
Photon

Release 1.0

Jun 23, 2018

Contents

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

3

$\phi h \Theta \tau \theta \eta$ V1.0

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: The 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 usefull function for smoothing / images
- fitsio v0.9.10: Fits file handling library

Unless python3.6 is your main python, you have to edit one source file: 'src/photon'. This file is the main source code of photon and begin with the shebang: a line starting with '#!' followed by the path of the python to use. This allows to not have to type 'python' each time. Change this path accordingly to your python3.6 path. Once done you can add the src directory to your path and start using it! Go to the next section to learn how to use it!

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:

```
1 Photon V0.1, R. Thomas, 2018, ESO, This program comes with ABSOLUTELY NO WARRANTY;  
2 and is distributed under the GPLv3.0 Licence terms. See the version of this Licence  
3 distributed along this code for details.  
4  
5 positional arguments:  
6 file                Your catalog of data to visualize, this is mandatory  
7                     (positionnal argument)  
8  
9 optional arguments:  
10 -h, --help          show this help message and exit  
11 -t, --header        To be used If a header is present in the catalog (#A B  
12                     C D...), you just write '-t'. If you do not use it,  
13                     each column will be renamed colX where X is the number  
14                     of the column (starting at 1).  
15 -c CONF, --conf CONF Properties configuration file. If none is given the  
16                     default configuration will be loaded  
17 -w WIDTH, --width WIDTH Width of the GUI, default = 780  
18
```

In details it means:

- Photon must come with a file as first argument. This file is mandatory. If you do not give it Photon will complain (you can see below for the format of the file you can provide photon with).
- Three other OPTIONAL arguments can be used:
 - -t: if each column of your file as a header (e.g. a string at the top with the name of the column) you can ask Photon to read it using this argument. If you do not use it, the name of the columns will be col1, col2, ..., col1..., colN.
 - -c +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.

1.1.3 Input file 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 'fistio'. Your image must contain only one table for the image. The fitsio command that is used is 'fistio.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:

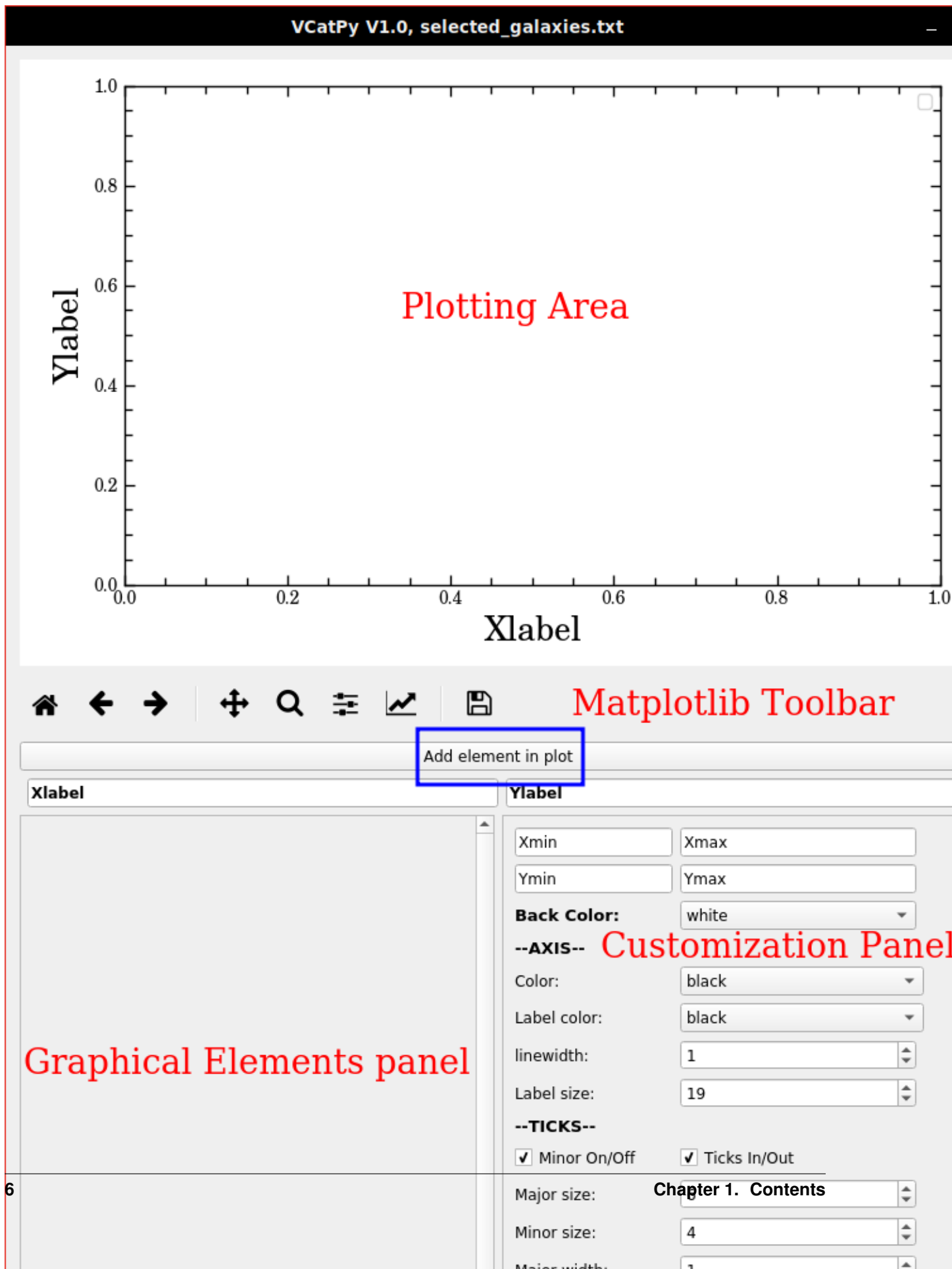
- line
- line / new file
- scatter
- scatter /new file
- histogram
- histogram /new file
- Error
- Error / new file
- Straight line
- Span
- Text
- Image
- Image / new file

