
Photon

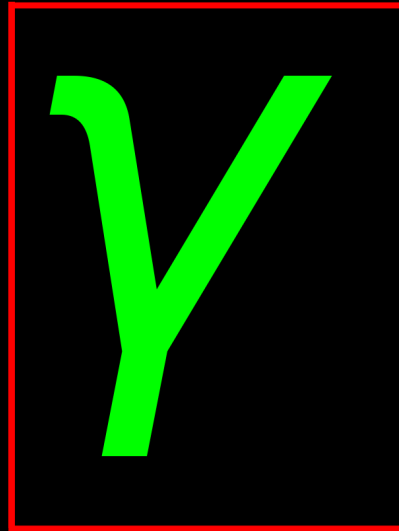
Release 0.3.1

Apr 21, 2022

Contents

1	Contents
----------	-----------------

3



PHOTON

PytHOn Tool for
Data plOttiNg

R. Thomas - 2018

1.1 Using Photon

1.1.1 Installation

For the moment, the only way to use Photon is to download the source code from the Github repository (<https://github.com/astrom-tom/Photon> - using the 'clone or download' button and download as zip). To make it work you have to use python 3.6 with the following library and versions:

- PyQt5 v5.10.1: This is for the graphical interface
- Matplotlib v2.2.2: This is for the plotting area.
- Numpy v1.14.2: catalog handling
- Scipy v1.0.1: Some useful function for smoothing / images
- astropy v3.0.2: For fits image opening

Photon is available in the pypi repository. To install it:

```
1 pip install photon_plot --user
```

1.1.2 The help

You start photon from a terminal. The program comes with a help that you can display in your terminal using the help command. Use it like this:

```
[user@machine]$ photon --help
```

This command will display the help of the program:

```
usage: photon [-h] [-f FILE] [-c CUSTOM] [-p PLOT] [-w WIDTH] [-d] [--version]
```

Photon, R. Thomas, 2018, This program comes **with** ABSOLUTELY NO WARRANTY; **and** **is** distributed under the GPLv3.0 Licence terms. See the version of this Licence distributed along this code **for** details. website: <https://github.com/astrom-tom/Photon>

-h, --help	show this help message and exit
-f FILE, --file FILE	Your catalog of data to visualize, this is mandatory if you do not use -p option
-p PLOT, --plot PLOT	Saved plot configuration file, if none is given you must provide a file
-c CONF, --conf CONF	Properties configuration file. If none is givenm the default configuration will be loaded
-w WIDTH, --width WIDTH	Width of the GUI, default = 780
--version	display version of photon

In details it means:

- Photon must come with a file as first argument OR configuration plot. If you do not give Photon one of these it will complain (you can see below for the format of the file you can provide photon with).
- Few arguments can be used:
 - -f: this is a catalog you want to display
 - -p: Plot configuration. This plot configuration is created using the save button in the GUI. It allows you to reload a plot.
 - -c: Custom configuration file: If you saved a customization configuration previously you can load it using this argument and giving the customization configuration.
 - -w: width of the window: The GUI is not resizable. Which means that the size of the plot is pre-fedined. By default the width of the window is sized at 780. Using this option, (e.g. -w 1080) will resize the GUI with a width of 1080.
 - -d: This will open the internal documentation
 - --version: Display in terminal the current version of the software

1.1.3 Catalog format

For the current version Photon accept only *ascii* catalogs of columns and *fits* image. Some precisions:

- The catalog must contain at least 2 columns and 2 lines.
- Each line must contain the same number of column.
- If your catalog starts with a header the header must start with a hash '#' and each column must be named.
- When plotting images, The fits handling library that is used in Photon is called 'fitsio'. Your image must contain only one table for the image. The fitsio command that is used is 'fitsio.read('image.fits')'.

1.1.4 The Graphical User Interface

When starting Photon, the graphical interface shows up in the screen (Screenshot below).

It is composed of five area:

1. The plotting Area, where the plot will be displayed.

2. The matplotlib toolbar. Allows you to zoom in/out, select specific part of the plot, etc. . .
3. The ‘Add element in plot’ button. It is the button you will use to add elements in the plot such as Line plot, Scatter plot, histograms, strips, text, etc. . .
4. Graphical Elements Panel. Then you click in the previous button, the widgets for the graphical elements you want to use will be displayed in that area. The widgets depend on the type of element you want to add into the plot. (*Available elements*)
5. Customization Panel. This Panel remains the same all the time and help you to customize graphical properties of the plot itself such as: background color, axis tickness or color, ticks label size, font sizes, etc. (*Available Graphical customization*)

When clicking on ‘Add element in plot’ you have these different choices:

- line
- line / new file
- scatter
- scatter /new file
- histogram
- histogram /new file
- Error
- Error / new file
- Straight line
- Span
- Text
- Image
- Image / new file

All the plotting elements are described here: *Available elements*. As you can see some plotting elements are repeated (e.g. ‘line’ and ‘line / new file’). As you start Photon with a particular file it can be useful to be able to load data from another file. If this is the case you can use ‘XX / new file’ and Photon will open dialog window to choose another file to use in photon. When loading a pre-existing plot configuration, if you want to add more data to your plot, you will only have the ‘New file’ choices.

1.2 Available elements

When clicking on ‘Add element in plot’ you have these different choices:

- line
- line / new file
- scatter
- scatter / new file
- scatter CB
- scatter CB / new file
- histogram

