

- ✓ Perceptually uniform
- ✓ Perceptually ordered
- ✓ Colour-vision-deficiency (CVD) friendly
- ✓ Readable as black and white print
- ✓ Provided in all major formats
- ✓ Citable & reproducible
- \* Diverging
- \* Light centred; 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)

---

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



# Contents

<b>1</b>	<b>Creators</b>	<b>4</b>
<b>2</b>	<b>Sources of inspiration</b>	<b>4</b>
<b>3</b>	<b>Acknowledgement</b>	<b>5</b>
<b>4</b>	<b>Instructions</b>	<b>6</b>
4.1	Using the .mat format (MatLab) . . . . .	6
4.2	Using the file-exchange app (MatLab) . . . . .	6
4.3	Using the .cpt format (GMT) . . . . .	6
4.4	Using the .ct format (VisIt) . . . . .	6
4.5	Using the .mat format (Mathematica) . . . . .	6
4.6	Using the .xml format (QGIS) . . . . .	7
4.7	Using the .xcmap format (SKUA-GOCAD) . . . . .	7
4.8	Using the init file (Python) . . . . .	7
4.9	Using the .txt format (Python) . . . . .	7
4.10	Using the .py format (plotly) . . . . .	8
4.11	Using the .xml format (d3) . . . . .	8
4.12	Using the .pal format (Gnuplot) . . . . .	8
4.13	Using the .lut format (ImageJ/Fiji) . . . . .	8
4.14	Using the NeuroCyto LUTs add-on (ImageJ/Fiji) . . .	9
4.15	Using the .alut format (Petrel) . . . . .	9
4.16	Using the .ncmap format (Ncview) . . . . .	9
4.17	Using the .spk format (Ferret) . . . . .	9
4.18	Using the .clm format (Kingdom) . . . . .	9
4.19	Using the .cmap format (Fledermaus & Qimera) . . .	10
4.20	Using the .pal format (Originlab) . . . . .	10
4.21	Using the scico package (R) . . . . .	10
4.22	Using the .gpl format (GIMP/Inkscape) . . . . .	10
<b>5</b>	<b>Software with built-in versions</b>	<b>10</b>
<b>6</b>	<b>References</b>	<b>11</b>
<b>7</b>	<b>License</b>	<b>12</b>

## 1 Creators

Author: [Fabio Crameri](#)

Contributors: [Grace Shephard](#) – Conversion to .cpt format and various other contributions

[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

[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

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

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

## 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 vik (Crameri 2018a) is used in this study to prevent visual distortion of the data and exclusion of readers with colour-vision deficiencies (Crameri 2018b)."

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

The research : **Crameri, F. (2018b), 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)**

## 4 Instructions

### 4.1 Using the .mat format (MatLab)

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
```

### 4.2 Using the file-exchange app (MatLab)

A convenient MatLab package provided by Chad Greene containing the full scientific colour-map suite is available on [MatLab file exchange](http://www.mathworks.com/matlabcentral/fileexchange/).

### 4.3 Using the .cpt format (GMT)

The file davos.cpt can be resampled for a given z-value range with the Generic Mapping Tools (GMT; <http://gmt.soest.hawaii.edu/>) 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
```

### 4.4 Using the .ct format (VisIt)

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.

### 4.5 Using the .mat format (Mathematica)

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

```
ColorMapSuite[name_String] := ColorMapSuite[name, -1]
ColorMapSuite[name_String, el_] := 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"]]
```

#### 4.6 Using the .xml format (QGIS)

Load the colour map into QGIS in:

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

#### 4.7 Using the .xcmap format (SKUA-GOCAD)

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 `.xcmap` text into the `*colormaps.xml*` file located in `*/Gocad/lib/app-defaults`.

#### 4.8 Using the init file (Python)

A simple init file located in `ScientificColourMaps5/+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, and then, when in python, use `import SCM5`, which allows for commands like `plt.imshow(some_data, cmap=SCM5.berlin)`.

#### 4.9 Using the .txt format (Python)

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

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

```
import numpy as np
cm_data = np.loadtxt("vik.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
vik_map = LinearSegmentedColormap.from_list('vik', cm_data)
```

### Complete example:

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

cm_data = np.loadtxt("vik_RGB(0-1).txt")
vik_map = LinearSegmentedColormap.from_list('vik', cm_data)

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

## 4.10 Using the .py format (plotly)

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>.

## 4.11 Using the .xml format (d3)

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

## 4.12 Using the .pal format (Gnuplot)

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

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

## 4.13 Using the .lut format (ImageJ/Fiji)

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**.

#### 4.14 Using the NeuroCyto LUTs add-on (ImageJ/Fiji)

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>.

#### 4.15 Using the .alut format (Petrel)

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 Using the .ncmap format (Ncview)

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.

#### 4.17 Using the .spk format (Ferret)

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).

#### 4.18 Using the .clm format (Kingdom)

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.

#### 4.19 Using the .cmap format (Fledermaus & Qimera)

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](#).

#### 4.20 Using the .pal format (Originlab)

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.

#### 4.21 Using the scico package (R)

'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'.

#### 4.22 Using the .gpl format (GIMP/Inkscape)

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.

### 5 Software with built-in versions

- [StagLab](#) 3.0 and later
- [GMT](#) 6.0 and later
- [TopoToolbox](#) 2.2 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)

## 7 License

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

Copyright (c) 2020, 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.