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.

1.2 Available elements

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 bars
- Error bars / new file
- Image



- Image / new file
- 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 chose 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.

----file: all_comb_SNR.txt

Label of the legend | Delete data

X: Ident_1

Y: Ident_1

Color: black

linestyle: -

fill_between: black

barplot: 0

Fig. 2: Line plot related widgets

1.2.2 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 chose 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. As previsouxly 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).

----file: all_comb_SNR.txt

Label of the legend | Delete data

X: Ident_1

Y: Ident_1

Color: black

Marker: .

Size: 100

Empty marker

Fig. 3: Scatter plot related widgets

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.

1.2.3 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 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.

Fig. 4: Histogram related widgets

1.2.4 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 number X' 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 need 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*.

1.2.5 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.

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.

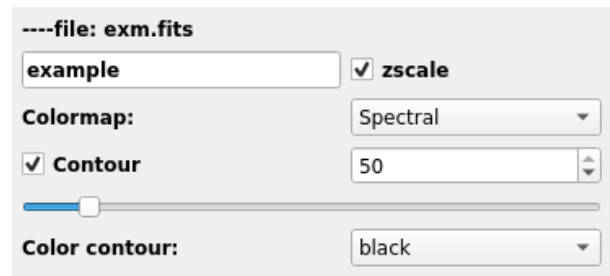


Fig. 6: Image plot related widgets

1.2.6 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 one or two number. If you choose between 'horizontal' and 'vertical' lines, only one number will be required. 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.

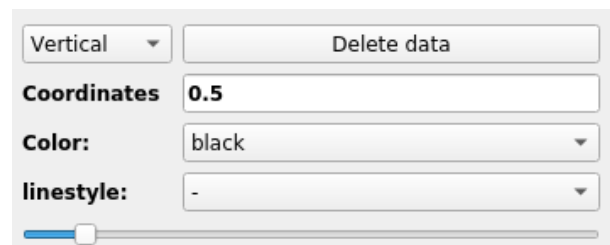


Fig. 7: Straight line related widgets

1.2.7 Spans

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

The first widget is a multiplt 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.

