [doi: 10.1038/s41467-020-19160-7](https://doi.org/10.1038/s41467-020-19160-7)

## 7 Version history

- Version 1 : Original colour-map suite (*bilbao*, *broc*, *cork*, *davos*, *devon*, *grayC*, *lajolla*, *oslo*, *vik*)
- Version 2 : Additional colour-map file formats (.ct, .py, .svg)
- Version 3 : Additional colour-map file format (.txt)
  - Dark background palettes (*berlin*, *lisbon*, *tofino*)
  - Additional sequential palettes (*lapaz*, *tokyo*, *turku*)
  - Surface topography special palette (*oleron*)
  - Seismic tomography special palette (*roma*)
- Version 4 : Additional sequential palettes (*acton*, *bamako*, *buda*, *hawaii*, *imola*, *nuuk*)
  - Scientific rainbow palette (*batlow*)
  - Additional colour-map file formats (xmlQGIS, .clr)
- Version 5 : **Discrete colour maps** (e.g., *batlow10*)
  - Improved perceptual uniformity of the original (v1) palettes
  - Minor colour adjustment to *vik*
  - Additional colour-map file formats (.alut, .ct, .lut, .ncmap, .pal, .spk, .xcmap)
- Version 6 : **Categorical colour maps** (*batlowS*, *devonS*, *davosS*, *osloS*, *lapazS*, *actonS*, *lajollaS*, *bilbaoS*, *grayCS*, *tokyoS*, *turkuS*, *bamakoS*, *nuukS*, *hawaiiS*, *budaS*, *imolaS*)
  - Cyclic colour maps** (*romaO*, *brocO*, *corkO*, *vikO*)
  - Improved colour-map diagnostics for perceptual uniformity
  - Additional colour-map file formats (.clm)
  - Improved User guide
  - Change to MIT License

Version 7 : Additional diverging palettes (**bam**, **vanimo**)  
Additional cyclic palette (**bamO**)  
Additional multi-sequential palettes (**bukavu**, **fes**)  
**batlowW**: batlow with white ending  
**batlowK**: batlow with black ending  
*cork*: Improved lightness symmetry  
*roma*: Improved lightness symmetry  
More formats (e.g., .pal for OriginLab)  
Updated `__init__.py` to flip colour gradients by Tobias Staal  
New Python package by Callum Rollo  
Colour map type and class flow chart

## 8 License

### The Scientific Colour Maps are licensed under a [MIT License](#)

Copyright (c) 2021, Fabio Crameri

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.