buda

a scientific colour map - www.fabiocrameri.ch/buda

Crameri (2018)

DOI 10.5281/zenodo.1243862

Characteristics

- Perceptually uniform ✔
- Perceptually ordered ✔
- Colour-vision-deficiency (CVD) friendly ✔
- Readable as black and white print ✔
- Sequential
- Low lightness gradient

Author

- **Fabio Crameri** - *Colour map* - contact

Contributors

- **Grace Shephard** – *Conversion to .cpt format and various other contributions* – contact
- **Clint Conrad** – *Wider compatibility of .cpt format* – contact
- **Matteo Albano** – *Conversion to .clr format and compatibility with QGIS* – contact
- **Casper Pranger** – *Mathematica compatibility* – contact
- **Alexis Plunder** – *Wider compatibility of .xml format* – contact
- **Krister Stræte Karlsen** – *User instruction for use with python* – contact
- **Philippe Rivière** – *Conversion instruction for d3* – contact
- **Emilia** – *Plotly versions* – contact
- **Thomas Lin Pedersen** – *The 'scico' package for use with R* – contact
- **Paul Wessel** – *Built-in version for 'GMT'* – contact
- **Wolfgang Schwanghart** – *Built-in version for 'TopoToolbox'* – contact
- **Chad Greene** – *MatLab file exchange version* – contact
- **Sean Trim** – *Conversion to .pal format* – contact
- **George Edward Campbell** – *Conversion to .lut format* – contact
- **Kirstie Wright** – *User instruction for use with Petrel* – contact
- **Jennifer Levett** – *Conversion to SKUA-GOCAD .xcmap format* – contact
- **Sam Hatfield** – *Conversion to Ncview .ncmap format* – contact
- **Patrick Brockmann** – *Conversion to Ferret .spk format* – contact
- **Mark Wieczorek** – *Import init file for Python* – contact

Acknowledgement

Please acknowledge the free use of the colour map.

> e.g., "The perceptually uniform colour map ***buda*** is used in this study to prevent visual distortion of the data (Crameri 2018a,b)."
>
> *Crameri, F. (2018a), Scientific colour-maps. Zenodo. http://doi.org/10.5281/zenodo.1243862*
>
> *Crameri, F. (2018b), Geodynamic diagnostics, scientific visualisation and StagLab 3.0, Geosci. Model Dev., 11, 2541-2562, doi:10.5194/gmd-11-2541-2018*

Instructions

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('buda.mat');
```

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

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

Then use it, for example, with:

```
figure(1)
colormap(buda)
colorbar
```

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.

Using the .cpt Format (GMT)

The file buda.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 -Cbuda.cpt -T-2000/2000/100 > buda_resampled.cpt
```

Using the .ct Format (VisIt)

The file buda.ct can 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.

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, {##}[[el]]] &
  ]
```

The function call `ColorMapSuite["name", i = -1]` returns a lambda function whose ith 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. `"buda"`. 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["buda"]]
```

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.
```

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`.

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)`.

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("buda.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
buda_map = LinearSegmentedColormap.from_list('buda', cm_data)
```

Complete example:

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

cm_data = np.loadtxt("buda_RGB(0-1).txt")
buda_map = LinearSegmentedColormap.from_list('buda', cm_data)

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

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.

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.

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"
```

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`.

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.

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 colourmaps available.

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

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 scales 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`.

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.

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*

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. Discuss., doi:10.5194/gmd-2017-328*

License

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.