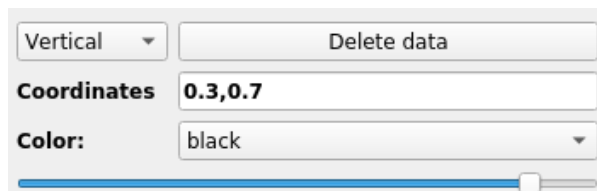


Fig. 8: Spanning stripes related widgets

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

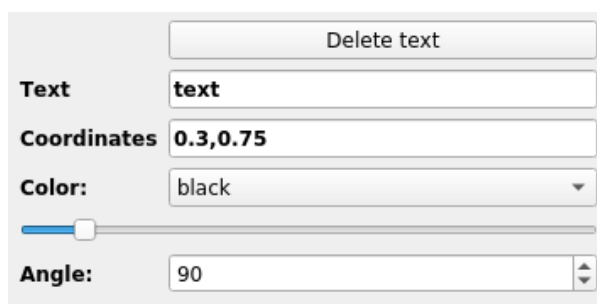


Fig. 9: Text related widgets

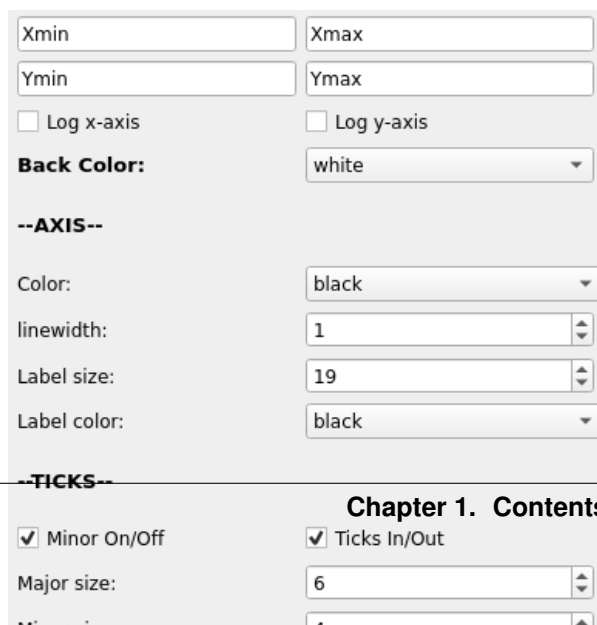
1.3 Available Graphical customization

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.

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
labels_size = 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