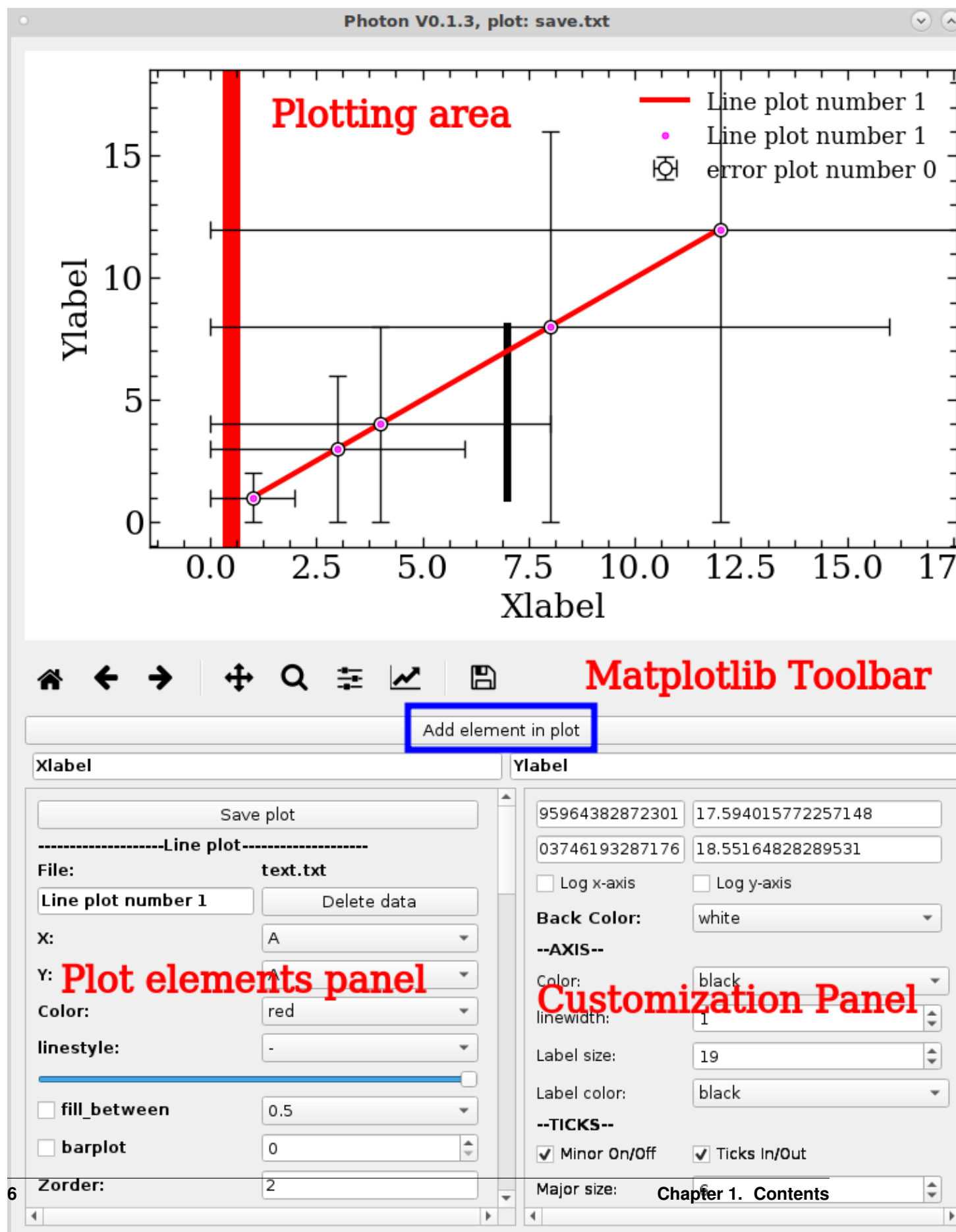


Fig. 1: Fig2: GUI of Photon

- histogram / new file
- Error bars
- Error bars / new file
- Image
- Image / new file
- Band
- Band / new image
- Straight line
- Span
- Text

1.2.1 Line plots

The figure on the right shows the widget that will appear when you want to add a line plot.

The first line shows you the name of the loaded file for which you choosed to draw the scatter plot. On the second line, you have a text field (with 'histogram number X' by default). This is where you enter The *delete button* allows you to remove the scatterp lot from the plot and also all the associated widgets in the graphic element panel. The *X:* and *Y:* widgets ask for the column corresponding to the X and Y axis of your plot from your input catalog. The *color* widget makes you change the color of each points. The *linestyle* widget allows you to change the line style of your line (dot, dashed...). The sidebar allows one to change the tickness of the line. The *fill between* and associated color list choice will fill the space between your plot and the x-axis with the chosen color. The checkbox *barplot* will transform your line plot to a line-bar plot. Instead of creating line directly from the poin it will draw steps at each point. Finally, the spinbox with number allows you to smooth your line plot by a gaussian filter. The width of the gaussian filter is given by the number chosen in the spinbox.

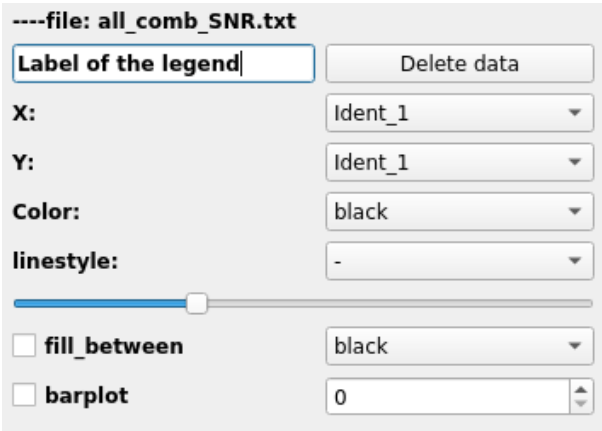


Fig. 2: Line plot related widgets

1.2.2 Band plots

The figure on the right shows the widget that will appear when y ou start a band plot. The first line shows you the file you use for this plot. The second line allows you to give the label of the plot (see below for details). The *delete button* allows you to remove the band plot from the plot and also all the associated widgets in the graphic element panel. Then you have to give *X*, *Y1*, *Y2*. Photon will create a plot filled with color between *Y1* and *Y2*. The *color Band* widget allows you to select the filling color. Finally

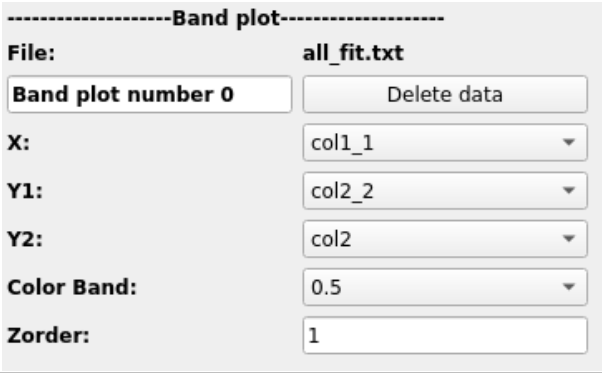


Fig. 3: Band plot related widgets

the *zorder* parameter allows you to choose how all the elements of the plot are piled up (which one is on top of eachother).

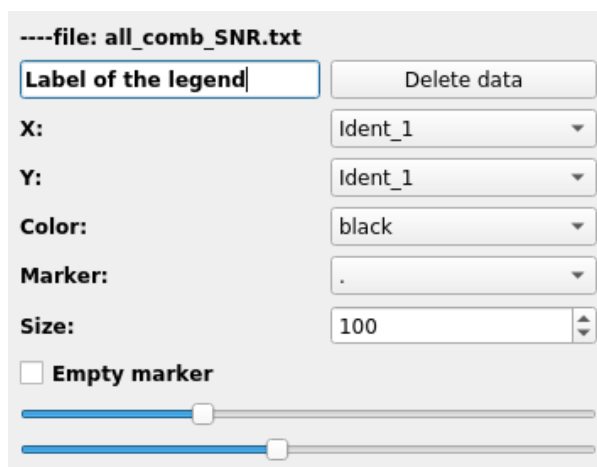
Warning: It is worth mentionning that the `fill_between` method of `matplotlib` does not have a `label` argument. Therefore a line is created at the mean of *Y1* and *Y2* with the same color and same *zorder* as the band (it is therefore invisible). This label is assigned to this invisible line in the legend.

1.2.3 Scatter plots

The figure on the right shows the widget that will appear when you want to add scatter plot.

The first line shows you the name of the loaded file for which you choosed to draw the scatter plot. On the second line, you have a text field (with 'histogram number X' by default). This is where you enter The *delete button* allows you to remove the scatter plot from the plot and also all the associated widgets in the graphic element panel. As previously the *X:* and *Y:* ask for the column corresponding to the X and Y axis of your plot from your input catalog. The *color* widget makes you change the color of each points. The *marker* widget allows you to change the symbol you want to use for your scatter plot. The *size* spinbox will change the size of the symbols. The checkbox *Empty Marker* will remove the filling of the marker (e.g. filled circle -> empty circle).

Warning: If your symbols as only lines (e.g. crosses, '+')... checking the 'Empty Marker' box will make your symbol disappear.



----file: all_comb_SNR.txt

Label of the legend | Delete data

X: Ident_1

Y: Ident_1

Color: black

Marker: .

Size: 100

☐ Empty marker

Two horizontal sliders for marker line thickness and transparency.


Fig. 4: Scatter plot related widgets

The two last slidebars make you control the tickness of the line drawing the marker and the transparency.

