colorview2d Documentation

Release 0.51

Alois Dirnaichner

February 13, 2015

Contents

1 gpfile **3**
1.1 Example 3

2 The plugin interface **7**
2.1 IMod 7
2.2 ModWidget 8

3 Indices and tables **9**

Python Module Index **11**

Contents:

CHAPTER 1

gpfile

A module to handle 3D gnuplot datafiles.

It can be used to get information on the file, access the contents via numpy arrays and extract the ranges along the three axes.

The files are represented by the gp_file object. The class provides methods to rotate, flipp, copy and save the datafile.

1.1 Example

```
file = gp_file.gp_file('data.dat',(0,1,2))
file.rotate_cw()
file.report()
file.save('newdata.dat')
```

class `colorview2d.gpfile.`**`gpfile`**`(`**args*`)`

A gnuplot data file object.

Usage:

The datafile object can be initiated by the use of a filename and a column tuple. Alternatively, one can provide a 2d array of data. The axes are then just given by the index ranges.

filename A gnuplot datafile to be read in.

columns A tuple of three integers to specify the three columns.

or:

data Alternatively, a 2d numpy array can be provided.

Attributes:

- **Zdata (numpy.array)** The two dimensional numpy arrray containing the actual data.
- **Xrange (numpy.array)** A one dimensional array representing the x axis range.
- **Yrange (numpy.array)** A one dimensional array representing the y axis range.
- **Xleft, Xright (float)** The values on the left/right of the x axis.
- **Ytop, Ybottom (float)** The values on the top/bottom of the y axis.
- **Zmin, Zmax (float)** The min/max values of the 2d array.
- **Xmin, Xmax (float)** The min/max values of the x axis range.
- **Ymin, Ymax (float)** The min/max values of the y axis range.

- **filename (string)** The filename of the ASCII file containing the data.

crop (*xleft*, *xright*, *ybottom*, *ytop*)
Crop the datafile to a subset of the array specifiying the corners of the subset in units of the axes ranges.

Parameters

- **xleft** –
- **xright** –
- **ybottom** –
- **ytop** –

deep_copy ()
Deep copy the datafile object.

Returns A copy of the datafile object.

flip_lr ()
Flip the left and the right side of the datafile. The axes are updated as well.

flip_ud ()
Flip the up and the down side of the datafile. The axes are updated as well.

get_xrange_idx (*value*)
Return the nearest index of a value within the x axis range.

Parameters **value** (*float*) – A value in the range of the x axis

Returns The closest index on the x axis range.

get_yrange_idx (*value*)
Return the nearest index of a value within the y axis range.

Parameters **value** (*float*) – A value in the range of the y axis

Returns The closest index on the y axis range.

report ()
Print a datafile report to the standart output.

rotate_ccw ()
Rotate the datafile counter-clockwise. The axes are updated as well.

rotate_cw ()
Rotate the datafile clockwise. The axes are updated as well.

save (*fname*, *comment='*')
Saves a datafile to a file with filename in the gnuplot format.

Parameters

- **fname** (*str*) – The filename of the ASCII file to contain the data.
- **comment** (*str*) – A comment that is added to the top of the file.

set_Zdata (*Zdata*)
Replace the 2d array containing the data. The Zmin and Zmax values are updated.

Parameters **Zdata** (*numpy.array*) – Two dimensional array to plot.

set_xyrange (*Xrange*, *Yrange*)
Specify the x and y ranges of the datafile.

Parameters

- **Xrange** (*numpy.array*) – One dimensional numpy array with the same length as the width of the data.
- **Yrange** (*numpy.array*) – One dimensional numpy array with the same length as the height of the data.

CHAPTER 2

The plugin interface

The plugin interface consists of two modules, the

IMod class interface specification

ModWidget widget base class

Please have a look at specific plugin implementations for examples.

2.1 IMod

Interface to the mod plugin class. Specification of the minimum requirements of a plugin implementation. It also provides basic interaction with the ModWidget class. In particular, the checkbox of the widget is handled.

class `colorview2d.IMod.`**`IMod`**

The interface class is an abstract base class. At present, none of the methods have to be overwritten, though.

The following attributes are defined:

Title (string) Title string of the plugin. Usually equal to the plugin/module name.

Args (tuple) The arguments that have to be provided to the mod to work.

Active (bool) The current state of the plugin.

`activate``()`

Activate the plugin. Usually, this method does not have to be overwritten by a plugin. The yapsy plugin activation routine is called. If a checkbox widget is defined, it is checked. The mod is added to the pipeline.

`apply`(*datafile*)

This method has to be overwritten to provide some useful functionality. It should modify the view object using the parameter in args.

Parameters **datafile** – A *gpfile* object.

`create_widget`(*panel*)

Create a widget for the plugin. This Method has to be overwritten to create a custom widget, i.e., a widget that contains more than a simple checkbox.

Parameters **panel** (*wx.Panel*) – The panel object to create the widget on.

Returns The widget object.

`deactivate``()`

Deactivate the plugin. Usually, this method does not have to be overwritten by a plugin. The yapsy plugin

deactivation routine is called. If a checkbox widget is defined, it is unchecked. The mod is removed from the pipeline.

update_widget()
Update the widget using the mod data.

2.2 ModWidget

Module provides the base class for plugin widgets.

It contains a checkbox widget and can be used directly in a mod plugin.

class ModWidget.**ModWidget**
Base class for the plugin widgets that can be provided to colorview2d.

The widget is a descendant of wx.BoxSizer. Childs should call init and update upon overwrite.

Variables

- **mod** – The mod class this widget is assigned to
- **panel** – The panel this widget lives on
- **title** – The title of the mod.
- **chk** – A handle for the checkbox widget.

update()
Update the mod widget to comply with the state of the mod. Should update all widgets contained in the ModWidget descendant.

on_chk(*event*)
Handler for the *event* that is triggered by activating/deactivating the checkbox.

It is not required to overwrite this function in a custom plugin implementation.

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*

Python Module Index

c

colorview2d.gpfile, 3
colorview2d.IMod, 7

m

ModWidget, 8

A

activate() (colorview2d.IMod.IMod method), 7
apply() (colorview2d.IMod.IMod method), 7

C

colorview2d.gpfile (module), 3
colorview2d.IMod (module), 7
create_widget() (colorview2d.IMod.IMod method), 7
crop() (colorview2d.gpfile.gpfile method), 4

D

deactivate() (colorview2d.IMod.IMod method), 7
deep_copy() (colorview2d.gpfile.gpfile method), 4

F

flip_lr() (colorview2d.gpfile.gpfile method), 4
flip_ud() (colorview2d.gpfile.gpfile method), 4

G

get_xrange_idx() (colorview2d.gpfile.gpfile method), 4
get_yrange_idx() (colorview2d.gpfile.gpfile method), 4
gpfile (class in colorview2d.gpfile), 3

I

IMod (class in colorview2d.IMod), 7

M

ModWidget (class in ModWidget), 8
ModWidget (module), 8

O

on_chk() (ModWidget.ModWidget method), 8

R

report() (colorview2d.gpfile.gpfile method), 4
rotate_ccw() (colorview2d.gpfile.gpfile method), 4
rotate_cw() (colorview2d.gpfile.gpfile method), 4

S

save() (colorview2d.gpfile.gpfile method), 4
set_xyrange() (colorview2d.gpfile.gpfile method), 4
set_Zdata() (colorview2d.gpfile.gpfile method), 4

U

update() (ModWidget.ModWidget method), 8
update_widget() (colorview2d.IMod.IMod method), 8