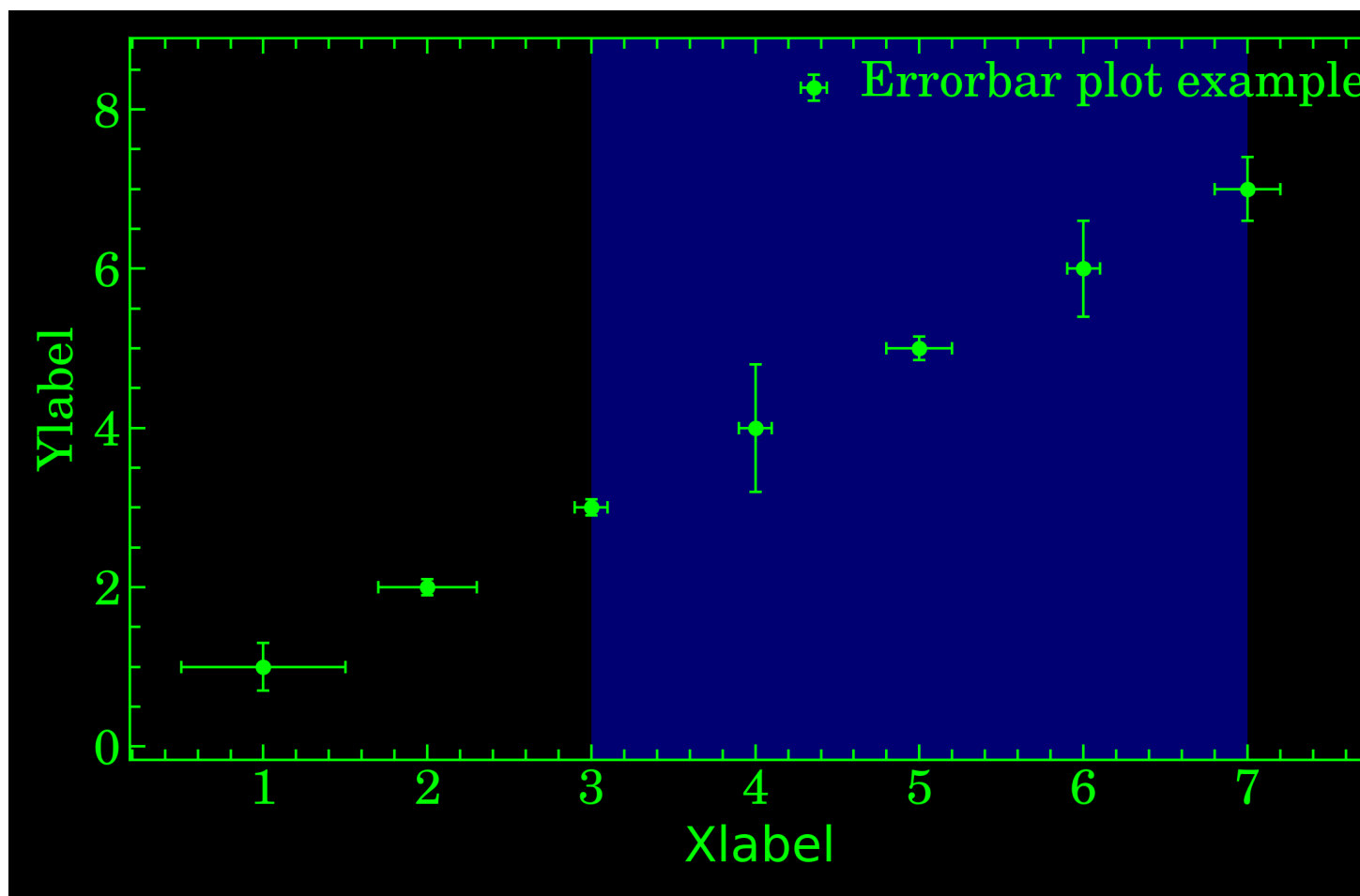
[LEGEND]
frame = On
font_size = 15
legend_font = DejaVu Math TeX Gyre
label_font_color = black
```

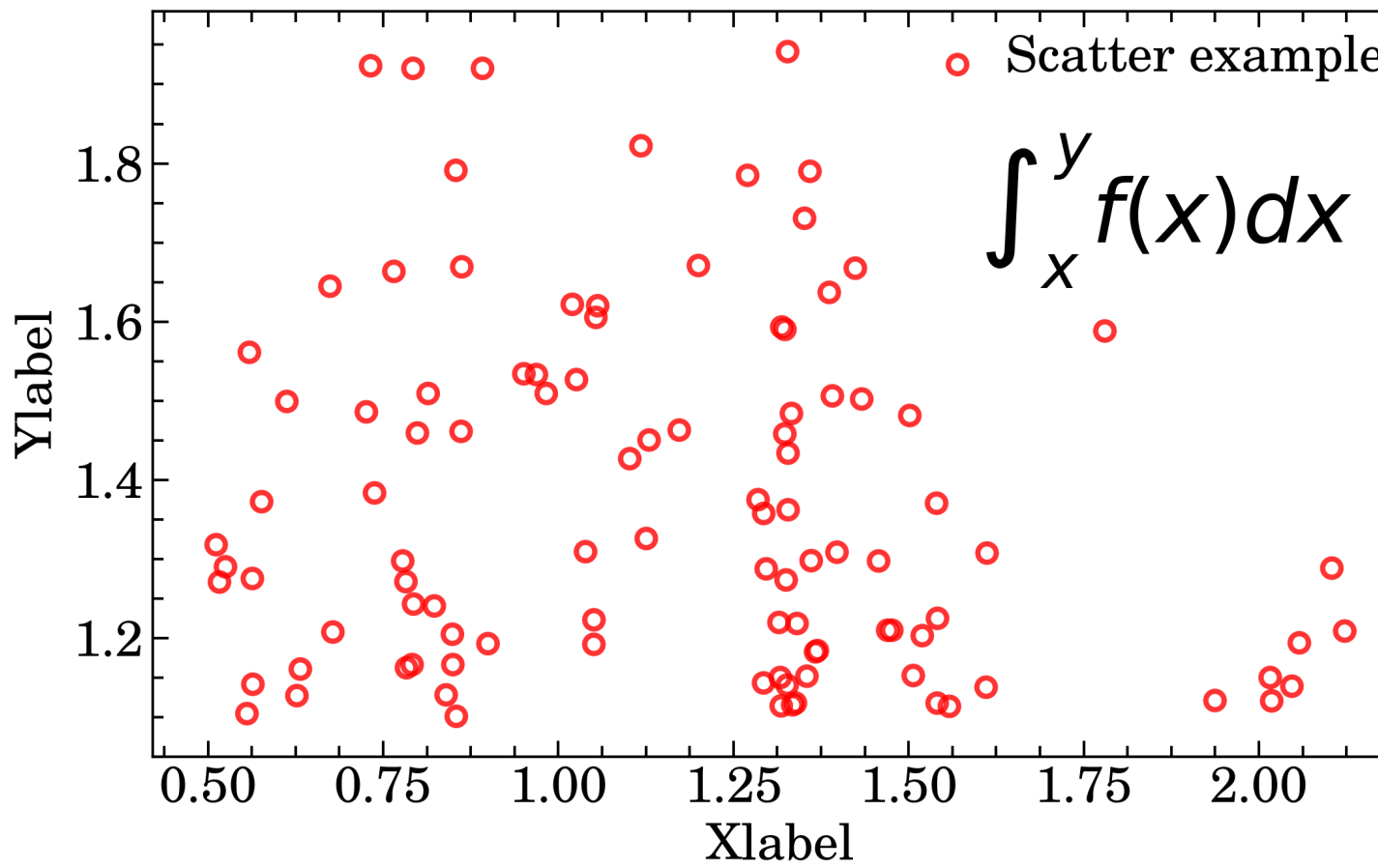
As shown in the page [Using Photon](#), you