1.2.4 Scatter plots with colorbar

The figure on the right shows the widget that will appear when you want to add scatter plot that includes a colorbar.

The first line shows you the name of the loaded file for which you choosed to draw the scatter plot. On the second line, you have a text field (with 'scatter plot with color bar number X' by default). This is where you enter the label of your plot for the legend. The *delete button* allows you to remove the scatter plot from the plot and also all the associated widgets in the graphic element panel. Then you have



-----Scatter plot with color bar-----

File: Mag_ABS_allselected.txt

scat color bar | Delete data

color bar label: final test

X: rmj

Y: nuvmr

Z: NUV_GALEX_Abs

Colormap: rainbow

zmax: -18

zmin: -20

Marker: .

the label for the color bar with *color bar label*. As previously the *X:* and *Y:* ask for the column corresponding to the X and Y axis of your plot from your input catalog. Now you have a third column to give (*Z*) that corresponds to the data given by the color bar. The *colorbar* widget makes you change the color of each points. The *vmin* and *vmax* will give the opportunity to modify the limits of the color bar. The *marker* widget allows you to change the symbol you want to use for your scatter plot. The *size* spinbox will change the size of the symbols. The checkbox *Empty Marker* will remove the filling of the marker (e.g. filled circle → empty circle).

Warning: If your symbols are only lines (e.g. crosses, '+', ...) checking the 'Empty Marker' box will make your symbol disappear.

The two last slidebars make you control the tickness of the line drawing the marker and the transparency.

Then you have the *zorder* parameter that controls how the data are pilling up. And finally you have all the properties for the color bar itself. You can control the fonts of both the label of the colorbar and the tick label as well as their sizes. The labelpad allows you to move the colorbar label closer to the colorbar or aways from it.

1.2.5 Histogram plots

The figure on the right shows the widget that will appear when you want to add an histogram plot.

The first line shows you the loaded file for which you choosed to draw the histogram. On the second line, you have a text field (with 'histogram number X' by default). This is where you enter the label of the histogram that will be displayed in the legend (can be left blank). As usual the delete button allows you to remove the histogram from the plot and also all the associated widgets in the graphic element panel.

Then, the '*X:*' widget ask you to choose the column of the input catalog whom the histogram is drawn from. The *Binning* widget allows you to choose the binning of the histogram. It need 3 numbers in the format '*xi*, *xf*, *dx*'. Where '*xi*' is the first bin, *xf* is the last one and *dx* the width of each bin. The *color* widget provides all the available colors. The *histtype* multiple choice widget let you choose between different histogram type (bars, steps, filledsteps...). If you choose the 'steps' style, it will just draw the contour of the histogram with a line. In that case you can choose the linestyle in the widget *linestyle*. Two slidebars are displayed after. The first controls the transparency of the

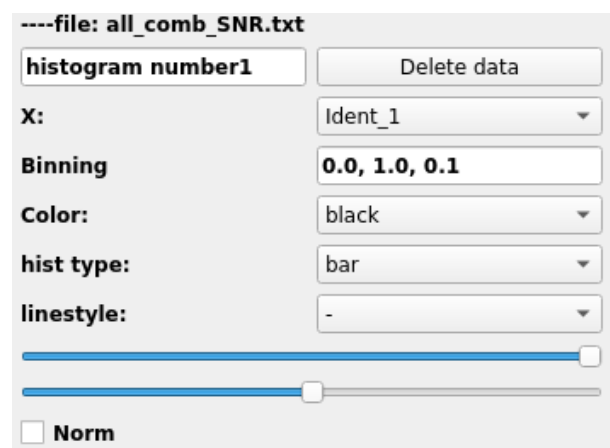


Fig. 6: Histogram related widgets

histogram, and the second controls the tickness of the line. Finally, a check box *norm* allows one to normalise the histogram. It will recompute all the bins so the sum of all of them is equal to one.

1.2.6 Errorbar plots

The figure on the right shows the widget that will appear when you want to add a plot with error bars. This kind of plot are basically equivalent to scatter plots but with some more information on how to plot the error bars

The first line shows you the name of the loaded file for which you choosed to draw the errorbar plot. On the second line, you have a text field (with 'Errorbar plot number1' by default). This is where you enter the label that will go into the legend of the plot. The *delete button* allows you to remove the errorbar plot from the plot and also all the associated widgets in the graphic element panel. As previously the *X:* and *Y:* ask for the column corresponding to the X and Y axis of your plot from your input catalog.

Then, you have to give **four** columns for the error bars. As errors can be asymmetric two values are needed along X and two values are also needed along Y. Along X, you have to give the two errors with the widgets *Xerr+* and *Xerr-*. Along Y, you have to give the two errors with the widgets *Yerr+* and *Yerr-*.

Then, you can choose the color (it will be the same for the marker and the errorbars) with the *color* widget. Additionally you can set the size of the marker using the *size* spinbox. The empty marker checkbox allows you to transform your marker (e.g. circles, squares, pentagons...) into empty marker where just the contour is displayed (see the warning above when checking this option). The two last slidebars make you control the tickness of the line drawing the marker and the transparency. Finally you can control the tickness of the errorbars and the size of the errorbar cap using the spinboxes *error bar tickness* and *error bar cap size*.

Fig. 7: Errorbar plot related widgets

1.2.7 Image plots

The figure on the right shows the widget that will appear when you want to plot an image.

As usual the first line shows the name of the loaded file (name of your image). On the second line you have the legend label ('image plot number1') and you can change it freely. On the same line you have the *zscale* checkbox that allows you to display the image with the ds9-like *zscale* colormap levels.

On the next line you have the possibility of changing the colormap. The list depends shows all the colorbars contained in matplotlib.

Fig. 8: Image plot related widgets

The last 3 lines allows you to plot contours over the image. To use them you have to check the 'Contour' checkbox and provide a threshold level. The scrollbar allows you to change the linewidth of the contour and the last line provide a list of colors for the contour line.

1.2.8 Straight lines

The figure on the right shows the widget that will appear when you want to add straight lines.

The first widget is a multiple choice button where you can choose between **vertical**, **horizontal** and **diagonal** lines. As usual the delete button allows you to remove the strip from the plot and also all the associated widgets in the graphic element panel.

The widget *coordinates* require two or three numbers. If you choose between 'horizontal' and 'vertical' lines, three numbers will be required. For example, if you choose vertical, you will have to give the X-coordinate, and the y-coordinate indicating the limits of the line you want to draw. If you choose 'diagonal' it will draw the line $x=y$ and you have to give the limits on the plot in the format 'xmin, xmax'. The widget *color* allows you to choose the color you want. The widget *linestyle* change the linestyle of the line. And finally the *slide bar* will help you to change the tickness of the line.

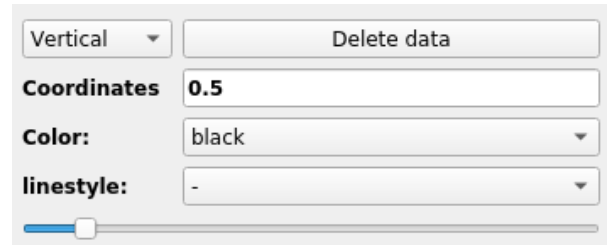


Fig. 9: Straight line related widgets

1.2.9 Spans

The figure on the right shows the widget that will appear when you want to add strips.

The first widget is a multiple choice button where you can choose between **vertical** and **horizontal** stripes. As usual the delete button allows you to remove the strip from the plot and also all the associated widgets in the graphic element panel. Then the coordinate field (coming by default at '0.3, 0.7') is where the limit of the stripes are going. The format must be ('x1, x2'). Then the color widget allows one to change the color of the span. And finally the slide bar change the transparency of the strip.

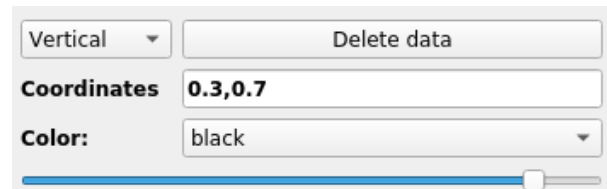


Fig. 10: Spanning stripes related widgets

1.2.10 Text

The figure on the right shows the widget that will appear when you want to add text. As usual the delete button allows you to remove the text from the plot and also all the associated widgets in the graphic element panel.

The text field (coming with an 'text' entry) is where you write the text you want to display. It can use Latex font. Then the coordinates field (coming by default at '0.3, 0.7') is where you must give the coordinates of the bottom left corner of the text. The format is 'x,y'. Then, the sidebar

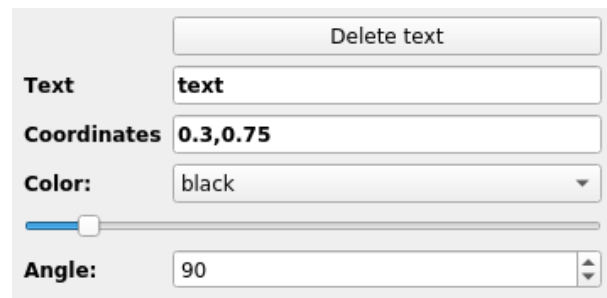


Fig. 11: Text related widgets

allows you to play with the size of the text. Finally the angle (from 0 degree to 360) allows one to rotate the text.

1.2.11 Save a plot configuration

Once your plot is finalized you can save all the configuration by clicking on the button ‘save plot’ (at the top of the Graphical element panel). Doing so will create a configuration file containing all your graphical elements input. Later on you can use this configuration plot and load it back to photon using the ‘-p’ argument. It will load all the widget and you will be able to modify your plot from where you stopped. An example of such plot looks like this:

```
[Types]
line = 2
scatter = 1
error = 1
text = 0
segments = 1
image = 0
diag = 0
hist = 0
strip = 1
xmin = -1.4095964382872301
xmax = 17.594015772257148
ymin = -1.0603746193287176
ymax = 18.55164828289531
x_label = Xlabel
y_label = Ylabel

[line_1]
file = text.txt
label = Line plot number 1
zorder = 2
x = A
y = A
color = red
style = -
color_fb = 0.5
fb = No
bp = No
thickness = 30
smooth = 0

[line_2]
file = /media/sf_Documents/text2.txt
label = Line plot number 1
zorder = 1
x = A
y = B
color = black
style = --
color_fb = 0.5
fb = No
bp = No
thickness = 10
smooth = 0
```

(continues on next page)

(continued from previous page)

```
[scat_1]
file = /media/sf_Documents/text2.txt
label = scatter plot number 1
zorder = 1
x = A
y = B
color = green
marker = D
empty = Yes
thickness = 10
transparency = 10
size = 100
```

```
[stra_1]
dir = Vertical
color = black
style = -
zorder = -1
thickness = 44
coor = 7, 1, 8
```

```
[stri_1]
dir = Vertical
color = red
zorder = 0
transparency = 100
coor = 0.3, 0.7
```

```
[erro_1]
file = text.txt
label = error plot number 0
zorder = 2
x = A
y = A
xerrp = A
xerrm = A
yerrp = A
yerrm = A
color = black
marker = .
empty = Yes
transparency = 10
size = 15
barsize = 10
capsize = 50
```

Warning: It is strongly suggested not to modify this file. As photon reads it, it might have trouble to reload your configuration if the file was modified by hand .

Warning: As you might see, for each type of plot the donfiguration gives the name of the file to be used. It will write the full path of the file with respect to the directory where you started photon.

1.3 Available Graphical customization

Photon allows you to customize your plot in many different aspects. This page will show what options you can use.

All the options are displayed in the figure on the right and we will detail them here.

1.3.1 Axis limits

The four first boxes (with 'Xmin', 'Xmax', 'Ymin' and 'Ymax') are made to constrain the axis limits of the plot. Of course, $X_{max} > X_{min}$ and $Y_{max} > Y_{min}$. To make it work you must fill all the boxes. If you only enter one Photon will not recognize it as a valid configuration and will set the constraints by itself to 0,1;0,1. Also you can choose to transform the x and y axis into log-scales. In order to do this you must check the Logx and/or Logy checkboxes.

1.3.2 Background color

The two widget about the background color (Label widget with 'back color', and color list) allow you to change the color of the background of your plot.

1.3.3 Axis properties

The program also let you play with the axis and their labels. This is done in the AXIS section. Here you can change the color of the axis, and the color of their labels as well. The tickness of the axis lines is controlled by the *linewidth* widget and the size of the labels is managed by the *Label size* widget.

1.3.4 Ticks properties

The properties of the ticks in the plot are controlled in the TICKS section. This section starts with 2 checkboxes. The first will add or remove the minorticks while the second will place them inside or outside. The four next widgets let you manage the length and width of the major and minor ticks (*Major size*, *Minor size*, *Major width*, *Minor width*). Then you can control the color of the ticks using the *Ticks color* widget as well as the color of the ticks labels with *Label Color*. Finally, the size of the ticks labels is modified with the *Label Size* widget.

The image shows a software interface for customizing a plot. It is organized into several sections with labels like --AXIS--, --TICKS--, --FONTS--, and --LEGEND--. Each section contains various input fields, checkboxes, and dropdown menus. At the bottom, there is a 'Save new Theme' button.

Xmin	Xmax
Ymin	Ymax
<input type="checkbox"/> Log x-axis	<input type="checkbox"/> Log y-axis
Back Color:	white
--AXIS--	
Color:	black
linewidth:	1
Label size:	19
Label color:	black
--TICKS--	
<input checked="" type="checkbox"/> Minor On/Off	<input checked="" type="checkbox"/> Ticks In/Out
Major size:	6
Minor size:	4
Major width:	1
Minor width:	1
Ticks Color:	black
Label Color:	black
Label Size:	19
--FONTS--	
Font axis label:	C059
Font ticks label:	C059
--LEGEND--	
<input checked="" type="checkbox"/> Frame On/Off	
Legend size:	19
Font legend:	C059
Label Color:	black
Legend location:	best
Save new Theme	

Fig. 12: Customization related widgets.

1.3.5 Fonts

The section FONTS will let you choose the font used by the labels. You can change both axis and ticks label fonts using *Font axis Label* and *Font ticks Label*, respectively. The fonts that you can use depends entirely on your system as Photon is asking Matplotlib what fonts are available (the font list is not hardcoded).

1.3.6 Legend

The LEGEND section help you to customize the legend of your plot. You have 4 options. The frame of the legend (box around the legend) can be turned On/Off with the first checkbox of the section. The legend size can also be modified using the *Legend Size* widget (this will modify the fontsize of the text in the legend). The font can also be changed using the *Font Legend* properties. The color of the text in the legend can be changed using the *Label color* widget. Finally, the legend position can be changed using the *Legend Location* widget. To let the legend be automatically placed you just have to choose 'best'. If you want it in a special place (9 possibilities) you can chose in the list what fits best for your plot.