can then load directly Photon using this configuration with the `-c conf.file` option.

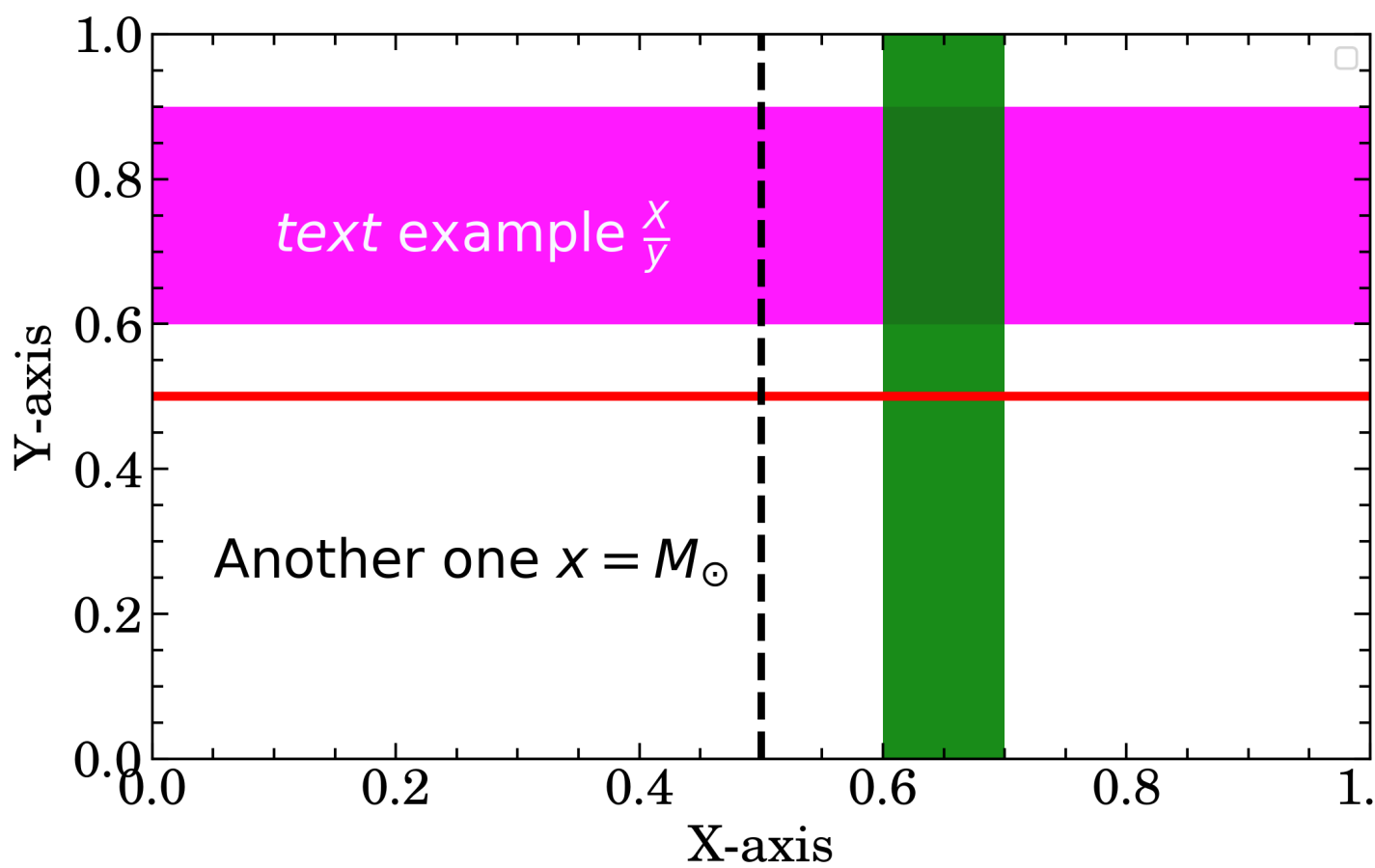
1.4 Gallery

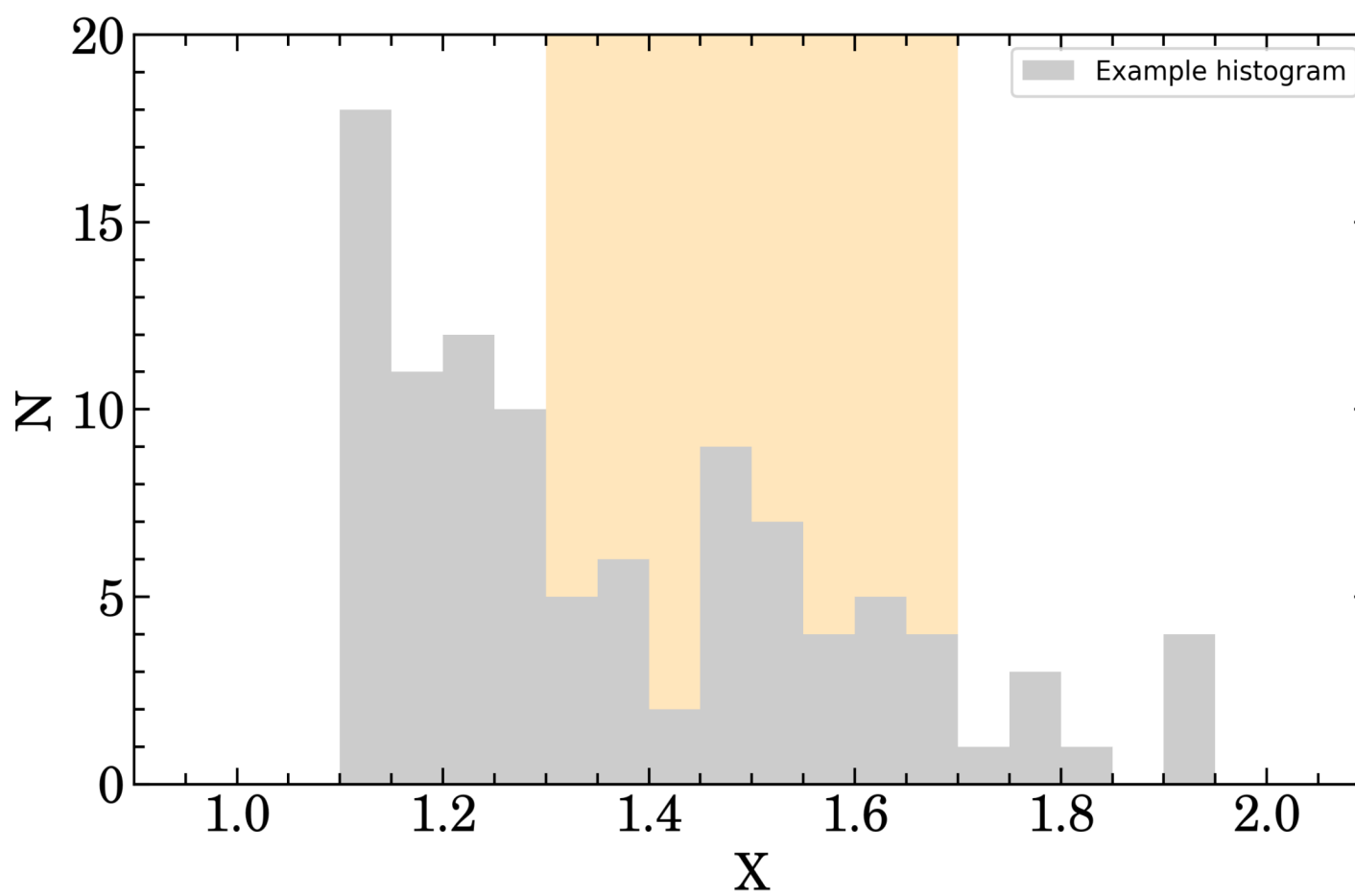
Photon is a GUI created to make simple 