1.3.7 Save the customization

Changing the configuration each time can become bothersome. If you want to keep the same style for all your plots you can save all the custom options in a configuration file. Once you are happy with your configuration, press the 'Save new theme' button and it will open a file saving dialog window. Here you enter the name of the configuration you want to save (like in any other program).

The Photon customization file look like this:

```
[background]
back_color = white

[AXIS]
color = black
label_color = black
linewidth = 1
labelsize = 17
axis_label_font = DejaVu Math TeX Gyre

[TICKS]
minor = on
placement = in
major_size = 6
minor_size = 4
major_width = 1
minor_width = 1
ticks_color = black
label_color = black
label_size = 17
ticks_label_font = DejaVu Math TeX Gyre

[LEGEND]
```

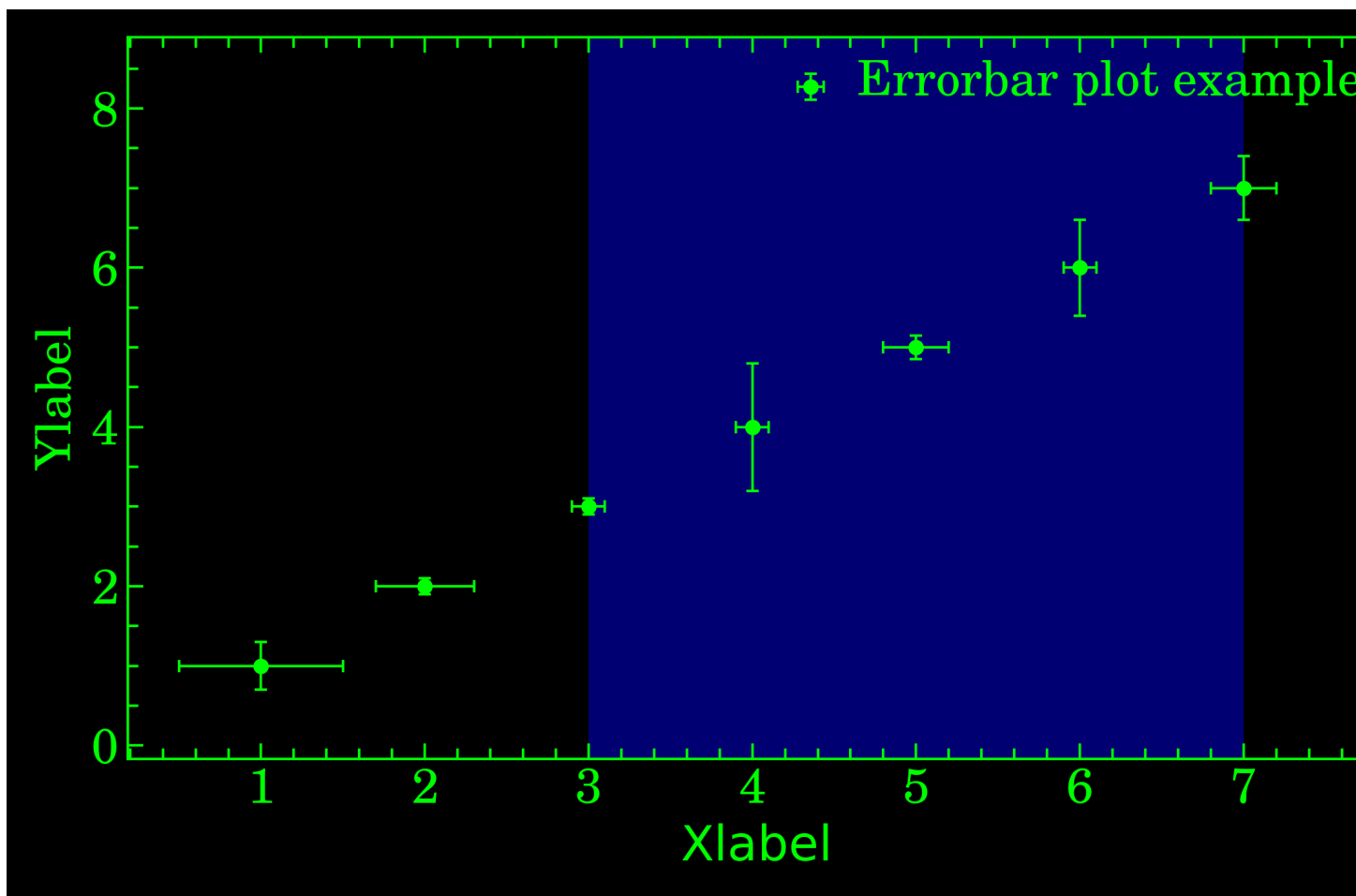
(continues on next page)

(continued from previous page)

```
frame = On
font_size = 15
legend_font = DejaVu Math TeX Gyre
label_font_color = black
```

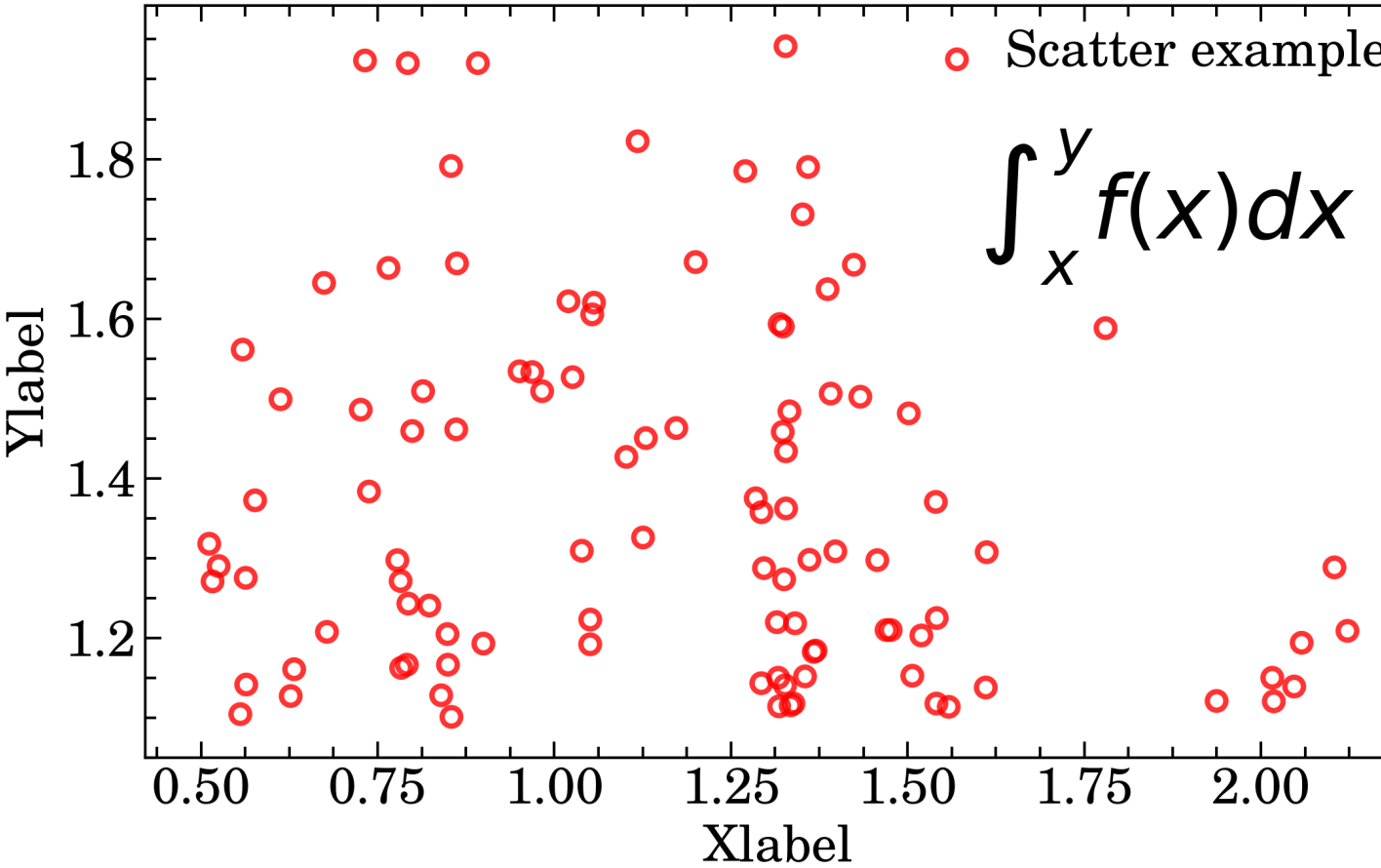
As shown in the page [Using Photon](#), you can then load directly Photon using this configuration with the `-c conf.file` option.

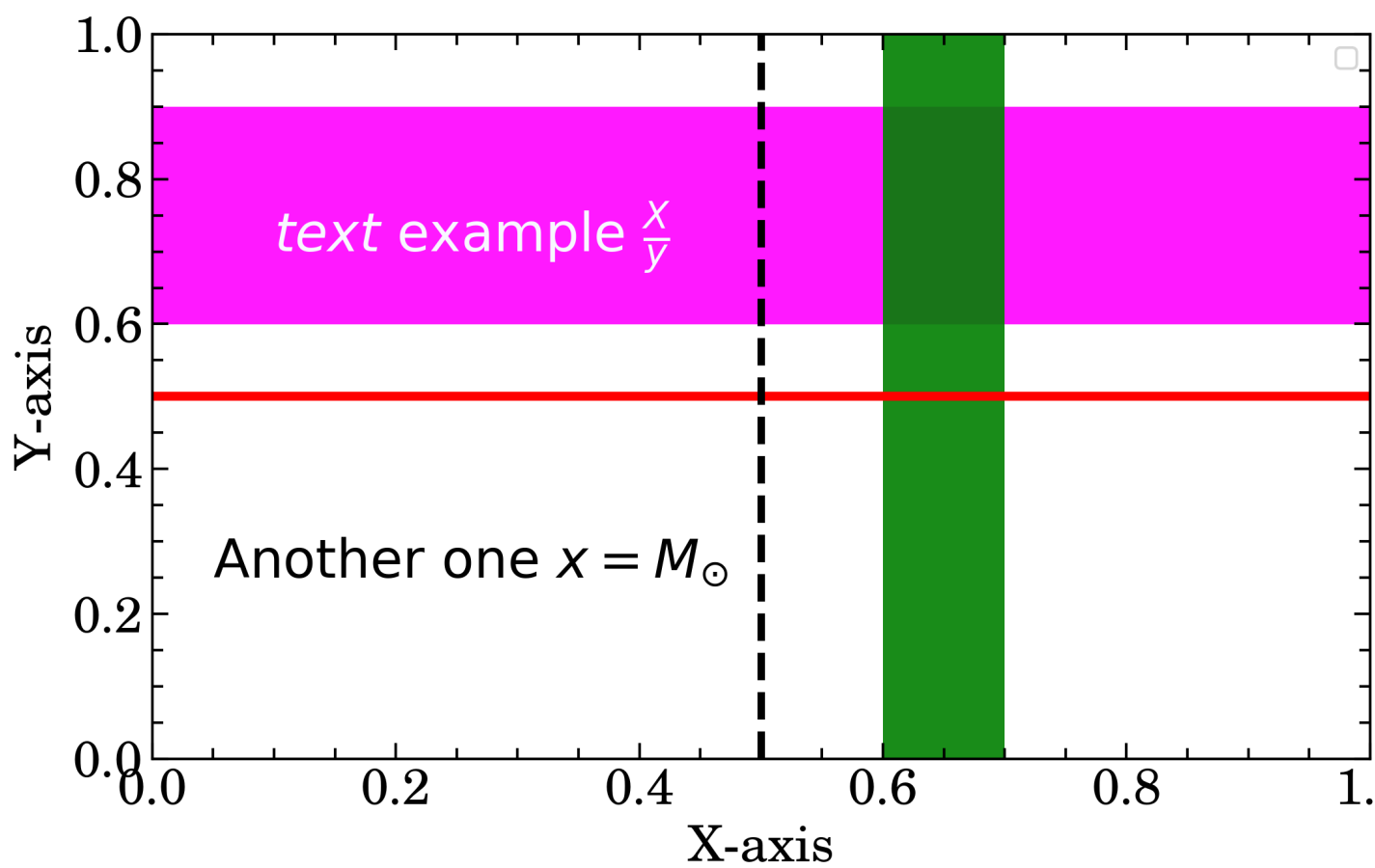
1.4 Gallery

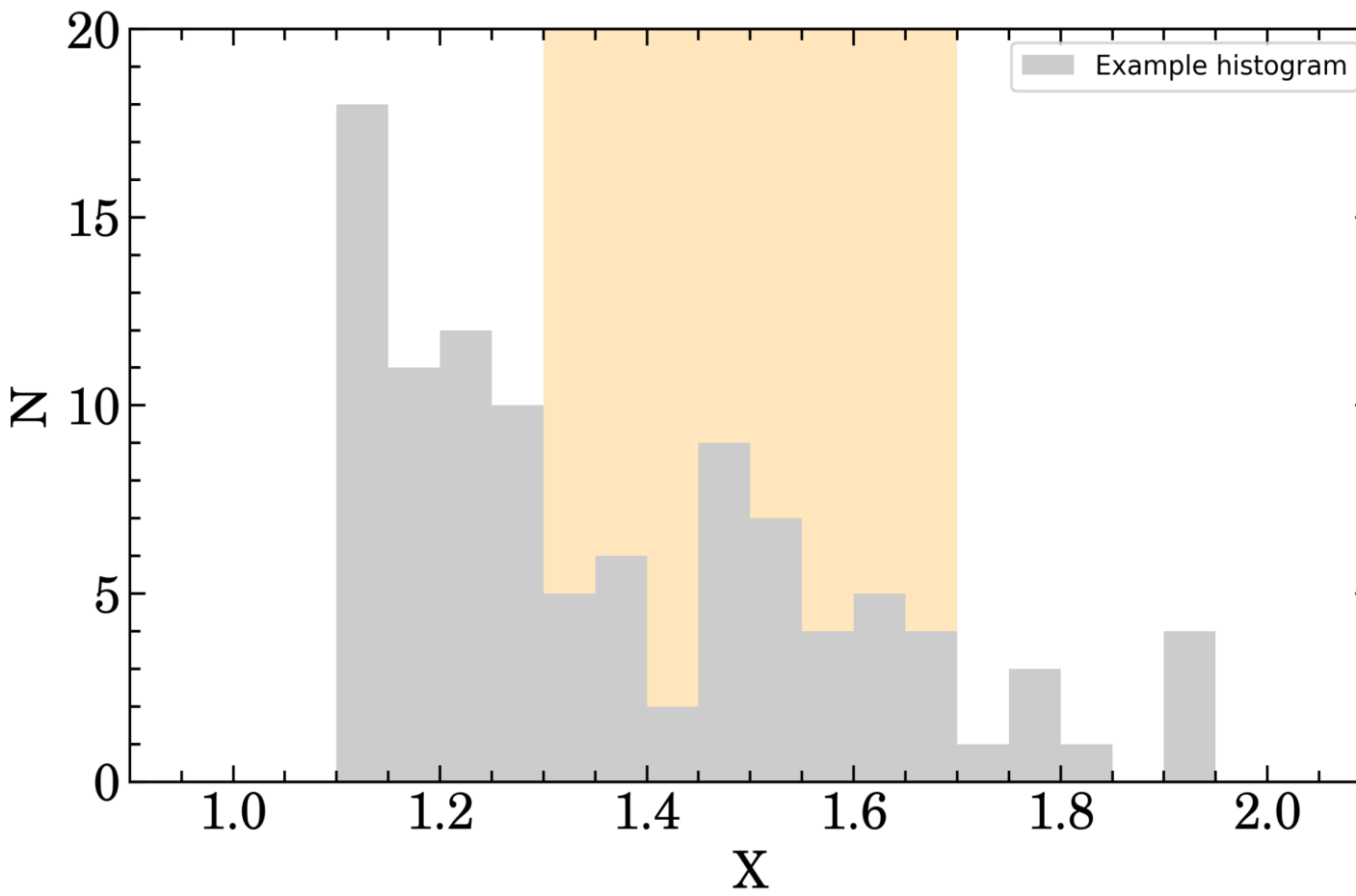


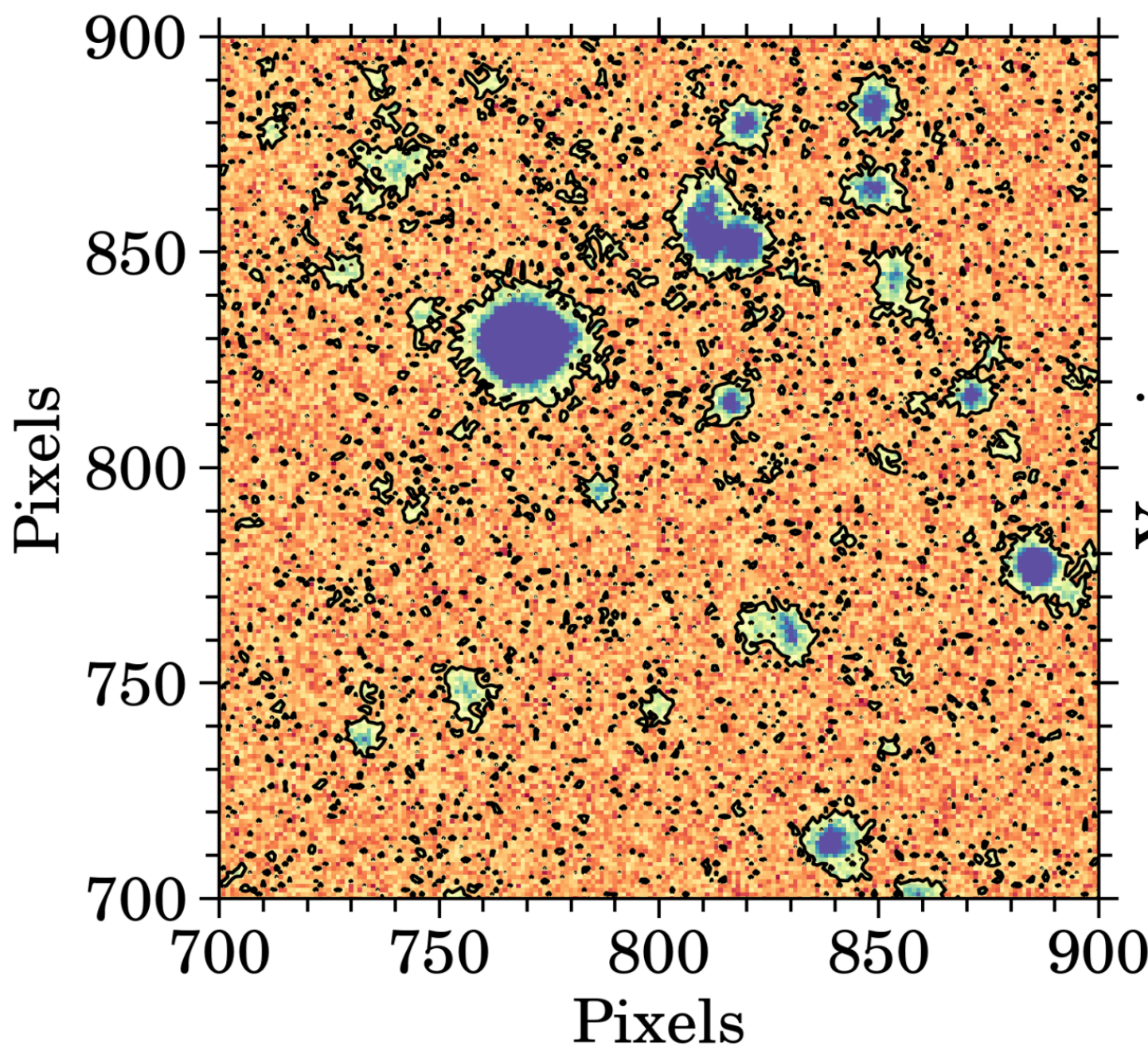