1D plot in python. It uses mainly matplotlib and PyQt5 and it has been build in the spirit of being fully customizable. You can change fontstyle, fontsize, fontcolors, linewidth of the axes, thickness, etc... and see the changes directly in the plot. Once a customization 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.

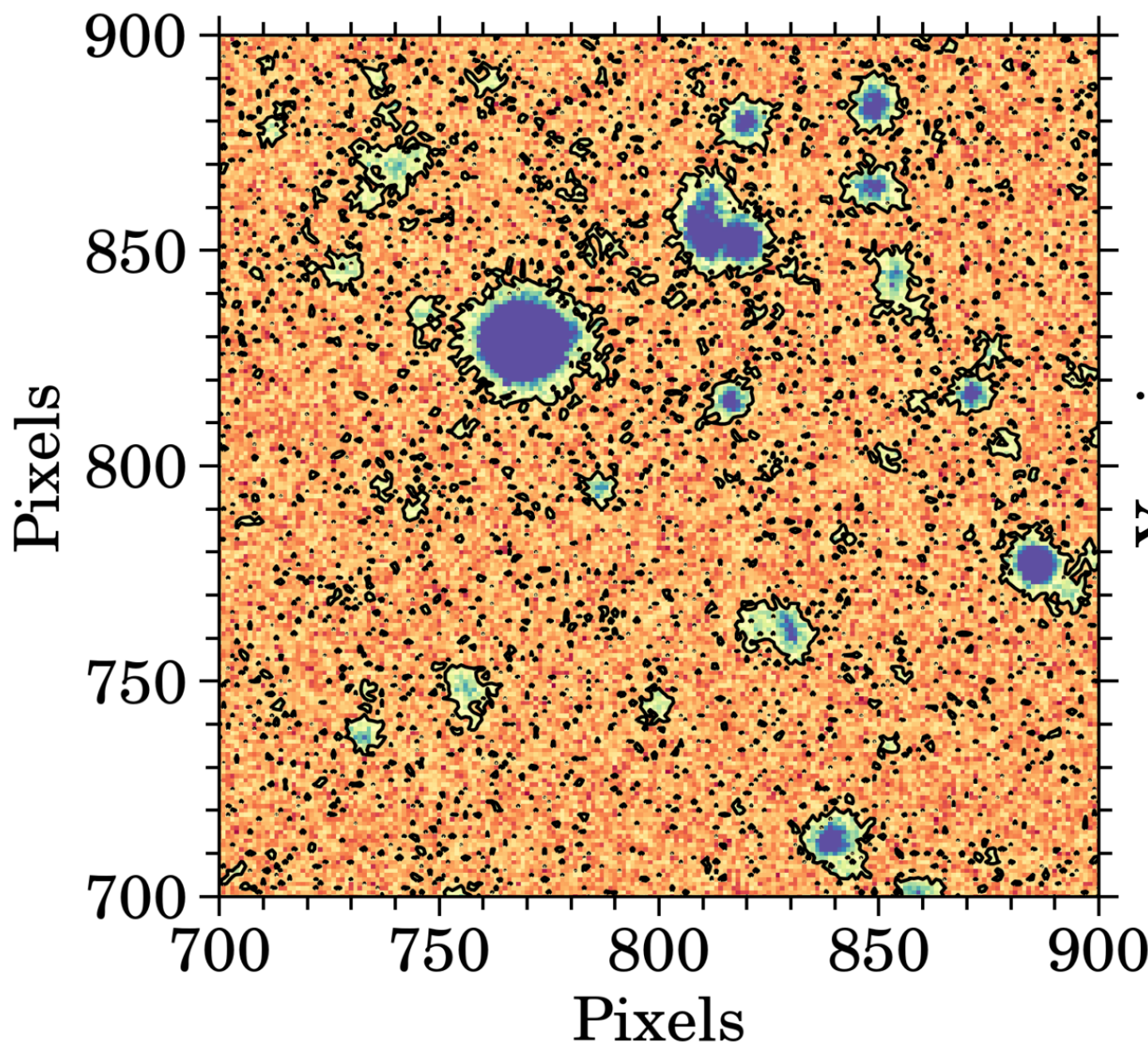
Contribute!











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/>.