1.5 Change log

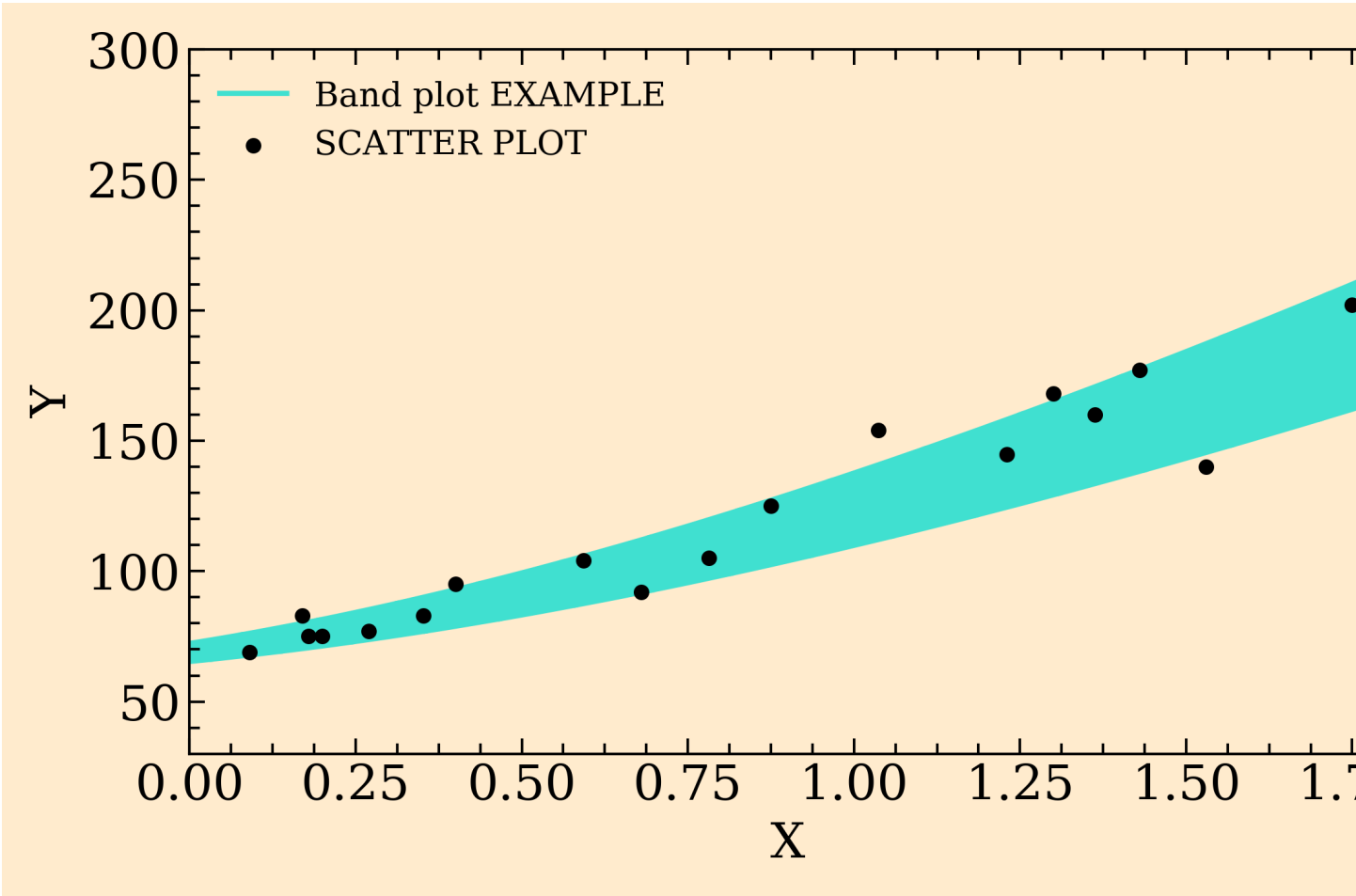
- photon 19.5.0:
 - Change in the numbering system Yr.month.change

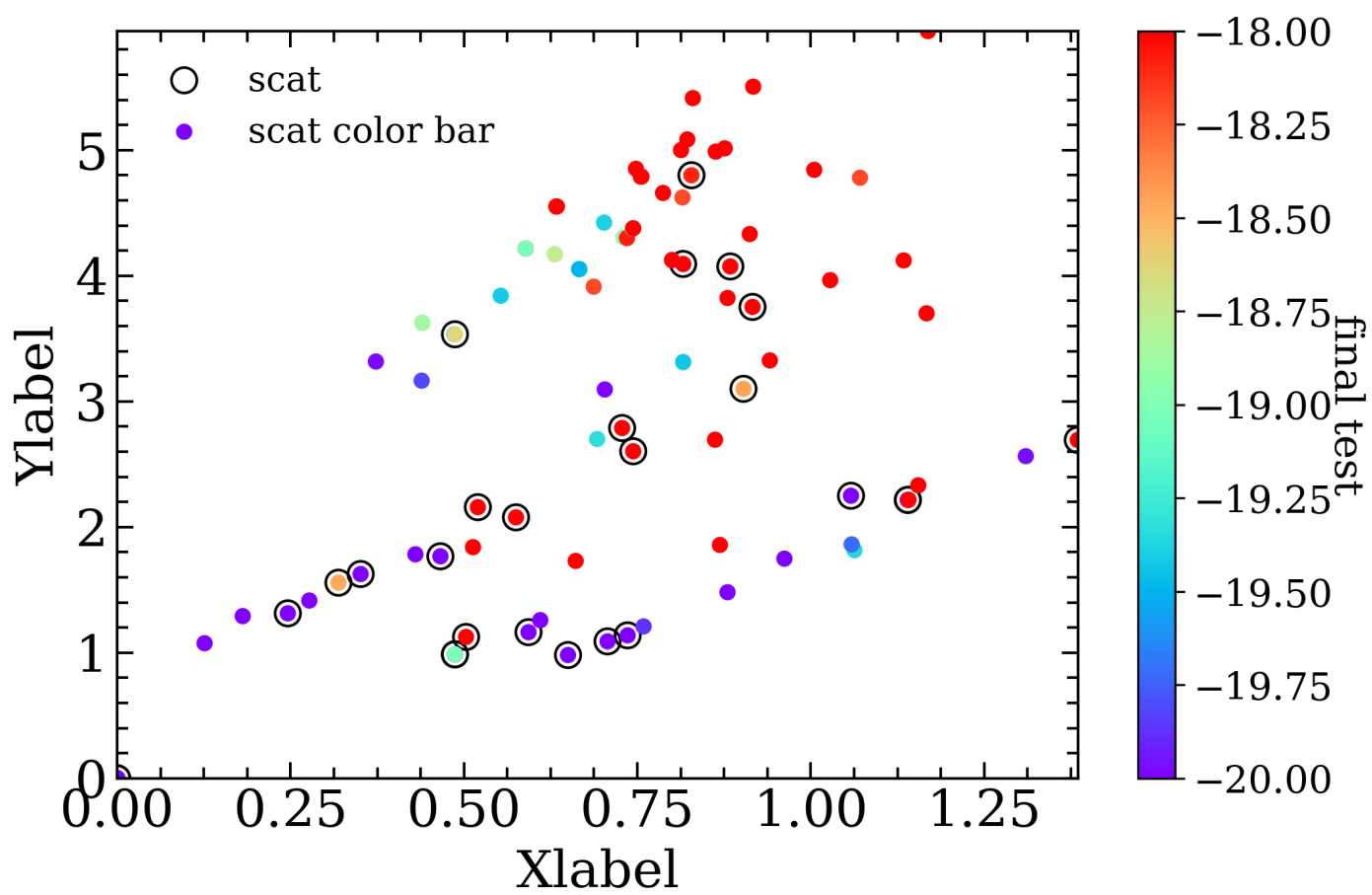












- Bug fix: For large amount of line plot
- **photon 0.3.5:**
 - Bug fix: column parameter
- **photon 0.3.3:**
 - add: possibility to split legend in multiplt columns
 - add: scatter bar with color bar
- **photon 0.2.2:**
 - modif: The file names are displayed now without the path.
 - bug fix: Straight line crashed photon when adding a straight line as first element
 - bug fix: zorder straight line was not taken into account
 - bug fix: Update the plot was modifying the style of the legend. Now it is fixed
- **photon 0.2.0:**
 - New **plot type**: band. This allows you to fill the space between two curves.
 - fitsio → astropy: mainly because some people reported problem with installing fitsio on py3.6.4.
- **photon 0.1.8:**
 - axis in error plot fixed (X and Y we reversed)
 - load of marker for error plot the marker that was saved in the plot configuration file was not extracted correctly
 - update Logo
 - update online and local documentation

Photon is a GUI created to make simple 1D plot in python. It uses mainly matplotlib and PyQt5 and it has been build in the spirit of being fully customizable. You can change fontstyle, fontsize, fontcolors, linewidth of the axes, thickness, etc... and see the changes directly in the plot. Once a customization is created, you can save it into a configuration file and reload it later when you open photon again. This allows you to keep the same customization for different plot. The main tool is a graphical user interface and it is started using a command line interface.

Contribute!

If you have any comment or anything you would like to be added to photon, or, even better, if you want to modify you can either do it yourself or please feel free to contact me! → @ the.spartan.proj@gmail.com

Warning: Copyright

Photon is a free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3 of the License.

Photon is distributed without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with the program. If not, see <http://www.gnu.org/licenses/>.